



كلية الطب
والصيدلة - مراكش
FACULTÉ DE MÉDECINE
ET DE PHARMACIE - MARRAKECH

Year: 2020

Thesis N° 218

Introducing Electronic Health Records to Automate Medical Research

THESIS

PRESENTED AND DEFENDED PUBLICLY ON 12/11/2020

BY

Mr. Zakaria CHABIHI

Born on 09 December 1994 in Dammam, Saudi Arabia

TO OBTAIN THE DEGREE OF DOCTOR OF MEDICINE

KEYWORDS

Electronic Health Record – Hospital Management System – Content Management
System – Clinical Data Warehouse – Ontology – Big Data – Data Science – Data
Analysis – Data Querying – Artificial Intelligence

JURY

Mr. A.R. EL ADIB

Professor of anesthesia and intensive care

PRESIDENT

Mr. F.M.R. MAOULAININE

Professor of pediatrics

SUPERVISOR

Mrs. L. ADARMOUCH

Professor of community medicine

JUDGES

Hippocratic Oath

I swear to fulfill, to the best of my ability and judgment, this covenant:

*I will respect the hard-won scientific gains of those physicians in whose steps I walk,
and gladly share such knowledge as is mine with those who are to follow.*

*I will apply, for the benefit of the sick, all measures [that] are required, avoiding
those twin traps of overtreatment and therapeutic nihilism.*

*I will remember that there is art to medicine as well as science, and that warmth,
sympathy, and understanding may outweigh the surgeon's knife or the chemist's drug.*

*I will not be ashamed to say "I know not," nor will I fail to call in my colleagues
when the skills of another are needed for a patient's recovery.*

*I will respect the privacy of my patients, for their problems are not disclosed to me
that the world may know. Most especially must I tread with care in matters of life
and death. If it is given me to save a life, all thanks. But it may also be within my
power to take a life; this awesome responsibility must be faced with great humbleness
and awareness of my own frailty. Above all, I must not play at God.*

*I will remember that I do not treat a fever chart, a cancerous growth, but a sick
human being, whose illness may affect the person's family and economic stability. My
responsibility includes these related problems, if I am to care adequately for the sick,*

I will prevent disease whenever I can, for prevention is preferable to cure.

*I will remember that I remain a member of society, with special obligations to all my
fellow human beings, those sound of mind and body as well as the infirm.*

*If I do not violate this oath, may I enjoy life and art, respected while I live and
remembered with affection thereafter. May I always act so as to preserve the finest
traditions of my calling and may I long experience the joy of healing those who seek
my help.*

Declaration of Geneva, 1948



Professors list

**UNIVERSITE CADI AYYAD FACULTE DE
MEDECINE ET DE PHARMACIE MARRAKECH**

Doyens Honoraires :

Pr. Badie Azzaman MEHADJI

Pr. Abdelhaq ALAOUI YAZIDI

Administration

Doyen :

Pr. Mohammed BOUSKRAOUI

Vice doyen à la Recherche et la Coopération :

Pr. Mohamed AMINE

Vice doyen aux Affaires Pédagogiques :

Pr. Redouane EL FEZZAZI

Secrétaire Générale :

Mr. Azzeddine EL HOUDAIGUI

Professeurs de l'enseignement supérieur

Nom et Prénom	Spécialité	Nom et Prénom	Spécialité
ABKARI Imad	Traumato- orthopédie	FAKHIR Bouchra	Gynécologie- obstétrique
ABOU EL HASSAN Taoufik	Anesthésie- réanimation	FINECH Benasser	Chirurgie – générale
ABOUCHADI Abdeljalil	Stomatologie et chir maxillo faciale	FOURAIJI Karima	Chirurgie pédiatrique
ABOULFALAH Abderrahim	Gynécologie- obstétrique	GHANNANE Houssine	Neurochirurgie
ABOUSSAIR Nisrine	Génétique	GHOUNDALE Omar	Urologie
ADALI Imane	Psychiatrie	HACHIMI Abdelhamid	Réanimation médicale
ADERDOUR Lahcen	Oto- rhino- laryngologie	HAJJI Ibtissam	Ophtalmologie
ADMOU Brahim	Immunologie	HAROU Karam	Gynécologie- obstétrique
AGHOUTANE EI Mouhtadi	Chirurgie pédiatrique	HOCAR Ouafa	Dermatologie
AISSAOUI Younes	Anesthésie – réanimation	JALAL Hicham	Radiologie
AIT AMEUR Mustapha	Hématologie Biologique	KAMILI El Ouafi El Aouni	Chirurgie pédiatrique
AIT BENALI Said	Neurochirurgie	KHALLOUKI Mohammed	Anesthésie- réanimation
AIT BENKADDOUR Yassir	Gynécologie- obstétrique	KHATOURI Ali	Cardiologie
AIT-SAB Imane	Pédiatrie	KHOUCHANI Mouna	Radiothérapie
ALAOUI Mustapha	Chirurgie- vasculaire	KISSANI Najib	Neurologie

	périphérique		
AMAL Said	Dermatologie	KOULALI IDRISSE Khalid	Traumato- orthopédie
AMINE Mohamed	Epidémiologie- clinique	KRATI Khadija	Gastro- entérologie
AMMAR Haddou	Oto-rhino-laryngologie	KRIET Mohamed	Ophtalmologie
AMRO Lamyae	Pneumo- phtisiologie	LAGHMARI Mehdi	Neurochirurgie
ANIBA Khalid	Neurochirurgie	LAKMICHI Mohamed Amine	Urologie
ARSALANE Lamiae	Microbiologie -Virologie	LAOUAD Inass	Néphrologie
ASMOUKI Hamid	Gynécologie- obstétrique	LOUHAB Nisrine	Neurologie
ATMANE El Mehdi	Radiologie	LOUZI Abdelouahed	Chirurgie - générale
BAIZRI Hicham	Endocrinologie et maladies métaboliques	MADHAR Si Mohamed	Traumato- orthopédie
BASRAOUI Dounia	Radiologie	MANOUDI Fatiha	Psychiatrie
BASSIR Ahlam	Gynécologie- obstétrique	MANSOURI Nadia	Stomatologie et chiru maxillo faciale
BELKHOU Ahlam	Rhumatologie	MAOULAININE Fadl mrabih rabou	Pédiatrie (Neonatalogie)
BEN DRISS Laila	Cardiologie	MATRANE Aboubakr	Médecine nucléaire
BENCHAMKHA Yassine	Chirurgie réparatrice et plastique	MOUAFFAK Youssef	Anesthésie - réanimation
BENELKHAIAT BENOMAR Ridouan	Chirurgie - générale	MOUDOUNI Said Mohammed	Urologie
BENHIMA Mohamed Amine	Traumatologie - orthopédie	MOUFID Kamal	Urologie
BENJILALI Laila	Médecine interne	MOUTAJ Redouane	Parasitologie
BENZAROUEL Dounia	Cardiologie	MOUTAOUAKIL Abdeljalil	Ophtalmologie
BOUCHENTOUF Rachid	Pneumo- phtisiologie	MSOUGGAR Yassine	Chirurgie thoracique
BOUKHANNI Lahcen	Gynécologie- obstétrique	NAJEB Youssef	Traumato- orthopédie
BOUKHIRA Abderrahman	Biochimie - chimie	NARJISS Youssef	Chirurgie générale
BOUMZEBRA Drissi	Chirurgie Cardio- Vasculaire	NEJMI Hicham	Anesthésie- réanimation
BOURRAHOUE Aicha	Pédiatrie	NIAMANE Radouane	Rhumatologie
BOURROUS Monir	Pédiatrie	OUALI IDRISSE Mariem	Radiologie
BOUSKRAOUI Mohammed	Pédiatrie	OULAD SAIAD Mohamed	Chirurgie pédiatrique
CHAFIK Rachid	Traumato- orthopédie	QACIF Hassan	Médecine interne
CHAKOUR Mohamed	Hématologie Biologique	QAMOUSS Youssef	Anesthésie- réanimation
CHELLAK Saliha	Biochimie- chimie	RABBANI Khalid	Chirurgie générale

CHERIF IDRISSE EL GANOUNI Najat	Radiologie	RADA Noureddine	Pédiatrie
CHOULLI Mohamed Khaled	Neuro pharmacologie	RAIS Hanane	Anatomie pathologique
DAHAMI Zakaria	Urologie	RAJI Abdelaziz	Oto-rhino- laryngologie
DRAISS Ghizlane	Pédiatrie	ROCHDI Youssef	Oto-rhino- laryngologie
EL ADIB Ahmed Rhassane	Anesthésie- réanimation	SAIDI Halim	Traumato- orthopédie
EL ANSARI Nawal	Endocrinologie et maladies métaboliques	SAMKAOUI Mohamed Abdenasser	Anesthésie- réanimation
EL BARNI Rachid	Chirurgie- générale	SAMLANI Zouhour	Gastro- entérologie
EL BOUCHTI Imane	Rhumatologie	SARF Ismail	Urologie
EL BOUIHI Mohamed	Stomatologie et chir maxillo faciale	SORAA Nabila	Microbiologie – Virologie
EL FEZZAZI Redouane	Chirurgie pédiatrique	SOUMMANI Abderraouf	Gynécologie- obstétrique
EL HAOURY Hanane	Traumato- orthopédie	TASSI Noura	Maladies infectieuses
EL HATTAOUI Mustapha	Cardiologie	TAZI Mohamed Illias	Hématologie- clinique
EL HOUDZI Jamila	Pédiatrie	YOUNOUS Said	Anesthésie- réanimation
EL IDRISSE SLITINE Nadia	Pédiatrie	ZAHLANE Kawtar	Microbiologie – virologie
EL KARIMI Saloua	Cardiologie	ZAHLANE Mouna	Médecine interne
EL KHAYARI Mina	Réanimation médicale	ZAOUI Sanaa	Pharmacologie
EL MGHARI TABIB Ghizlane	Endocrinologie et maladies métaboliques	ZIADI Amra	Anesthésie – réanimation
ELFIKRI Abdelghani	Radiologie	ZOUHAIR Said	Microbiologie
ESSAADOUNI Lamiaa	Médecine interne	ZYANI Mohammed	Médecine interne
FADILI Wafaa	Néphrologie		

Professeurs Agrégés

Nom et Prénom	Spécialité	Nom et Prénom	Spécialité
ABIR Badreddine	Stomatologie et Chirurgie maxillo faciale	EL OMRANI Abdelhamid	Radiothérapie
ADARMOUCH Latifa	Médecine Communautaire (médecine préventive, santé publique et hygiène)	FAKHRI Anass	Histologie– embryologie cytogénétique
AIT BATAHAR Salma	Pneumo– phtisiologie	IHBIBANE fatima	Maladies Infectieuses
ALJ Soumaya	Radiologie	KADDOURI Said	Médecine interne
ARABI Hafid	Médecine physique et réadaptation fonctionnelle	LAHKIM Mohammed	Chirurgie générale
ARSALANE Adil	Chirurgie Thoracique	LAKOUICHMI Mohammed	Stomatologie et Chirurgie maxillo faciale
BELBACHIR Anass	Anatomie– pathologique	MARGAD Omar	Traumatologie – orthopédie
BELBARAKA Rhizlane	Oncologie médicale	MLIHA TOUATI Mohammed	Oto–Rhino – Laryngologie
BELHADJ Ayoub	Anesthésie –Réanimation	MOUHSINE Abdelilah	Radiologie
BENALI Abdeslam	Psychiatrie	NADER Youssef	Traumatologie – orthopédie
BENJELLOUN HARZIMI Amine	Pneumo– phtisiologie	OUBAHA Sofia	Physiologie
BOUZERDA Abdelmajid	Cardiologie	RBAIBI Aziz	Cardiologie
BSISS Mohamed Aziz	Biophysique	SAJIAI Hafsa	Pneumo– phtisiologie
CHRAA Mohamed	Physiologie	SALAMA Tarik	Chirurgie pédiatrique
DAROUASSI Youssef	Oto–Rhino – Laryngologie	SEDDIKI Rachid	Anesthésie – Réanimation
EL AMRANI Moulay Driss	Anatomie	SERGHINI Issam	Anesthésie – Réanimation
EL HAOUATI Rachid	Chirurgie Cardio– vasculaire	TOURABI Khalid	Chirurgie réparatrice et plastique
EL KAMOUNI Youssef	Microbiologie Virologie	ZARROUKI Youssef	Anesthésie – Réanimation
EL KHADER Ahmed	Chirurgie générale	ZEMRAOUI Nadir	Néphrologie
EL MEZOUARI El Moustafa	Parasitologie Mycologie		

Professeurs Assistants

Nom et Prénom	Spécialité	Nom et Prénom	Spécialité
ABDELFETTAH Youness	Rééducation et Réhabilitation Fonctionnelle	ELOUARDI Youssef	Anesthésie réanimation
ABDOU Abdessamad	Chiru Cardio vasculaire	EL-QADIRY Rabiyy	Pédiatrie
ABOULMAKARIM Siham	Biochimie	ESSADI Ismail	Oncologie Médicale
ACHKOUN Abdessalam	Anatomie	FDIL Naima	Chimie de Coordination Bio- organique
AIT ERRAMI Adil	Gastro-entérologie	FENNANE Hicham	Chirurgie Thoracique
AKKA Rachid	Gastro – entérologie	HAJHOUI Farouk	Neurochirurgie
ALAOUI Hassan	Anesthésie – Réanimation	HAJJI Fouad	Urologie
AMINE Abdellah	Cardiologie	HAMMI Salah Eddine	Médecine interne
ARROB Adil	Chirurgieréparatrice et plastique	Hammoune Nabil	Radiologie
ASSERRAJI Mohammed	Néphrologie	HAMRI Asma	Chirurgie Générale
AZIZ Zakaria	Stomatologie et chirurgie maxillo faciale	JALLAL Hamid	Cardiologie
BAALLAL Hassan	Neurochirurgie	JANAH Hicham	Pneumo- phtisiologie
BABA Hicham	Chirurgie générale	LAFFINTI Mahmoud Amine	Psychiatrie
BELARBI Marouane	Néphrologie	LAHLIMI Fatima Ezzahra	Hématologie clinique
BELFQUIH Hatim	Neurochirurgie	LAHMINI Widad	Pédiatrie
BELGHMAIDI Sarah	OPhtalmologie	LALYA Issam	Radiothérapie
BELLASRI Salah	Radiologie	LAMRANI HANCH Asmae	Microbiologie-virologie
BENANTAR Lamia	Neurochirurgie	LOQMAN Souad	Microbiologie et toxicologie environnementale
BENNAOUI Fatiha	Pédiatrie	MAOUJOURD Omar	Néphrologie
BENZALIM Meriam	Radiologie	MEFTAH Azzelarab	Endocrinologie et maladies métaboliques
BOUTAKIOUTE Badr	Radiologie	MILOUDI Mohcine	Microbiologie – Virologie
CHAHBI Zakaria	Maladies infectieuses	NASSIH Houda	Pédiatrie
CHETOUI Abdelkhalek	Cardiologie	NASSIM SABAH Taoufik	Chirurgie Réparatrice et Plastique
CHETTATI Mariam	Néphrologie	OUMERZOUK Jawad	Neurologie
DAMI Abdallah	Médecine Légale	RAGGABI Amine	Neurologie

DARFAOUI Mouna	Radiothérapie	RAISSI Abderrahim	Hématologie clinique
DOUIREK Fouzia	Anesthésie–réanimation	REBAHI Houssam	Anesthésie – Réanimation
EL– AKHIRI Mohammed	Oto– rhino–laryngologie	RHARRASSI Isam	Anatomie–patologique
EL AMIRI My Ahmed	Chimie de Coordination bio–organnique	ROUKHSI Redouane	Radiologie
EL FADLI Mohammed	Oncologie médicale	SALLAHI Hicham	Traumatologie–orthopédie
EL FAKIRI Karima	Pédiatrie	SAYAGH Sanae	Hématologie
EL GAMRANI Younes	Gastro–entérologie	SBAAI Mohammed	Parasitologie–mycologie
EL HAKKOUNI Awatif	Parasitologie mycologie	SEBBANI Majda	Médecine Communautaire (médecine préventive, santé publique et hygiène)
EL HAMZAoui Hamza	Anesthésie réanimation	SIRBOU Rachid	Médecine d'urgence et de catastrophe
EL KHASSOUI Amine	Chirurgie pédiatrique	WARDA Karima	Microbiologie
ELATIQI Oumkeltoum	Chirurgie réparatrice et plastique	ZBITOU Mohamed Anas	Cardiologie
ELBAZ Meriem	Pédiatrie	ZOUIZRA Zahira	Chirurgie Cardio–vasculaire
ELJAMILI Mohammed	Cardiologie		

LISTE ARRÊTÉE LE 01/10/2020

Dedications

«Let us be grateful to the people who give us happiness; they are the charming gardeners by whom our souls are flowered»

Marcel Proust.



I fully admit my gratitude to all the people who supported me during my journey, who were able to hoist me up to achieve my goal. It is with love, respect, and gratitude:

I dedicate this thesis 

***To my beloved Father, Mr. Nordine
CHABIHI***

All the existing words of love in this universe can't express the endless gratitude and the love that I carry for you. If I could write a story, it would be the best story of a brave father who protects and provides for his children, a perfect father with the most beautiful soul and the purest heart on this earth. Every step I took was made easier with your blessed prayers and your supports. You have always been my source of tenderness, inspiration, guidance that whispers wisdom in my ears and enlightens my path.

My dear dad, I owe you everything in my life, and I hope that I have been up to your expectations, and I have made you so proud as you have always been my source of pride and my example. May Almighty ALLAH protect you and grant your health, long life, and happiness.

I LOVE YOU SO MUCH, DAD

To my beloved Mother, Mrs. Najat AJAJ

No words can be spoken to express my love and attachment to you. You have always been my example because, throughout your life, I have seen only love, respect, seriousness, and kindness. You have always given me your time, energy, freedom, heart, and love. On this day, I hope to fulfill one of your dreams, dear mother, and sweet creature, knowing that everything I could do or say could not equal what you gave me and did for me.

My dear mom, I owe you everything in my life, and I hope that I have been up to your expectations, and I have made you so proud as you have always been my source of pride and my example. May Almighty ALLAH protect you and grant your health, long life, and happiness.

I LOVE YOU SO MUCH, MOM

To my beloved Sister, Mrs. Yassamine CHABIHI

You who were always somewhere by my side, you who blow me words of hope and love and tenderness, you who give me each time the courage to continue my path, it is by your actions and your words through your looks and your smiles, that I was able to cross this long road, and to hold until the end, my dear ones, as loving as they are kind, I offer you this work, which is yours before being mine.

My dear sister, I owe you everything in my life, and I hope that I have been up to your expectations, and I have made you so proud as you have always been my source of pride and my example. May Almighty ALLAH protect you and grant your health, long life, and happiness.

I LOVE YOU SO MUCH, SISTER

***To the memory of my beloved Grandfathers Abdeslam AJAJ
and Almaati CHABIHI and my Grandmother Fetouma
JABER***

*Your unique love has always submerged my life. Thank you
for your sincere prayers that helped me through life. I pray
Allah to grant you the highest ranks of Jannah.*

***To all my adorable family members: My Grandmother
Rabha CHABIHI, My Uncles and Aunts,***

***My cousins Soufiane, Amine, Omar, Houssam, Saad,
Adnane, Faical, Rajae, Kawtar***

*Families are the first in our lives; as we grow, we always
return to those who love, encourage, and support us the most,
our family. Throughout the years, I had many challenges, but
you were still there, offering support and encouragement.
Your kindness, generosity, and help have inspired me to
overcome all those challenges. Please find in this modest
work the expression of my deep affection and sincere
gratitude.*

To my dearest friend and soulmate, Achrafe HDIDOU

I always considered you as a brother of mine, my soulmate and my dearest friend, an extraordinary ability to create joy out of simple things; I will never forget all those outrageous times. no one can imagine how much you always we think the same thing, sincere and honest, you can always count on him.

To my dearest friend and colleague Mohammed RAMI

You are the humblest person I've ever known; you are my brother, my colleague, and my soulmate; moreover, you are a genius, and you spark with inspiration; Your speech always motivates me to accomplish the most challenging things.

To my dearest friend and my colleague Abdellah CHAFYQ

The definition of wisdom and charisma, CHAFYQ, is the intellectual doctor by excellence, with whom I can discuss everything conspiratorial/metaphysical/paranormal for centuries.

***To my dearest friend and my colleague Mohammed
BOUSSIF***

The perfect example of the humblest straightforward man, a great man you can count on.

To my dearest friend and my colleague Mohammed

CHEQBOUB

*Mohammed is the most acknowledged straightforward friend;
He always instructs me to do the right things by the simplest
manners.*

To my dearest friend and my colleague Youness

ELKHADIR

*This man shines with wisdom and knowledge; He sparks like
a beam in a sea of darkness.*

To my dearest friend and my colleague Youness WAQILI

*Also known as the Father, Funny and outrageously
straightforward, and he speaks charisma.*

To all my best friends and colleagues:

***My brothers: Nouredine, Farouq, Mohammed, Rachid,
Amine, Hassan, Oussama, Ayoub, Hamza, Mustapha,
Omar.***

***My sisters: Fadwa Charif, Fadwa Chahboun, Boussouab,
Bougherbal, Lamia, Sara, Rihab, Kawtar.***

*We shared memorable moments throughout those past years
we shared memorable moments We worked together, laughed,
and cried together, we made up and shared many more
precious moments of life. Our friendship overpassed the
university benches and hospital wards, and it continues to
bring me joy and happiness in my life.*

*To all of you, my friends that I have mentioned and my friends
that I forgot to mention unintentionally. I am honored and so
grateful that I had the opportunity to meet you. Thank you for
giving me memories that I am going to hold close to my heart
and cherish forever. And May you shine in your career and
your personal lives.*

***To my dear master, Professor Fatiha BENNAOUI Professor
of Newborn Intensive Care Unit***

*I would like to express my sincere gratitude and my deep
consideration for your valuable participation and help in the
development of this work and for your tireless
encouragement.*

*During our sixth-year internship and throughout the period I
have known you, I have only seen but humbleness, kindness,
and respect. I have always admired your big heart and your
politeness.*

*Your kindness, your generosity, and your humility that you
share your knowledge with deserves all of respect and
esteem. Thank you for accepting to be part of this jury .and
please find here the testimony of my high consideration and
deep appreciation.*

***To my dear master, Professor REBAHI Houssam Professor
of Anesthesia and Intensive Care***

*I would like to express my sincere gratitude and my deep
consideration for your valuable participation and help in the
development of this work and for your tireless
encouragement.*

*I am particularly touched by your kindness and humility with
which you have agreed to assess this work. Your professional
background, your undeniable competence, and your human
qualities make you a great professor and inspire me with
great admiration. Allow me, dear master, to express my deep
respect and my highest consideration.*

***To all medical and paramedical staff of the Mohamed VI
University Hospital of Marrakech, Regional hospital of
Beni Mellal.***

***To all the people who have supported me during these years
of medical studies.***

***Please find this work the testimony of my sincere gratitude
and my deep appreciation and respect.***

Acknowledgments

***To my dear master and thesis supervisor, Professor
MAOULAININE Fadl Mrabih Rabou Professor of
Newborn Intensive Care Unit***

I would like to express my sincere gratitude and my deep respect for trusting me to Conduct this study. Your kindness, your human and professional qualities deserve all of admiration, and they make of you a role model for me and for all your students and trainees.

Thank you for your patience, your pertinent advice, and for your guidance in every step during this work. I owe you this research experience, and I hope that I have been up to your expectations.

***To my dear master and thesis president, Professor EL ADIB
Ahmed Rhassane, professor of Anesthesia and Intensive
Care***

Thank you for granting me this great honor by accepting the presidency of this honorable jury. You are the example of a professor with great human and professional qualities. Your seriousness, your competence, and your sense of duty have always been a source of inspiration for me and all your students. Please accept through this work the expression of my sincere gratitude and my deep consideration and respect.

***To my dear master and thesis judge, Professor
ADARMOUCH Latifa, Professor of Community Medicine***

*You have granted me a great honor by accepting being part
of this respectful jury. The kindness you have shown while
receiving this thesis was incredibly touching. Thank you for
your availability, kindness, and professional dedication that
make you a great Master to follow. please find here the
expression of my sincere gratitude and my deep
consideration.*

List of abbreviations

List of Abbreviations

ABAC	Attribute-Based Access Control	IETF	Internet Engineering Task Force
AC	Access Control	ISO	International Organization for Standardization
ACL	Access Control List	IT	Information Technology
AI	Artificial Intelligence	JS	JavaScript
AJAX	Asynchronous JavaScript and XM	JSON	JavaScript Object Notation
ALAT	Alanine Aminotransferase	JWA	JSON Web Algorithms
API	Application Programming Interface	JWK	JSON Web Key
ASAT	Aspartate Aminotransferase	JWS	JSON Web Signature
AST	Aspartate Aminotransferase	JWT	JSON Web Token
ATC	Anatomical Therapeutic Chemical Classification System	LL	Left-to-right parser
BERT	Bidirectional Encoder Representations from Transformers	LR	Left-to-right parser
CBAC	Context-Based Access Control	LTR	Left to right
CBUE	Cytology-bacterial Urine Examination	MAC	Mandatory Access Control
CDR	Clinical Data Repository	MBPS	Megabits Per Second
CDS	Clinical Decision Support	MHz	Megahertz
CDSS	Clinical Decision Support System	MIT	Massachusetts Institute of Technology
CDW	Clinical Data Warehouse	NIST	National Institute of Standards and Technology
CIM	Classification International des Maladies	NLI	Natural Language Interface
CMS	Content Management System	NLM	National Library of Medicine
CNN	Convolutional Neural Network	NLP	Natural Language Processing
CRP	C Reactive Protein	NLU	Natural Language Understanding
CRUD	Create, Read, Update and Delete	NPM	Node Package Manager
CSF	Cerebrospinal Fluid	OED	Oxford English Dictionary
CSRF	Cross-Site Request Forgery	OPTICS	Ordering points to identify the clustering structure
CSS	Cascading Stylesheets	OS	Operating System
CSV	Comma Separated Values	PHR	Personal Health Record
DAC	Discretionary Access Control	PMS	Practice Management System
DBSCAN	Density-Based Spatial Clustering of Applications with Noise	POS	Part-of-speech

DDD	Domain-Driven Development	QR	Quick Response code
DDI	Development Dimensions International	RBAC	Role-based access control
DICOM	Digital Imaging and Communications in Medicine	RFC	Request for Comment
DQL	Data Query Language	RNN	Recurrent Neural Network
EHR	Electronic Health Record	ROS	Review of Systems
EM	Expectation-Maximization algorithm	RTL	Right to Left
EMR	Electronic Medical Record	SDD	Security-driven development
ERBAC	Entity-Relationship Based Access Control	SDLC	Software development lifecycle
ERD	Entity-Relationship Diagram	SLA	Service Liability Agreement
GAN	Generative Adversarial Networks	SQL	Structured Query Language
GB	Gradient Boosting	TDD	Test-Driven Development
GBAC	Graph-Based Access Control	TGO	Glutamate-oxaloacetate-transaminase
GUI	Graphical User Interface	TGP	Glutamate-oxaloacetate-transaminase
HDD	Hard Disk Drive	TLS	Transport Layer Security
HIPAA	Health Insurance Portability and Accountability Act	UAT	User Acceptance Testing
HIS	Healthcare Information System	UI	User Interface
HMS	Hospital Management System	UML	Unified Modeling Language
HR	Human Resources	URL	Uniform Resource Locator
HTML	Hypertext Markup Language	US	United States
HTTP	Hypertext Transfer Protocol	USA	United States of America
IAM	Identity and Access Management	UX	User Experience
IBM	International Business Machines Corporation	WHO	World Health Organization
ICASSP	International Conference on Acoustics, Speech and Signal Processing	WHOCC	WHO Collaborating Centre for Drug Statistics Methodology
ICD	International Classification of Diseases	WYSIWYG	What You See is What You Get
ID	Identity	XML	Extensible Markup Language
IDS	Intrusion Detection System	YAML	YAML Ain't Markup Language
IEEE	Institute of Electrical and Electronics Engineers		

List of tables

List of Tables

Table I: major themes covered by meta-research [1].....	13
Table II: major application contexts	31
Table III: data quality guarantees missing on free-entry input systems	34
Table IV: <i>meta</i> suite dependencies	40
Table V: admission unconventional CRUD actions	67
Table VI: admission record components	69
Table VII: admission listing variants.....	70
Table VIII: procedure unconventional CRUD actions	71
Table IX: supported procedure report types	72
Table X: procedure listing variants.....	78
Table XI: prescription unconventional CRUD actions.....	80
Table XII: prescription record components.....	81
Table XIII: prescription listing variants	84
Table XIV: consulting unconventional CRUD actions.....	85
Table XV: consulting listing variants	86
Table XVI: consent unconventional CRUD actions.....	88
Table XVII: informed consent requirements [109,111]	88
Table XVIII: consent listing variants	89
Table XIX: appointment unconventional CRUD actions	91
Table XX: appointment listing variants.....	92
Table XXI: <i>metatype</i> primitive datatypes.....	98
Table XXII: segment property metadata	99
Table XXIII: supported scheme types	101
Table XXIV: action entity types	105
Table XXV: action entity metadata	106
Table XXVI: action entity metadata availability matrix.....	107
Table XXVII: variant entity metadata	111
Table XXVIII: item entity metadata	113
Table XXIX: view metadata components.....	115
Table XXX: listing component supported operations	116
Table XXXI: listing component elements	117
Table XXXII: listing types configurations	118
Table XXXIII: supported viewer types	119
Table XXXIV: grid viewer metadata	119
Table XXXV: row viewer metadata	120
Table XXXVI: list viewer metadata	121
Table XXXVII: layout entity metadata	122

Table XXXVIII: user experience principles followed when crafting the editor component.....	125
Table XXXIX: supported listing component operations	127
Table XL: supported account management operations	138
Table XLI: supported user management operations	141
Table XLII: supported operations that can be carried on ontology	145
Table XLIII: themes covered by the ontology	146
Table XLIV: attributes covered by the ontology	147
Table XLV: supported list data transformations.....	149
Table XLVI: supported visualization plot types	153
Table XLVII: <i>metatext</i> built-in locale-specific features	158
Table XLVIII: patients gender count	161
Table XLIX: admissions count by gender and status	162
Table L: admissions count by reason	164
Table LI: admissions count by reason category	165
Table LII: Apgar score mean by admission status	166
Table LIII: Farr score mean by admission status	166
Table LIV: unretained admission diagnosis category count.....	167
Table LV: maternal age mean by status.....	168
Table LVI: procedure category count.....	169
Table LVII: procedure requests count.....	171
Table LVIII: mean of admission anthropometrics by status	172
Table LIX: correlation between the instructor occupation and admission status	173
Table LX: correlation between delivery place and admission status	174
Table LXI: correlation between delivery route and admission status	175
Table LXII: correlation between delivery durations with admission status	175
Table LXIII: correlation between pregnancy follow-up status and admission status	176
Table LXIV: correlation between consanguinity status and admission status.....	177
Table LXV: correlation between consanguinity status and admission diagnosis	178
Table LXVI: parity and gestation mean by admission status.....	179
Table LXVII: data quality dimensions [48]	183
Table LXVIII: query writing constraints.....	193

List of figures

List of Figures

Figure 1: research lifecycle by step [2]	15
Figure 2: data lifecycle steps [52]	17
Figure 3: research data lifecycle [53]	17
Figure 4: software development lifecycle (SDLC) [56]	19
Figure 5: Agile software development methodology [57,58]	21
Figure 6: Scrum framework [63]	22
Figure 7: multilayered defense in depth [71]	28
Figure 8: level 1 c4 diagram applied to our system	32
Figure 9: level 1 container c4 Model diagram for content container	33
Figure 10: Trie data structure (digital prefix tree) [81]	35
Figure 11: ontology container C4 Model level 2 component-level diagram	35
Figure 12: basic datatypes in programming	36
Figure 13: parsing container and its layers of abstraction	39
Figure 14: captured patient demographic information	42
Figure 15: patient record relation with admission record	43
Figure 16: admission status component	43
Figure 17: medical history component	44
Figure 18: disease ontology picker	44
Figure 19: surgical history component	45
Figure 20: allergic history component	45
Figure 21: obstetric history component	46
Figure 22: gynecologic history component	46
Figure 23: delivery history component	47
Figure 24: family history component	47
Figure 25: problem history component	48
Figure 26: general examination component	49
Figure 27: pulmonary examination component	49
Figure 28: cardiovascular examination component	50
Figure 29: abdominal examination component	50
Figure 30: genitourinary examination component	51
Figure 31: neurologic examination component	51
Figure 32: rheumatologic examination component	52
Figure 33: newborn examination component	52
Figure 34: newborn reflexes examination component	53
Figure 35: newborn apparent malformations examination component	53
Figure 36: newborn unapparent malformations examination component	54
Figure 37: assessment component	54

Figure 38: intervention component.....	55
Figure 39: intervention types ontology picker	55
Figure 40: diagnosis component	56
Figure 41: vitals monitoring component	57
Figure 42: scope monitoring component	57
Figure 43: ventilation monitoring component	58
Figure 44: blood gas monitoring component	58
Figure 45: ionogram monitoring component	59
Figure 46: kidney monitoring component	59
Figure 47: CBUE monitoring component	60
Figure 48: blood culture monitoring component.....	60
Figure 49: blood cell count monitoring component.....	61
Figure 50: biochemistry monitoring component	61
Figure 51: obstetric monitoring component.....	62
Figure 52: anthropometrics monitoring component.....	62
Figure 53: summary component.....	63
Figure 54: hospital management system	64
Figure 55: create patient action	66
Figure 56: patients listing interface	66
Figure 57: create admission action	68
Figure 58: request admission action	68
Figure 59: admission records listing.....	70
Figure 60: supported procedure templates	73
Figure 61: blood count panel report	73
Figure 62: ionogram panel report.....	73
Figure 63: kidney function panels report	74
Figure 64: liver function panels report.....	74
Figure 65: cardiac function panels report	74
Figure 66: gasometry panel report.....	75
Figure 67: hematology panel report.....	75
Figure 68: iron metabolism panel report.....	75
Figure 69: lipid metabolism panel report	75
Figure 70: common biochemistry panels report.....	76
Figure 71: cyto-bacterial urine examination procedure report	76
Figure 72: microbiology panels report.....	76
Figure 73: lumbar punction panel report	76
Figure 74: imaging procedures report	77
Figure 75: generic biochemistry table report	77
Figure 76: generic text report.....	77

Figure 77: generic key/value report.....	77
Figure 78: generic x/y report	77
Figure 79: create procedure action	78
Figure 80: procedure reporting interface	79
Figure 81: procedure records listing	79
Figure 82: prescription intake component	81
Figure 83: prescription enteral feeding component.....	82
Figure 84: prescription electrolyte perfusion component	82
Figure 85: prescription parenteral perfusion component.....	82
Figure 86: prescription auto perfusion component	82
Figure 87: prescription medication component.....	83
Figure 88: prescription transfusion component.....	83
Figure 89: prescription records listing	84
Figure 90: generic consulting reporting component.....	85
Figure 91: create consulting action.....	86
Figure 92: request consulting action.....	87
Figure 93: consulting records listing	87
Figure 94: create consent action.....	90
Figure 95: consent records listing	91
Figure 96: generic appointment reporting component	92
Figure 97: create appointment action	93
Figure 98: appointment records listing	93
Figure 99: create guide action	94
Figure 100: guide records listing	95
Figure 101: data structures hierarchy	96
Figure 102: JSON data structures.....	97
Figure 103: segment entity editor	100
Figure 104: segment property editor	100
Figure 105: segment entities listing.....	101
Figure 106: list scheme editor	102
Figure 107: map scheme editor	103
Figure 108: reference scheme editor	103
Figure 109: relation scheme editor	104
Figure 110: segment scheme editor.....	104
Figure 111: scheme entities listing	105
Figure 112: action create type	107
Figure 113: action update type.....	108
Figure 114: action delete type	109
Figure 115: action read type.....	110

Figure 116: action entities listing	111
Figure 117: variant entity editor	112
Figure 118: variant entities listing	113
Figure 119: item entity editor	114
Figure 120: item entities listing	115
Figure 121: view entity editor	116
Figure 122: type configuration editor	118
Figure 123: grid viewer editor	120
Figure 124: row viewer editor	120
Figure 125: list viewer editor	121
Figure 126: view entities listing	122
Figure 127: layout entity editor	123
Figure 128: layout entities listing	124
Figure 129: partial printing example	126
Figure 130: printing result example	127
Figure 131: listing filters component	128
Figure 132: listing sorting component	129
Figure 133: listing range checking component	130
Figure 134: file upload component	131
Figure 135: image on-the-fly transformation	131
Figure 136: upload file preview	132
Figure 137: upload file details	132
Figure 138: upload file metadata	133
Figure 139: storage items listing	133
Figure 140: role-based access control	134
Figure 141: role grants matrix editor	135
Figure 142: role entities listing	136
Figure 143: attribute-based access control	137
Figure 144: user metadata attributes	137
Figure 145: register new account interface	139
Figure 146: login to an existing account	139
Figure 147: send recovery code interface	139
Figure 148: recover account interface	140
Figure 149: change account details interface	140
Figure 150: user records listing	141
Figure 151: sessions listing	143
Figure 152: Trie data structure (digital prefix tree)	144
Figure 153: clinical data warehouse [124]	148
Figure 154: available predefined query templates	150

Figure 155: query expression editor	151
Figure 156: query visual editor	151
Figure 157: query results interface	152
Figure 158: data visualization plots	153
Figure 159: chart variant editor	154
Figure 160: chart entity editor	155
Figure 161: chart entities listing	155
Figure 162: board entity editor.....	156
Figure 163: board entities listing.....	157
Figure 164: tag entity editor.....	159
Figure 165: tag entities listing.....	159
Figure 166: language entity editor.....	160
Figure 167: patient gender count query.....	162
Figure 168: admissions count by gender and status	163
Figure 169: admissions count by reason query	164
Figure 170: admissions count by reason category query	165
Figure 171: Apgar score mean by admission status query.....	166
Figure 172: Farr score mean by admission status query.....	167
Figure 173: unretained admission diagnosis category count query	168
Figure 174: maternal age mean by status query.....	169
Figure 175: procedure category count query	170
Figure 176: procedure requests count query	171
Figure 177: mean of admission anthropometrics by status query.....	172
Figure 178: correlation between the instructor occupation and admission status query.....	173
Figure 179: correlation between delivery place and admission status query	174
Figure 180: correlation between delivery route and admission status query	175
Figure 181: correlation between delivery durations with admission status query	176
Figure 182: correlation between pregnancy follow-up status and admission status query	177
Figure 183: correlation between consanguinity status and admission status query.....	178
Figure 184: correlation between consanguinity status and admission diagnosis query	179
Figure 185: parity and gestation mean by admission status query	180
Figure 186: the seven "V"s of Big Data [136]	185
Figure 187: supervised learning vs. unsupervised learning [147].....	192
Figure 188: types of artificial neural networks	195
Figure 189: natural language processing applications [88,142,155].....	197
Figure 190: security by design applied to the Agile methodology [171]	221
Figure 191: security architecture [172]	222
Figure 192: security measures [172].....	223
Figure 193: identity management [175].....	224

Table of contents

Table of Contents

Introduction.....	1
Background	3
I. Medical record	4
1. Definition	4
A. Administrative record.....	4
B. Professional record	4
C. Nursing record.....	4
2. History	4
3. Structure	5
4. Roles	6
A. Symbolic role	6
B. Ease physicians workflow	6
C. Coordination and communication tool.....	7
D. Care roles	7
E. Medico-legal role	7
F. Assessment tool.....	7
G. Teaching and research tool	8
II. Computerization of the medical record	8
1. Motive	8
A. Drawbacks of the paper form	8
A. The record size has increased	9
B. The number of records has increased	9
C. The practical importance of the record has increased	9
2. History	9
3. Benefits	10
4. Solutions	11
III. Meta-research	12
1. Research limitations.....	13
2. Medical research.....	14
3. Research lifecycle	15
4. Data lifecycle	16
Materials and methods	18
I. Methodology.....	19
1. Software development lifecycle	19
2. Agile methodology.....	20
3. Scrum framework.....	21
II. Workflow	22
1. Artifacts	22
A. Product backlog	22

B. Sprint backlog.....	23
C. Increment	23
2. Events	23
A. Sprint.....	23
B. Sprint planning	23
C. Daily scrum.....	24
D. Sprint review	24
E. Sprint retrospective.....	24
3. Roles	24
A. Product owner.....	24
B. Development team	24
C. Scrum master	24
4. Software testing.....	25
A. Unit testing	25
B. Integration testing	25
C. Acceptance testing.....	26
III. Considerations.....	26
1. Standardization	26
2. Structuring	26
1. Software quality	27
2. Security	27
3. Legislation and ethics	28
4. User experience	29
Results	30
I. Architecture	31
1. Level 1: System diagram	31
2. Level 2: container diagram.....	32
A. Content container	32
B. Ontology container	34
C. Datatype container.....	35
D. Expression container.....	37
E. Internationalization container.....	38
F. Parsing container	38
II. Technology stack	39
1. Backend stack.....	39
2. Frontend stack.....	40
3. Development stack	41
III. System containers	42
1. Electronic medical record.....	42
A. Patient demographics.....	42
B. Admission status	43

C. History component.....	43
D. Examination component.....	48
E. Assessment component	54
F. Intervention component	55
G. Diagnosis component.....	56
H. Monitoring component.....	56
I. Summary component	62
2. Hospital management system	63
A. Patient management	64
B. Admission management.....	67
C. Procedure management.....	70
D. Prescription management.....	79
E. Consulting management	84
F. Consent management	87
G. Appointment management.....	91
H. Guide management.....	93
3. Multi-purpose content management system	95
A. Prerequisites	97
B. Segment entity	98
C. Scheme entity	101
D. Action entity	105
E. Variant entity	111
F. Item entity	113
G. View entity	115
H. Layout entity	122
I. Editor component	124
J. Listing component	127
K. File and storage management.....	130
4. Identity, access management, and security.....	134
A. Role-based access control.....	134
B. Attribute-based access control.....	136
C. Hierarchical role-based access control	138
D. User management.....	138
E. Session management	141
5. Ontology container	143
A. Ontology structure	143
B. Ontology features	144
C. Ontology overview	145
6. Data warehouse system	147
A. Expression container.....	148
B. Query entity	150
C. Chart entity.....	152

D. Board entity	156
7. Localization and internationalization system	157
A. Tags and templates	157
B. Tag entity	158
C. Language entity	160
IV. Software testing	160
1. Unit testing.....	160
2. Integration testing	161
3. Acceptance testing	161
A. Objectives	161
B. Materials and methods	161
C. Results.....	161
Discussion.....	181
I. Project impact.....	182
II. Data quality	183
III. Big data	185
IV. Practical constraints	187
V. French experience.....	188
VI. Research automation	189
1. Headline suggestion	190
2. Data exploration.....	191
3. Data querying	192
4. Reports writing	196
5. Citations management.....	197
6. Research quality and peer-reviewing.....	198
VII. Deployment	199
1. Requirements	199
2. Deployment at scale	200
Conclusion	201
Appendix.....	203
I. Appendix 1: Scrum framework	204
II. Appendix 2: software quality dimensions	218
III. Appendix 3: security considerations and measures.....	220
IV. Appendix 4: legislative considerations	225
V. Appendix 5: user experience considerations	234
Abstract	236
Bibliography	240

Introduction

The automation of research has been talked about in various pretenses over the last few years. The evolutionary steps taken by the scientific community were born out of the desire for a quicker and cheaper route to enhanced knowledge and understanding to elaborate more research publications.

Meta-research is the study of research using scientific paradigms to minimize waste and maximize research quality in all fields. It concerns itself with research efficiency improvement and the detection of bias, methodological flaws, and other errors and inefficiencies [1].

Reviewing research and data lifecycles, we find that data quality, data governance, and data analysis are the pivotal elements in both research and data lifecycles [2,3,4,5]. Assessing the research process, we uncover that it suffers from many limitations subdivided into three main categories [6,7,8,9]: Flaws are methodological errors primarily due to insufficient training and are nearly unintentional [6,10,11]; Fraud is usually deliberate and corrupts scientific findings [6,7,8,9,12]; Constraints are difficulties facing the overall process which can lead to increased financial expenditure, time-waste in addition to low investment return (ROI) ratios [1,7,8,9,11,13,14].

To address these limitations, we sought to: shorten the lifecycle of clinical research, ensure flexible, accurate, and exhaustive data capture of all medical record elements within the minimum duration, assert data quality guarantees, support polymorph, and flexible record schemes, offer a frictionless experience to lower the barrier of entry for researchers, ensure the ease of data exploitation, democratize, anonymize and secure access to clinical data, reduce the linguistic overhead, ensure conformance with personal data protection regulations and laws, being able to construct arbitrary data queries, and to integrate charting and data visualization solutions.

To automate research, these limitations must be addressed; Towards complete pipeline automation, all aspects must be automated, including headlines generation, data quality, data governance, data exploration, data analysis, reports writing, and research quality tools and assessment [15,16].

Therefore, we aimed to introduce health informatics solutions combined with data science solutions to establish a continuous research pipeline and assert data quality attributes and data governance by constructing four major systems: an Electronic Health Record (EHR), a Clinical Data Warehouse (CDW), a Hospital Management System (HMS), and a Practice Management System (PMS).

Background

I. Medical record

1. Definition

The medical record is an integral, written, and continuously updated record that encompasses everything that can be memorized in a patient: administrative, clinical, paraclinical, diagnostic, therapeutic, preventive, prognosis data as well as the intervention of all care providers [17].

A. Administrative record

It captures all the administrative elements that allow a patient to be identified: his professional occupation, his social security coverage, his date of entry into the establishment, and his discharge date [18].

B. Professional record

It is the written memory of clinical, biological, diagnostic, and therapeutic information about a patient, both individual and collective, regularly updated [19].

C. Nursing record

It captures the preventive, curative, educational, relational aspect of care, and the care plan that should be established with the person being treated [19].

2. History

For a long time, the medical record was the simple materialization of a need of the doctor who, fearing his memory's betrayal, kept personal notes that allowed him to remember his patient's history [20].

The earliest patient records were written on the columns of temples in ancient Egypt, Greece, and Rome. Thus the papyrus discovered by Smith near THEBES dated 1700 BC contained the structured description of 48 clinical cases whose descriptive quality was at the origin of the fame of Hippocrates [21].

In the 9th century, when Muslim doctors, such as Rhazès (865–925), Avicenna (930–1037), or Avenzoar (1073–1162), created clinical medicine. The history of interesting cases is thus written and kept in registers such as the "Hospital observations", to which Rhazès alludes in his "Continens" [22].

The notion of a medical record attached to each patient did not appear until the end of the 18th century, like a patient register in god hotels, but the content of which remains succinct [22].

It was until the 19th century when modern hospitals were created that the medical record appeared: it included medical, social, and administrative data. Some hospitals, like the Mayo Clinic in the United States, have placed great importance on the patient's record for medical research. As early as 1931 in the USA, a "medical record", quality recording of medical data in hospitals was considered an ethical requirement [22].

The years 1945–1970 will see major turning points in terms of patient rights and the evolution of medical practices with a deep awareness of reintegrating the person into the patient–doctor relationship, that is to say, from an ethical and legal point of view, integrating information and consent into medical research and care practice [22].

Today, the medical record appears to be a multifaceted tool, defined more by its use than its intrinsic properties [22].

The emergence of new information and communication technologies in the health sector opened wide the doors to the computerization of hospital information systems and the electronic medical record's birth.

3. Structure

Structuring can be defined as the grouping of isolated elements to form more complex objects [18].

The records can be structured along different axes [23]:

- **Structure according to the actors:** medical record, administrative record, nursing record, physiotherapist record, social assistance record, etc.
- **Structure by type of information:** letters, operative reports, delivery reports, anesthesia records, vaccinations, biology examinations, imaging, etc.
- **Temporal structuring:** the various data and information are recorded chronologically, for example, in the observation form, which records the various elements concerning the patient over time.
- **The structuring according to the source:** the current medical record traditionally remains oriented according to the source (or the origin). That is to say, the data obtained from the interview (history, symptoms), clinical examination, additional examinations, diagnostic,

therapeutic, and prognostic data are grouped into separate sections. The diagnostic section is the most crucial section since its function is to integrate as much data as possible to derive the correct and relevant decisions. This model is perfectly functional for specialty records because patients are often hospitalized for a single problem, which gives importance to the data classified by sources that lead to the diagnosis and management of this problem.

- **Structuring by medical problem handled:** the Anglo-Saxons' problem-oriented medical record based on the idea of structuring the plan and progress notes according to a hierarchy rooted in the list of problems. The concept of problem is a broader concept than the concept of diagnosis since it includes any condition requiring further attention for diagnosis, treatment, or monitoring (a problem can be a symptom or a diagnosis). This structure has two main advantages. On the one hand, it highlights the importance of the list of problems (the root of the tree). On the other hand, it forces the user to take a systemized, problem-based approach. However, it includes a certain number of constraints, in part inherent in the hierarchical model chosen, and which limit its dissemination. To be effective, such a model requires relative independence of the branches of the tree, that is, of the data associated with each problem, which is rarely the case for problems concerning the same patient. This model has more space in the files of family physicians who follow their patients for long periods of time. It also finds its application in computerized medical files.

4. Roles

The medical record must first and foremost remain a tool for improving the doctor's work quality and the main instrument for centralizing and coordinating activities within a healthcare establishment for better management of patient health [20].

A. Symbolic role

The medical record represents the sign of the presence, the assumption of responsibility, and the permanence of the bond between the doctor and the patient. It is integrated into the trust contract between the parties. Above all, it must remain an element of the doctor-patient relationship and, as such, guarantee confidentiality and professional secrecy [24].

B. Ease physicians workflow

- **Memory aid**

The amount of information collected is such that it is impossible to memorize everything. The role of the record is to remedy these shortcomings [24]. This function represents the very basis of the creation of the first medical records in history.

- **Guarantee of the medical process**

The medical record contains all patient's past and present decisions to facilitates the development and monitoring of the process of the diagnostic, therapeutic, and preventive actions based on effective scientific methods [24].

C. Coordination and communication tool

A record is a communication tool for patients, fellow specialists, general practitioners, and any other caregiver or person involved in the care process made in compliance with the legal and ethical rules of professional secrecy and with the sole aim of promoting patient care [24,25].

D. Care roles

- **Continuity of care**

A recording of the successive episodes, in addition to the restitution of the antecedents relating to the health problems of the moment, makes it possible to guarantee as much as possible the continuity of care [24].

- **Integration of care**

Good medical record-keeping makes it possible to integrate the curative and preventive aspects in the same episode of care, which implies a multidisciplinary vision of the patient's problems [24].

E. Medico-legal role

Personal health-related data can be used as legal evidence in legal cases where the responsibility of the doctor is engaged. Patients' rights are also better guaranteed on such bases by the adequate recording of data in the event of an accident, disability, etc. [24].

F. Assessment tool

The assessment of medical practices consists, according to the High Authority of Health (HAS), in regularly analyzing its actual practice against the recommendations. Two main methods are possible:

- The a posteriori evaluation (retrospective approach) which consists in analyzing the data on clinical activity (kept in the medical file) against a recommendation for practice.
- The a priori assessment which consists of practicing by following a recommendation, for example by applying a pre-established protocol (prospective approach).

This assessment, made compulsory in France in 2005, is necessary to improve the quality of care [24].

G. Teaching and research tool

Patient records accumulate a large amount of health-related data over time; when made available to researchers, studies are constructed to improve medical practices and knowledge and also contribute to the teaching of new generations of practitioners and in the orientation of health policy in terms of health management and economics [24].

II. Computerization of the medical record

1. Motive

The computerization of the medical record is not only a natural evolution of things, but it also responds to a real need [26].

A. Drawbacks of the paper form

Limits of paper-based medical information management are intuitively apparent and especially in the face of considerable advances in information technology that make these limits clearer and easier to demonstrate [27]. These limits are either attached to the paper medium itself, or to the content and structure of the paper record.

The various studies on the quality of the medical record, of which the clinical audit remains the main one, have made it possible to highlight these drawbacks. These audits have shown that the paper records do not contain, in a large number of cases, elements considered essential such as the identity of the patient or the writer of the record, the reason for hospitalization, the date of entry and discharge, hospital reports and discharge conclusions, etc. And even if they exist, they are sometimes unreadable and poorly maintained. On the other hand, the paper medical record offers only limited access in time and space; that is, the record cannot be accessed when and where we want.

This also hinders its communication. We sometimes see a loss of one or more elements of the record during the transfer between practitioners and sometimes a loss of the entire record. Likewise, the paper record has limitations in terms of archiving since paper has limited durability over time. Thus, several records get torn and degraded after a while and sometimes are not even found. Also, by the volume they constitute, paper records are often eliminated after a given time due to lack of storage space.

All this has harmful consequences on the functions of the medical record, the quality of care, and on scientific and medico-economic research despite all the efforts made [28,29].

A. The record size has increased

For a given patient, for the same disease, in the same structure, the mass of information collected has increased considerably. There are many reasons for this: the shift to a written culture, an effort to be exhaustive, an increase in examinations, the difficulty of sorting information, the management of chronic diseases and aging, the widespread practice of copies, regulatory constraints, and medico-legal concerns, the appearance of paramedical records, etc.

Faced with this inflation, each practitioner or each hospital service tried to organize and structure the record.

In practice, the records are thick, information is difficult to find, and the systematic search for precise information is almost impossible.

B. The number of records has increased

Increasingly, a person's health information is fragmented into multiple specialties. The number of professionals concerned has continued to increase: alongside the general practitioner, the specialist, the nurse, and the pharmacist organize their records. Likewise, the flow of patients to medical services has multiplied.

C. The practical importance of the record has increased

More and more, particularly in an emergency and in a hospital environment or at the interface between care structures, exemplary patient care depends heavily on the information in a previous record that is generally inaccessible. It is not so much general information that the patient knows, but detailed, ad hoc information. In addition to this function of continuity of care, a "perfect" record could also allow a significant development of prevention, evaluation of the quality of care, its timeliness and costs as well as the development of clinical and epidemiological research. . This need is also increasing in the face of the opportunities offered by new information technologies, which can hardly be overlooked [26].

Hence, the computerization of medical records may prove to be beneficial or even necessary [30].

2. History

During a first period, lack of power, everything had to be "coded": in the form of a questionnaire, finely developed, and easily exploitable, which could only concern a very limited field from a research perspective without much practical interest [29].

In a second period, computer power helping; it was possible to integrate whole texts, even images or other signals into the computer records. Electronic storage of an entire record became possible, revealing undeniable logistical advantages. The need to structure the record has, therefore, been reduced [29].

In a later period: electronic information can be shared between health actors, certain standardization of the record, and the principles of exchange were required [29].

3. Benefits

Computerization makes it possible to significantly improve the quality of medical records essentially by the structuring and organization that it provides, and by the IT tools and the possibilities specific to the technologies of information.

A computerized medical record is more readable, more precise, and more complete than a paper record. It can be more exhaustive without being difficult to complete. The fill rate is often high, regardless of the age and IT experience of the physicians who use it. The intra- and inter-doctor variability is significantly reduced compared to the traditional record. Data entry can be guided based on quantitative semiology, checking the entered values against limits, alerting to the absence of essential data thus allowing quality control at source [17,26,31].

An additional contribution consists of the IT tools integrated into the electronic medical record: pharmaceutical and semiological databases such as electronic Vidal, Banque Claude Bernard, etc. are valuable tools for the prescriber. The automatic callback feature helps with care planning and patient education. In this way, screening examinations and vaccinations can be programmed (chronological alerts). The same is true for the additional examinations to be carried out periodically, which contributes enormously to improving medical monitoring, especially in the long term. The possibility of editing pre-established information sheets on an illness, a complimentary examination or prevention sheets (dietary advice, lifestyle, etc.) also help inform and empower patients. The alert function strongly contributes to the quality and safety of care. Thus, for example, the prescription of drugs that interact with each other gives rise to warning messages that allow the doctor to adapt his prescriptions better. It is the same for a patient allergic or carrier of certain pathologies or a pregnant woman. The automatic calculation of certain values or individual scores and the automated production of summaries, reports, and conclusions offer additional support for diagnosis and prescription. They facilitate doctors' work and save a large part of their time, which they can invest with their patients. The representations of information in an adequate form allow a faster and safer synthesis through summary records and chronological graphical views which make the monitoring

of the evolution of clinical and biological parameters easier and more efficient [30,31,32].

Likewise, the management of medical information benefits enormously from computerization. Electronic medical records are both more accessible and better protected than paper records. Access to an electronic record is immediate and can be consulted anywhere and by several stakeholders at the same time. Physical protection is facilitated by their small size, it can be supplemented by logical measures (passwords, data encryption, etc.). Computerized records facilitate data sharing, communication and coordination between the various partners in the healthcare system [30,32,33].

Its communication is fluid, fast and efficient without the risk of loss of documents or data. It is also monitored and regulated in order to protect the ethical and professional rules. The information, once archived and stored in an appropriate format, in a short space of time and space, can be displayed or retrieved in multiple ways depending on the needs of the physician or teacher [26,32,33].

The contribution of computerization to the collective use of medical records is considerable. In public health, epidemiology and health security (sentinel networks, registers, etc.) become more efficient and reliable with the electronic transmission and rapid dissemination of information. Computerized records facilitate the gathering of data for clinical research and the evaluation of practices by ensuring the combination of essential elements: accessibility, completeness, organization, and reliability of data as well as the multiple ways of carrying out research (depending on age, weather, symptoms, treatments, etc.) [26,28].

4. Solutions

Health informatics is an interdisciplinary study of designing, implementing, introducing, and applying IT-based technologies in health services [34].

Health care informatics covers sub-fields of clinical informatics, such as pathology informatics, clinical science informatics, imaging informatics, public health informatics, community health informatics, home health informatics, medical informatics, consumer health informatics, clinical bioinformatics, and health and medicine research and education informatics [35,36,37,38].

There are many systems that can be categorized as clinical informatics; these systems interfere with each other in terms of concern; these systems help enhance patient care by digitizing paper trails, thereby providing better transparency, legibility, portability, usability, and asserts some data quality guarantees depending on implementation [37,38,39].

Health information systems are usable by healthcare professionals. This involves those directly working with patients. Healthcare practitioners gather and compile data to make healthcare decisions for patients, client groups, and the general public.

Health information systems include:

- **Electronic Health Record:** EHR is a structured digital collection of patient and community health information. These documents can be spread through healthcare settings. Records are exchanged across network-connected, enterprise-wide, or other knowledge networks and exchanges. EHRs may include a variety of data, including demographics, medical history, prescription and allergies, immunization status, laboratory test results, radiology images, vital signs, personal statistics like age and weight, and/or billing information [40].
- **Electronic Medical Record:** an electronic medical record (EMR) is a subset system of electronic health record systems dedicated to clinical information only [37].
- **Personal Health Record:** PHR is a health record where the patient maintains his health records and other patient-related information. This contrasts with the more commonly used electronic medical record run by institutions (such as hospitals) and includes clinician data. A PHR offers a full and comprehensive description of an individual's medical history [41].
- **Hospital Management System:** the hospital management system (HMS) is an integrated software that handles different clinic workflows. It manages the smooth healthcare performance along with administrative, medical, legal, and financial control. That is a cornerstone of the successful operation of the healthcare facility [42].
- **Practice Management System:** PMS is a form of healthcare software that manages the day-to-day operations of a clinic, hospital, or medical cabins, such as scheduling appointments, billing, and other administrative tasks; it is a subset of the HMS [43].
- **Clinical Decision Support:** CDS offers expertise and patient-specific information to physicians, nurses, patients, or others, intelligently filtered, or delivered appropriately. CDS encompasses numerous methods to improve clinical workflow decision-making; CDSS is a vital topic in medical artificial intelligence [44].
- **Clinical Data Warehouse:** CDW is an important solution for achieving clinical stakeholders' goals by combining heterogeneous data sources into a central repository and discovering strategic clinical-related responses, thus supporting clinical decisions [45].

III. Meta-research

Meta-research is the study of research using research methods. Also known as "research on research", it aims to reduce waste and increase research quality in all fields and concerns itself with research efficiency improvement and the detection of bias, methodological flaws, and other errors and inefficiencies [1].

Table I: major themes covered by meta-research [1]

Area	Definition
Methods	Performing research, study design, methods, statistics, research synthesis, collaboration, and ethics
Reporting	Communicating research, reporting standards, study registration, disclosing conflicts of interest, information to patients, public, and policy-makers
Reproducibility	Verifying research, sharing data and methods, repeatability, replicability, reproducibility, and self-correction
Evaluation	Evaluating research, prepublication peer review, post-publication peer review, research funding criteria, and other means of assessing the scientific quality
Incentives	Rewarding research, promotion criteria, rewards, and penalties in research evaluation for individuals, teams, and institutions

1. Research limitations

Meta-research studies revealed many limitations on the research process; these limitations are known as frauds, flaws, and constraints; flaws are methodological faults that are mostly due to the lack of adequate training and are almost unintentional; frauds are usually deliberate and compromise the integrity; Constraints are challenges facing the whole process, which might result in increased expenses, time wastes and poor investment income (ROI) ratios [6,7,8,9]:

Frauds in research vary from [6,10,11]:

- Plagiarism.
- Deception.
- Attribution theft.
- Biases.
- Conflict of interest.
- Data and results truncation.
- Procrastination.

Flaws in research processes are [6,7,8,9,12]:

- Improper data collection.
- Improper data exploration methodology.
- Improper data analysis algorithms.
- Improper reporting and discussion materials.
- Improper report writing syntax, grammar, or styles.

Constraints in research are [1,7,8,9,11,13,14]:

- Lack of human resources.
- Feedback loop delay.
- Time and resource constraints.
- Financial costs.
- No or few data quality assertions.
- Data is not governed.
- Lack of process standardization.
- Sample sizes and data collection constraints.
- Impact is limited to minorities.
- Language constraints.
- Research quality.
- Unexpected research outcome or poor outcome.

2. Medical research

Medical research (or biomedical research), also known as experimental medicine, covers a broad variety of studies, ranging from fundamental research — including fundamental scientific principles that may relate to preclinical understanding — to clinical research involving studies of patients who are subjects in clinical trials [46].

Medical research often involves fundamental sciences such as mathematics, physics, chemistry, and philosophy during its lifecycle and as a paradigm.

Medical research is either interventional or non-interventional; Research is said to be interventional if it interferes with patient management or requires an additional or unusual monitoring or diagnostic procedure. Interventional research involves biomedical research and healthcare interventions; as of non-interventional research, it often applies statistical analysis and inference in the form of prospective, retrospective, and clinical essay trials [47].

Each type of typologies is conducted upon data collected or centered on patients, which puts data collection and the quality of data at the utmost priority of every medical research; it also determines the quality of the research, and how long it would take to complete, as well as how much it would cost (financial spending) [48].

3. Research lifecycle

The research lifecycle is the steps taken in a research from inception to completion. Research data management is involved in each step of the research process [2]:

- Provision of good practice advocacy in the creation of a research methodology.
- Creation of data processing plans and related preparation.
- Efficient data generation and processing in research workflows.
- Appropriate documentation of analysis techniques used to interpret data.
- Promoting organized data curation and dumping in a repository or file store via institutional policy creation.
- The writing and publication of an article in an appropriate journal.
- Focusing on data publishing and reuse to assign credit and attribution to those involved.



Figure 1: research lifecycle by step [2]

Automation can be implemented across many research processes; it starts by ensuring data quality and data management, data warehouses can be developed to enforce these guarantees, additionally data discovery and report publishing can be streamlined as well [15,16].

Research aspects that can be automated [15,16]:

- **Data quality:** data quality assertions like its validity, veracity, variety, etc.
- **Data governance:** access and data management frameworks.
- **Headline suggestion:** research subject suggesting and literature research.
- **Data exploration:** data exploration before data analysis.
- **Data querying:** data analysis and query execution.
- **Reports writing:** the writing of ready to publish research reports.
- **Citations management:** automated annotations, citations quality assessment ... etc.
- **Research quality:** research quality assessment, writing style assessment, critical reviewing of researches.
- **Peer reviewing:** evaluating research by one or more people with similar competencies as the producers of the work.

4. Data lifecycle

The data lifecycle represents all data stages through its life, from its conception for research to distribution and reuse. The data lifecycle starts with a developer designing a study concept; once a study concept is created, the study collects data. After collecting data, it is stored for other end-users for distribution to be and used later. Data discovery leads to data repurposing, by creating a continuous loop back to the data processing stage where the repurposed data are archived and distributed for discovery [3,4,5].

Data protection and privacy are essential aspects of data governance and are subject to local laws and regulations. They must be enforced in-depth (In transit, at rest, and at use) whenever possible [49].

The data lifecycle overlaps in its steps with the research lifecycle for various reasons [50,51]:

- Research does make use of data iteratively and exploit data for research.
- Data is considered the starting point of any work, and the purpose of any study is to infer and summarize new data.
- Research transforms data to draw conclusions.
- Quality data and quick data retrieval and transformation are essential to accelerated quality

research iterations.

- Automated data solutions can result in shorter research lifecycles.
- Continuous/automated research pipelines require automated data solutions.

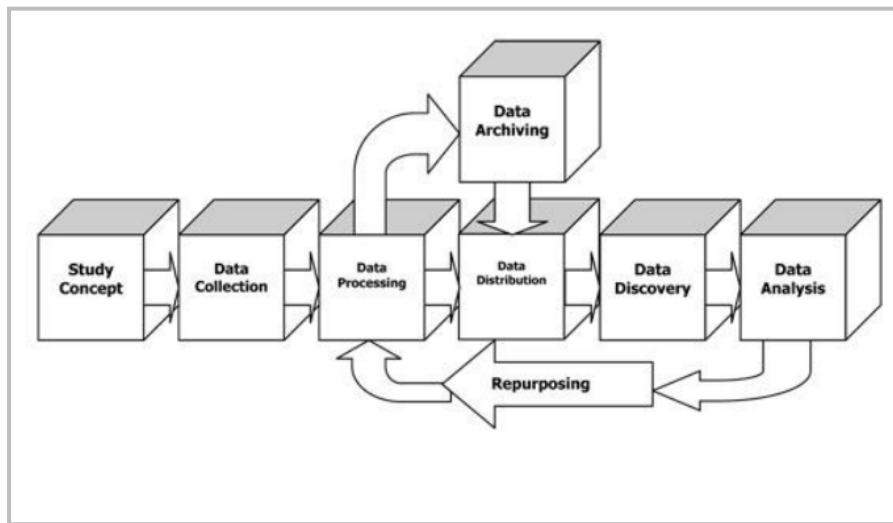


Figure 2: data lifecycle steps [52]

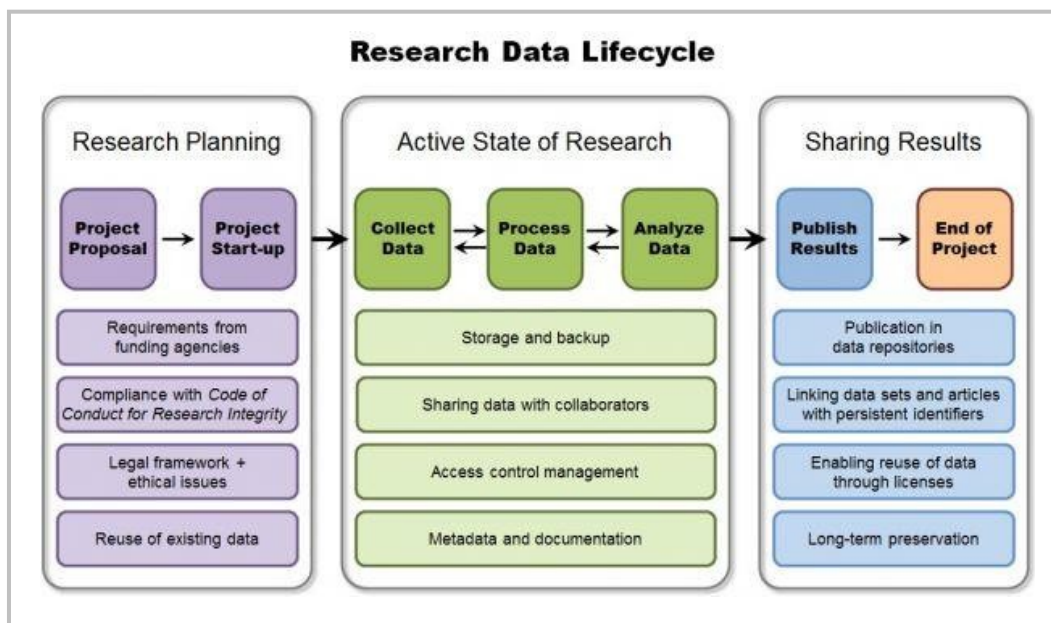


Figure 3: research data lifecycle [53]

Materials and methods

I. Methodology

1. Software development lifecycle

Software is a set of instructions that tell the machine how to function. This contrasts with the physical hardware on which the machine is designed and actually operates [54,55].

Software development lifecycle, also known as systems/application development life cycle (SDLC), is a process for planning, creating, testing, and deploying an information system. There are usually six stages in this cycle: planning, analysis, design, implementation, testing & integration, and maintenance [56]; Software development lifecycle dictated our methodology.

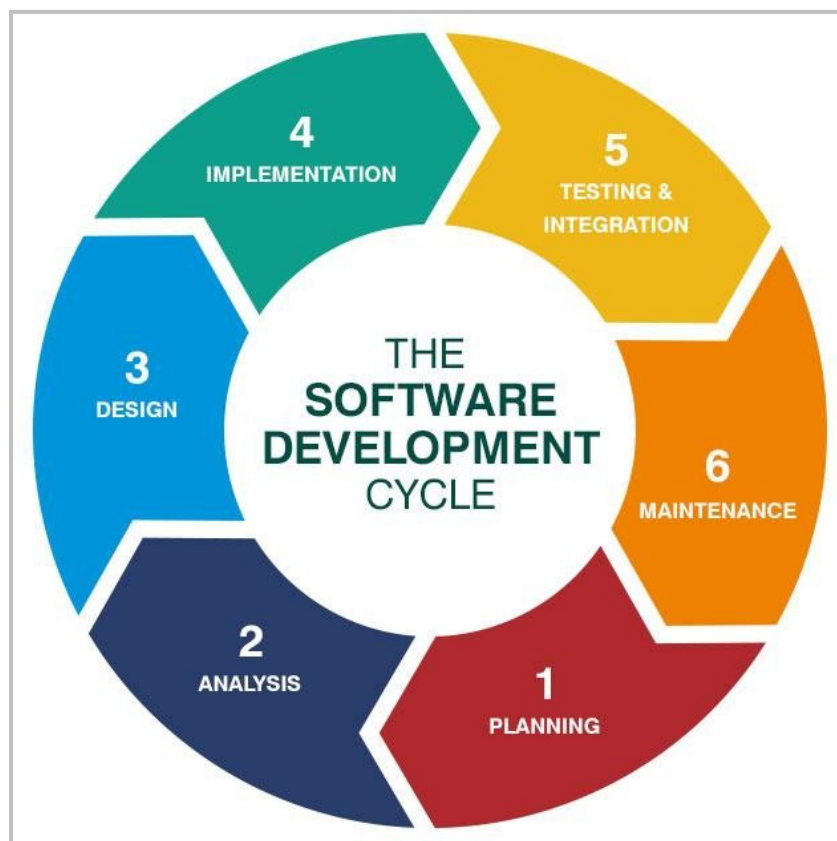


Figure 4: software development lifecycle (SDLC) [56]

2. Agile methodology

Numerous software development methodologies exist: Agile, DevOps, Waterfall, Rapid, Prototype, Dynamic System, Spiral, Feature driven, Joint Application, Lean Development... etc.

We've chosen the Agile methodology for the benefits it promises to deliver and for the advantages it provides over the other methodologies.

In software development, Agile approaches are specifications for development and solutions related to the collective effort of self-organizing, cross-functional teams and their stakeholders. It promotes adaptive planning , evolutionary development, early delivery and quality improvement, encouraging flexible responses to change [57,58].

Agile software development advocates these principles and values, including [58,59]:

- End-user satisfaction by early and continuous delivery of valuable software.
- Welcome changing requirements, even in late development.
- Deliver working software frequently (weeks rather than months).
- Close, daily cooperation between end-users and developers.
- Projects are built around motivated individuals, who should be trusted.
- Face-to-face conversation is the best form of communication (co-location).
- Working software is the primary measure of progress.
- Sustainable development, able to maintain a constant pace.
- Continuous attention to technical excellence and good design.
- Simplicity—the art of maximizing the amount of work not done—is essential.
- Best architectures, requirements, and designs emerge from self-organizing teams.

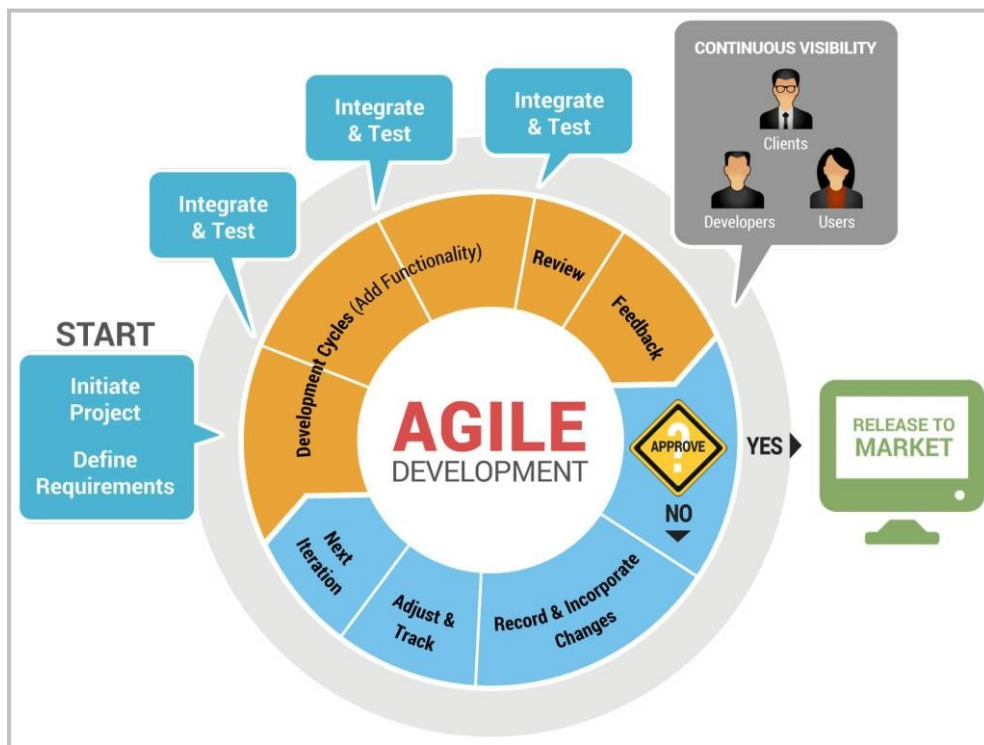


Figure 5: Agile software development methodology [57,58]

3. Scrum framework

Scholars consider Agile methodology to be mindset of values and principles rather than a methodology process; thereby many Agile methods or process exist with each has its own focus, scale, and application domain [60,61,62].

We've chosen the Scrum framework because it is a lightweight, simple to understand framework, and because it fits for our scale, focus on people and outcome, and advocates adaptive planification [63].

Scrum is a process framework used to manage product development and other knowledge work with people as the core focus, it consist of three roles: Product Owner, Development Team, and Scrum Master; three artifacts: Product Backlog, Sprint Backlog and Increment; five events: Sprint, Sprint Planning, Daily Scrum, Sprint Review, and Sprint Retrospective [63] (see Appendix 1).

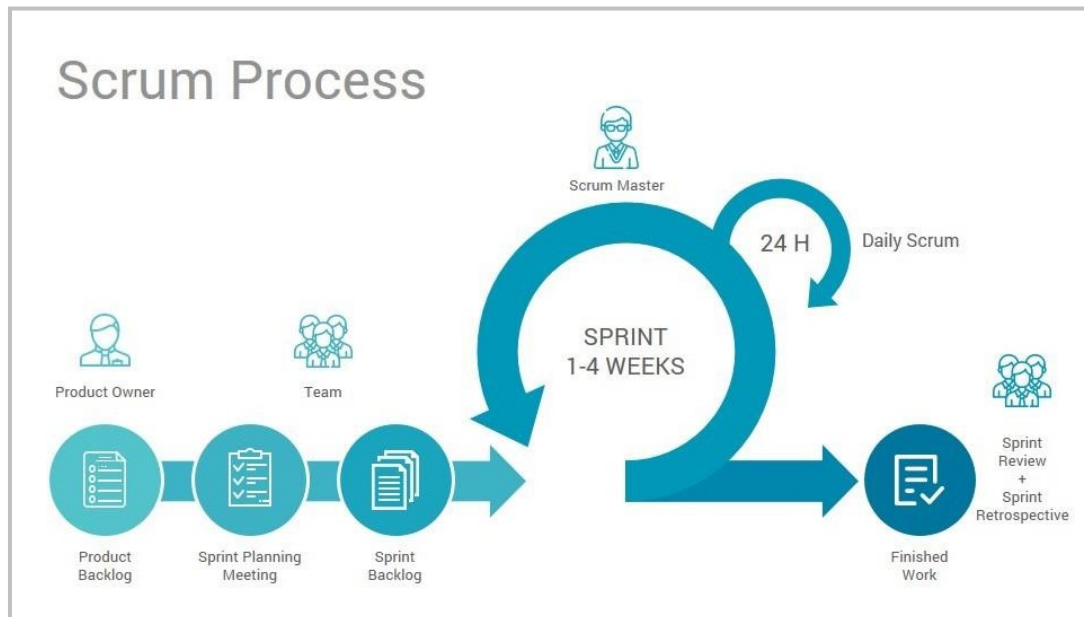


Figure 6: Scrum framework [63]

II. Workflow

1. Artifacts

A. Product backlog

The Product Backlog is an ordered list of prioritized features that are needed as a part of the product (see Appendix 1).

Backlog items (or User Stories) were constructed from ideas, features, requirements, functions or enhancements, each item has a description, value, class, and an order (see Appendix 1).

Stories were classified by their impact and their progress status; semantic impact inspired by Semver (semantic versioning) was used.

Semantic impact classifies stories by their consequences on backward compatibility; backward breaking changes are considered as major changes; backward compatible changes are considered as minor changes, whereas changes that do not modify the interface are considered as patch changes; since the project was a work-in-progress (WIP), another class has also been introduced for refactoring changes [64].

Semantic impact classification has many advantages, such as its being interoperable with semantic versioning (which is the leading standard of versioning in software engineering) [64]. With the use of semantic commit messages; Changelogs can be automatically generated, and version numbers can be bumped automatically.

Each story is assigned a story points (value) that estimates its development cost and time complexity, so each Sprint can comprise no more than a threshold of story points.

B. Sprint backlog

The Sprint Backlog is a list of tasks to be done during a single sprint. These items are picked from the Product Backlog during the Sprint Planning Meeting based on the priorities set by the Product Owner and the team's vision (see Appendix 1).

C. Increment

The Product Increment is the summation of overall backlog items finished during the Sprint and the previous completed Sprints (see Appendix 1).

2. Events

A. Sprint

The heart of Scrum is a Sprint, a consistent iterative time-box during which a "Done" releasable product Increment is created (see Appendix 1); In our workflow sprints were timeboxed as a one-week duration.

B. Sprint planning

The work to be performed in the Sprint is planned at the Sprint Planning by the collaborative work of the entire Scrum Team (see Appendix 1).

In our workflow, sprints were planned at the beginning of every sprint; while the next sprint items are being organized during the current sprint, items from the Product Backlog gets moved constantly to the next Sprint backlog.

C. Daily scrum

The Daily Scrum is a daily 15-minute time-boxed event for the Development Team during which the team plans work for the next 24 hours, inspects the current day work and communicates any roadblocks (see Appendix 1).

D. Sprint review

A Sprint Review is held at the end of the Sprint to inspect the Increment and adapt the Product Backlog if needed. During the Sprint Review, the Scrum Team and stakeholders collaborate about what was done in the Sprint and the next things that could be done to optimize the workflow (see Appendix 1).

E. Sprint retrospective

The Sprint Retrospective is an opportunity for the Scrum Team to inspect itself and create a plan for improvements to be enacted during the next Sprint (see Appendix 1).

3. Roles

A. Product owner

This role was played both by the thesis laureate and the supervisor; The product owner is a person(s) responsible for working with the development team to decide the features to be included under the product (see Appendix 1);

B. Development team

This role was ensured by the thesis laureate; The Development Team consists of persons who do the actual implementation work of the features, requirements or enhancements described in the Sprint Backlog (see Appendix 1);

C. Scrum master

This role was ensured by the thesis laureate; The Scrum Master is responsible for promoting and supporting by helping everyone understand Scrum theory, practices, rules, and values (see Appendix 1);

4. Software testing

Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test [65].

As the number of possible tests for even simple software components is practically infinite, all software testing uses some strategy to select feasible tests for the available time and resources [54,66,67].

There are at least three testing levels: unit testing, integration testing, and system testing. However, developers may provide a fourth level, acceptance testing [54,66,67].

A. Unit testing

Unit testing refers to tests that verify a particular code operation, usually at the function level, it involves a coordinated implementation of defect prevention and detection techniques to minimize software development risks, time, and expense [68,69].

Unit testing seeks to remove building errors before code is promoted for additional testing; this approach aims to enhance the consistency of the resulting software and the overall development process [68,69].

The software was developed following a test-driven development approach, Mocha and Istanbul are two major development dependencies, acting as framework and runner respectively. Targeted unit tests extensively tested each piece of functionality; used methods include fuzzy testing, behavior testing, static testing, fault injection, etc.

External dependencies were selected by the code-coverage metric; only software with code coverage above 95% and that is battle-tested is retained.

B. Integration testing

Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design [69].

The system components and interfaces were thoroughly tested via integration tests; testing techniques include mocking, stubbing, and snapshot testing.

C. Acceptance testing

Acceptance testing serves as a final check of the system 's efficiency and proper operation, emulating real-world conditions. If the program works as required and without problems during daily use, the same degree of stability could be reasonably extrapolated in production [69,70].

Our acceptance testing objective was to ensure that all HMS and EHR components work properly, the EHR system captures all medical record data, all EHR and HMS data are exploitable and the ability to construct arbitrary real-time data queries (continuous clinical research pipelines).

III. Considerations

1. Standardization

It corresponds to the precise definition of the semantic categories of medical language, to the organization of terms within each category and to the precise definition of each term. Three main international classifications exist: the ICD (International Classification of Diseases), SNOMED (Systematized Nomenclature of Medicine), and MeSH (Medical Subject Headings) [17].

The classification effort has mainly focused on establishing lists of terms. A particular effort must be made to standardize the linking elements of a sentence (verbs, conjunctions) and to the treatment of adjectives [17].

2. Structuring

Like the paper record, electronic records can be oriented according to the source, the problem, the chronology, with structures more suited to particular uses than others (general medicine, specialized file, etc.).

But the peculiarity of computerized records is that the techniques systems now allow more complex structures to be considered, making a clear distinction between the model of the internal representation of data (its deep structure, or the structure of the database) and the view (s) that users may have of this data (the different surface structures or user interfaces). The theoretical objective is then to design the deepest structure as general as possible and to establish bridges between the surface structures seen by the user (for example, structure can be oriented according to the source or according to the problems) [17].

Medical data can be entered in three ways: either as free text or as precise forms and questionnaires or a combination of both. The first case offers great ease and freedom of expression and description. Still, it does not make it possible to benefit from the possibilities of computerization, such as the speed and efficiency of entering, classifying, and the use of

accumulated data. On the contrary, hyper-structured records allow better efficiency and reliability of the data collected but can be less flexible and less accepted by practitioners. Their use remains limited to the specialized record of specific problems or pathologies (followed by diabetes, alcoholism, dialysis, etc.) [17,31].

1. Software quality

In this project, software quality was a major focus; the essence of the development process was Domain-driven (DDD), and Test-driven (TDD), a continuous integration (CI) pipeline was put in place, tests were automated, the code was statistically reviewed, code style was enforced thanks to linters, code parts were subdivided into modules and were correctly bundled and compacted.

Code quality was a major concern during development. We have chosen Typescript language as the primary programming language due to its type-checking capabilities; Code was linted by TSLint from code smells and style inconsistencies.

Checklists for software reliability, efficiency, security, and maintainability were made and where been adequately assessed (see Appendix 2).

2. Security

To ensure patient privacy and data security, compliance with local legislation is mandatory. Data used in clinical research must be anonymized and access regulated, data must be secured from breaches and security measures must be enforced in-depth, for data in transit, at rest, and in use [71] (see Appendix 3).

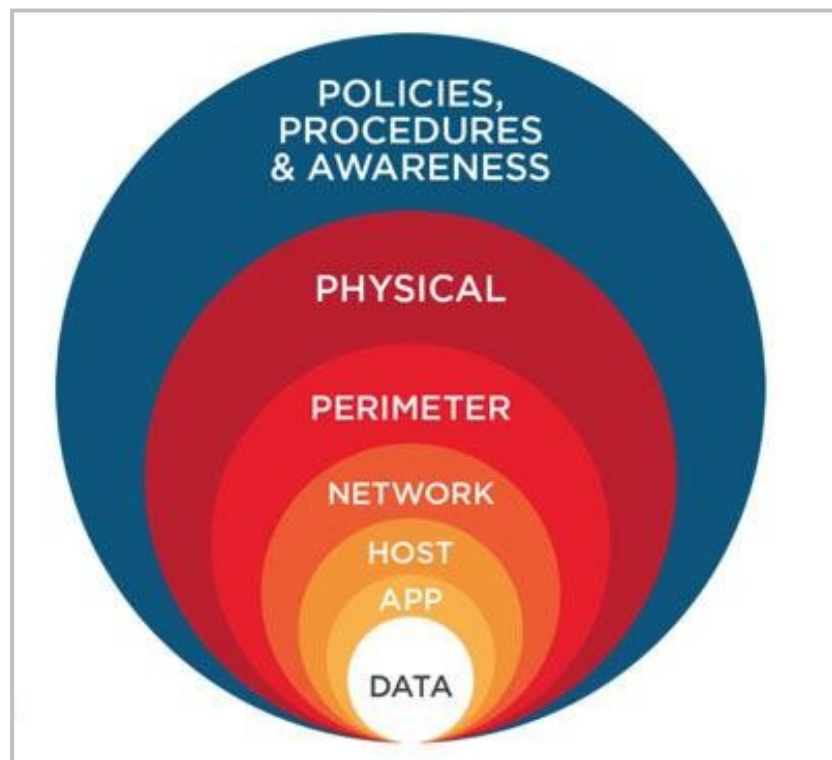


Figure 7: multilayered defense in depth [71]

3. Legislation and ethics

In Morocco, the medical record is an ethical obligation cited in Articles 22, 24 and 60 of the Code of Physicians and in Article 44 of the Code of Dental Surgeons. It became legal with the promulgation of Law 65-00 on compulsory health insurance. However, no law specifies the modalities of its holding, nor the sanctions applied in the event of derogation; Other related regulations are the internal regulations of hospitals in Articles 137 and 140 which asserts ownership and responsibility (see Appendix 4).

In contrast, Moroccan regulations appear to be shallow compared to the French counterpart which elaborates more on record types and components, data privacy, access and governance, and archiving

Many local laws and regulations related to the protection of personal data exist, mainly the 09-08 laws (published in 2009) regarding the protection of personal data and its related regulations [72,73] (see Appendix 4).

The Moroccan regulations compared to the international regulations appears to be sufficiently covering; in contrast with the HIPAA (Health Insurance Portability and Accountability Act of the U.S.), the HIPAA regulation is more concise on accountability as it defines penalties and measures such as the implementation of audit trails, assignment of data privacy officers, governed access control [74,75] (see Appendix 4).

4. User experience

User Experience (UX) is a person's perceptions and attitudes towards using a product, system, or service. It covers the practical, experiential, affective and substantive dimensions of human-computer interaction. It also involves a person's understanding of system aspects such as utility, accessibility and efficiency [76]; Several user experience principles were taken into consideration [77,78] (see Appendix 5).

Results

I. Architecture

Visualizing a software architecture and decomposing it into components can be tedious; the C4 model is the most commonly used model to visualize hierarchies in software engineering [79,80].

The C4 model is an "abstraction-first" approach to software architecture diagramming, it illustrates how software architects and developers think and construct software. The few abstractions and diagram types make the C4 model easy to learn and use [80].

1. Level 1: System diagram

A system context diagram is a good starting point for diagramming and illustrating a software system, visualizing the overall system, users and surrounding contexts [80].

- **Core context:** universal content management system (CMS), which theoretically can be tailored for any need to accommodate any schema.
- **Major contexts:** include hospital management system, practice management system, electronic health record, and clinical data warehouse contexts.
- **Person:** the intended audience is the whole medical and para-medical staff.
- **Technology:** the technologies used are mainly web technologies; following the latest trends, we used Typescript – a statically type-checked superset of JavaScript – in both backend (through NodeJS) and frontend development, NPM (Node Package Manager) was used to manage external dependencies (see Technology Stack chapter).

Table II: major application contexts

Context	Description
HMS context	The system must offer HMS capabilities such as HR management, procedure management, and other administrative procedures
PMS context	Users of the system and physicians must be able to carry their practice routines
EHR context	Patient charts, procedures, interventions, prescriptions, and all their admission data must be recorded and displayed in highly accessible interfaces and/or format
CDW context	Clinical data must be centralized, and its quality assessed, governed, and easily accessed through secure channels, and can be transformed and queried arbitrarily in real-time

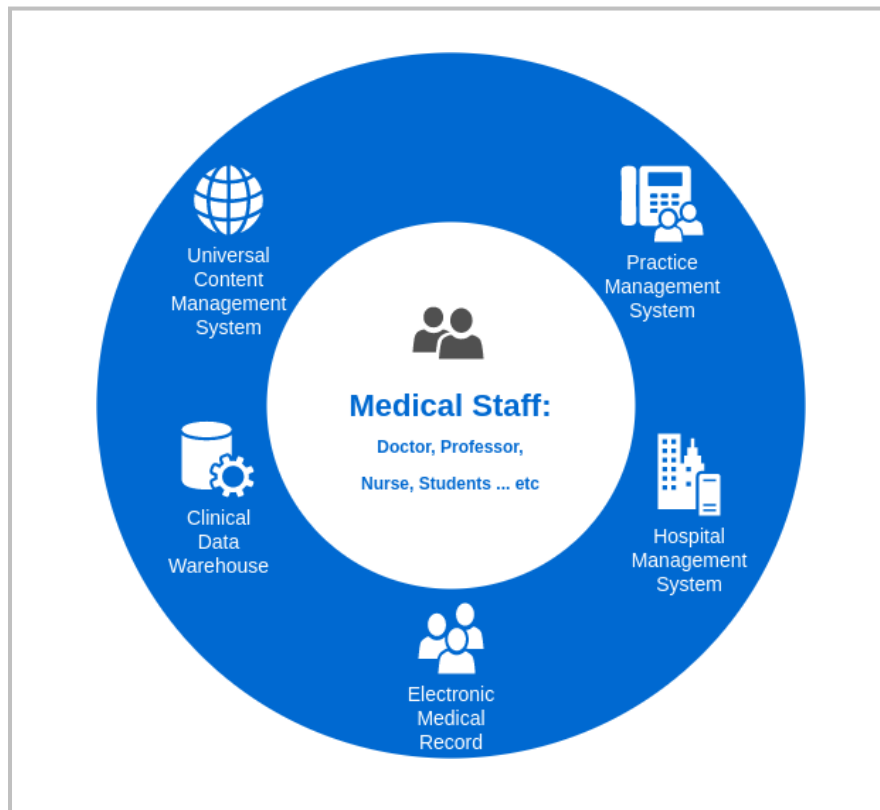


Figure 8: level 1 c4 diagram applied to our system

2. Level 2: container diagram

The container diagram illustrates the architecture's high-level shape and how tasks are spread through it. It also demonstrates the major technology choices and how containers communicate altogether. It is a simple, high-tech diagram, useful for software developers and support / operation employees alike [80].

A. Content container

This container is a multifaceted container with many interlaced components; It was built bottom-up with some critical considerations in mind:

- **Scheme elasticity:** accomplished through dynamic expressions, preconditions, and fault tolerance.
- **Fault tolerance:** to ensure backward and forward compatibility on scheme changes.
- **Record actions:** allow create, update, read, and delete actions to be carried on single and bulk records.
- **Query variants:** display custom variant views of the same scheme records.
- **Type/Expression visual inference:** display inputs and visual representation of datatype, customize inference through visual hints.
- **Custom listing views:** customize listing displays, including element positioning, view types, and item card customization.
- **Print everything:** enable two appearance variants for each scheme part, and enable scheme

records to be printed with ease.

Key container components are ordered by the level of abstraction, incrementally:

- **Segment:** is the cornerstone unit of schemes; each segment is defined by its key/name pairs mapping, it has at least one property, segments are stored as key–value maps.
- **Scheme:** is the next level of abstraction; a scheme can be a list of segments/schemes, or a map of segments/schemes, or a relation scheme of a scheme.
- **Action:** one type of CRUD action can be mapped as one action with read–only keys, editable keys, and fallback mappings.
- **Variant:** is a how scheme records can be queried, where results can be limited, filtered, transformed, or sorted.
- **Element:** is a single editable, single–action, single–scheme record, can be used to map configurations or to create calculator interfaces.
- **View:** is a custom mapping between variants into a single display, with custom element positioning, custom record card configurations, and custom media mappings.
- **Interface:** they are used to group views and items into custom menus.

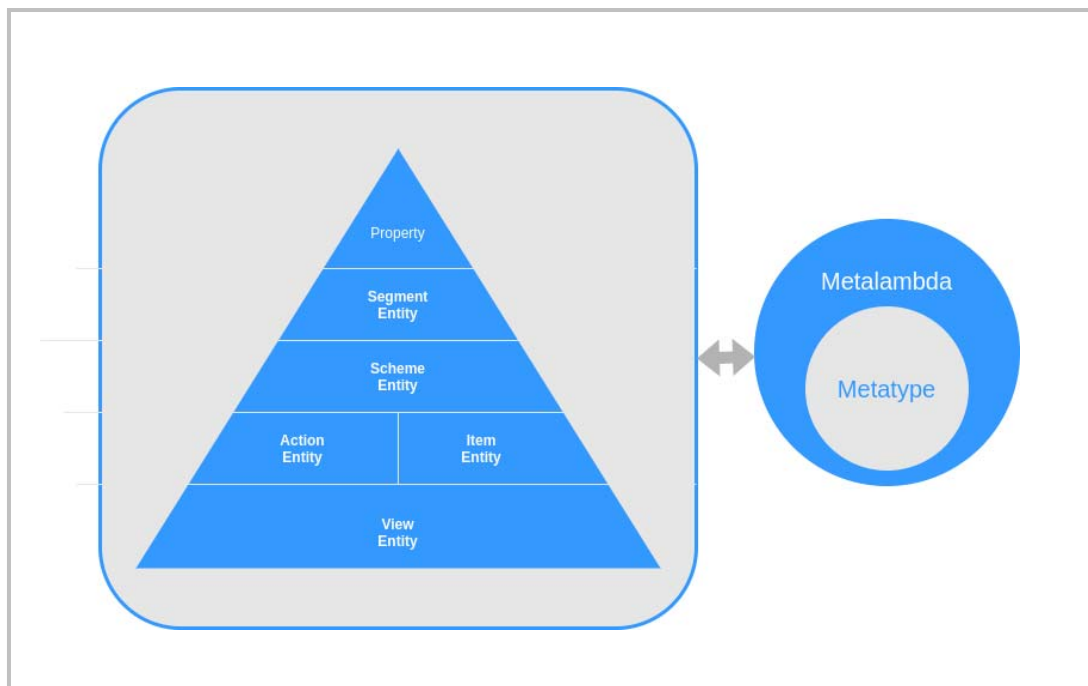


Figure 9: level 1 container c4 Model diagram for content container

B. Ontology container

Ordinary scheme expressions and datatypes are not enough to warrant all data quality promises, primarily due to free-entry drawbacks, in particular:

Table III: data quality guarantees missing on free-entry input systems

Attribute	Description
Semantic	Ensuring each data concept has a semantic meaning, and its semantic meaning is correct
Hierarchical	Ensuring that hierarchical data is represented in such a manner and its hierarchy is respected
Encoding	Ensuring that encodable data are correctly encoded and can be correctly decoded if supported
Variety	Support of similar data varieties encodings
Veracity	Ensuring data correctness

The ontology container comes in place to solve those problems providing a single source of truth for an auto-encodable hierarchical, semantic concepts tree.

Keys features of the ontology container are [81]:

- High-performance in-memory representation of the whole concepts tree.
- Realtime ontology lookup and traversal using a graph data structure that allows both depth and breadth-first search.
- Concepts are automatically sorted alphabetically.
- Concepts are auto-encoded as a digital prefix tree (Trie), which allows both retrieval and traversal in $O(1)$ – constant time.
- Concept codes are semantic and hierarchic, thanks to the Trie data structure.
- Concept lookups are efficiently cached and memoized.

The ontology container offers useful properties in runtime:

- Traversal and retrieving of parent and children relations for a concept.
- Grouping, categorization of ontological concepts by ancestors.
- Encoding and decoding of concepts codes and concept labels and relations.
- Ability to exclude parenting or sibling relationship between two concepts in constant time thanks to the probabilistic nature of the Trie data structure using prefix and suffix binary

comparison.

- Ability to create and carry complex ontology aggregation and grouping transformations.
- Data compression and minimal storage capacity consumption by storing only concept codes which are space efficient small strings.
- Ensure a free-entry like experience while asserting variety, semantic, hierarchy and organization quality attributes.

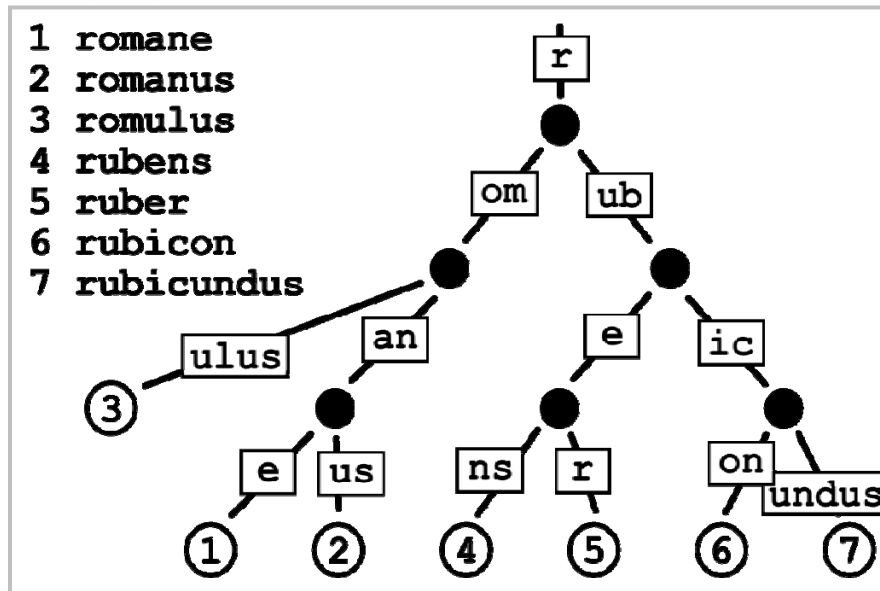


Figure 10: Trie data structure (digital prefix tree) [81]

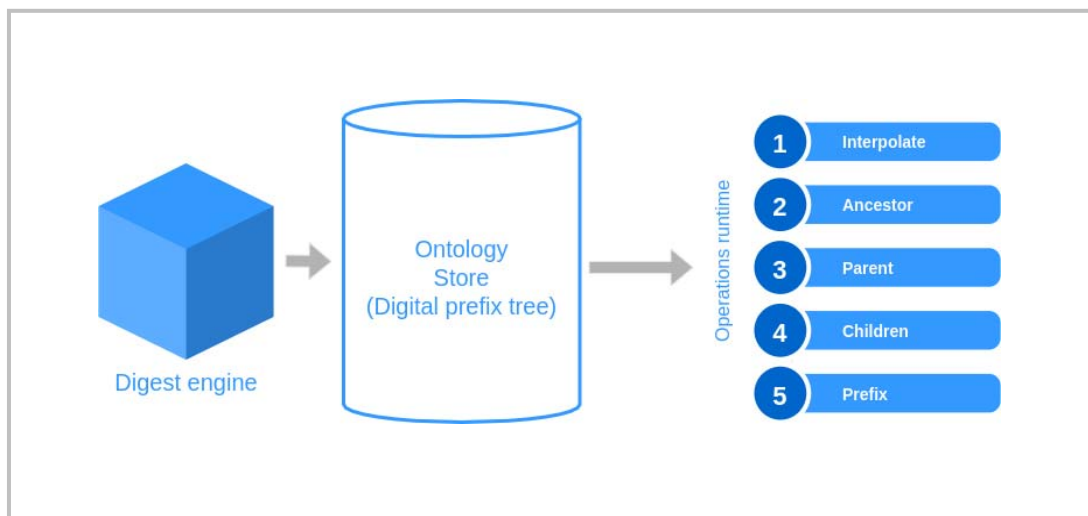


Figure 11: ontology container C4 Model level 2 component-level diagram

C. Datatype container

A data type or simply type is a data attribute that tells the compiler or interpreter how to store, retrieve and use data [82].

Datatypes are an essential concept in our application; datatypes are the cornerstone of expressions, segments, and conditions.

Datatype engine was supplied by *metatype*, which is a proprietary datatype engine that parses datatypes, validates data against datatypes, generate fake realistic data based on datatypes, generate empty deterministic placeholders, can be used to infer and apply arithmetic on datatypes, among other features.

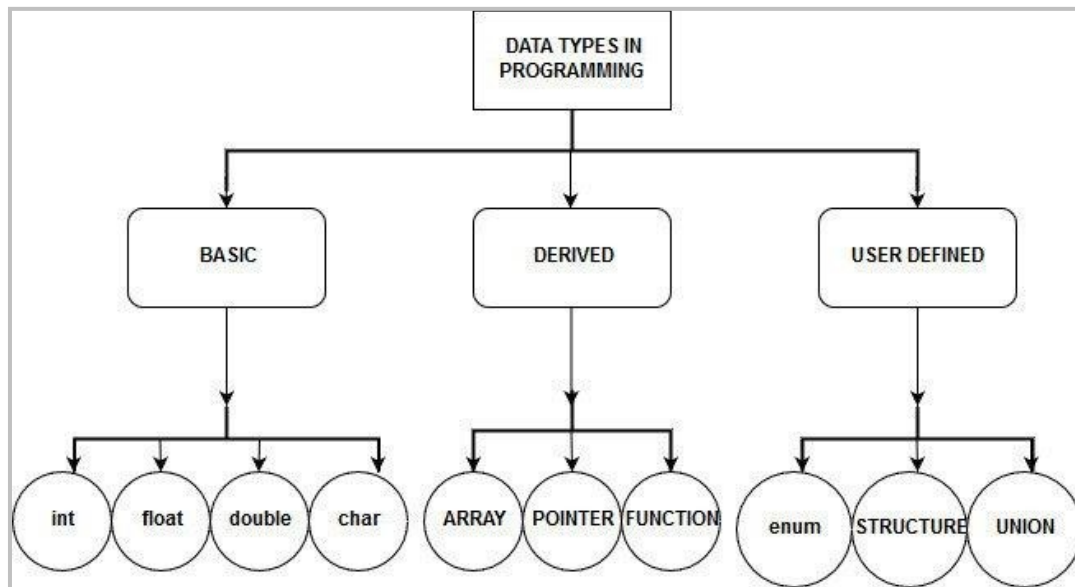


Figure 12: basic datatypes in programming

It comes with native support for 22 scalar types, with support for compound types such as arrays and objects, and merge types such as union and intersection.

Scalar types can be extended with custom attributes and pattern names and flags with over 40 attributes and 60 patterns supported.

For example, to define a datatype that matches emails strings between 10 and 60 characters, one can use *"string(min=10,max=60) as email"*.

It supports type coercion, normalization, sanitization, synonym down-sampling, range trimming, and fault tolerance through custom tolerance flags.

Type coercion stands for type conversion capability on inter-coercible types, such as the coercion of a string of digits into an integer [83].

Data normalization or canonicalization is the operation of normalizing similar data into a lookalike pattern such as lower-casing all letters; normalization can be contextual, e.g., name cases must not be normalized, whereas usernames must be all lower-cased [84].

Synonym downsampling is the process of normalizing synonyms of the same concept into one concept, e.g., “true,” “yes,” “on” and “valid” would be all down-sampled into TRUE Boolean value.

Sanitization is the process of stripping out unwanted characters to clean up data, e.g., removing white-spaces out passwords, emails, or usernames [85].

Type arithmetic consist of union and intersection merges and subset and superset checks.

Also, it supports file manipulation, including file trans-coding, file validation, and asynchronous file transformation and coercion of image, video, audio, and document files.

Every feature of *metatype* has been concisely used in our application; for example, type arithmetic and type checking were used in the inference engine and the validation engine respectively of the Expression container. File manipulation capabilities of *metatype* were mandatory to allow file management.

metatype also allows access to datatype metadata from raw parsed data; this property is effective in generating input and form interface to ensure total synchrony with the datatype counterpart in real-time.

D. Expression container

The expressions container is provided by *metalambda*, which is a proprietary general-purpose expression evaluation engine.

metalambda is powered both by *metaparser* and *metatype*; it is a procedural programming language, it supports all major arithmetic operators (summation, subdivision, multiplication, modulus, exponent) in addition to string concatenation operator, it supports piping, conditions, type guards, and typecasting, it comes with hundreds of utility functions for string, number, date, duration manipulation in addition to math utilities.

Expressions have useful properties:

- It can be used as dynamic fallback values.
- It can be used to infer the datatype of similar values.
- It can be statically analyzed for external variable dependencies.
- It can be used to infer the visual appearance of its input.

Expressions are used across the project, particularly in:

- Segment properties.
- Scheme/property preconditions.
- Variant queries.
- Action values.
- Data queries.

In theory, expressions can represent an unlimited combination of datatypes and, therefore, might be used to capture many datatypes.

E. Internationalization container

internationalization and localization, abbreviated i18n and l10n respectively, are means of adapting computer software to different languages. Internationalization the engineering process, in contrast Localization is the process of doing the actual translation [86].

Internationalization support was been implemented thanks to *metatext*, which is a proprietary i18n and l10n application ecosystem.

Internationalization capabilities are further enforced within the system through:

- The widespread use of entity keys and property tags.
- Dynamic extraction of keys for translation.
- Automated key translations.
- Double sources of translations, one for system and the other for content.
- The support of both LTR and RTL languages.
- Proper data encoding.
- Compressed just-in-time translation tags and templates delivery system.

F. Parsing container

Parsing, syntax analysis, or syntactic analysis is the process of analyzing a string of symbols, either in natural language, computer languages, or data structures, conforming to the rules of a formal grammar [87].

Parsing consists of many steps before obtaining the abstract syntax tree; these steps are tokenization, lexing, and parsing [88].

The parsing container is the core of every operation in our application; it is used to parse datatypes, expressions, schemes, and language templates.

Parsing syntax is an intensely repetitive task, and therefore it must be optimized, cached, and predictable.

Our parser is a top-down LL parser-combinators parser that syntactically analyzes expressions using custom grammar trees.

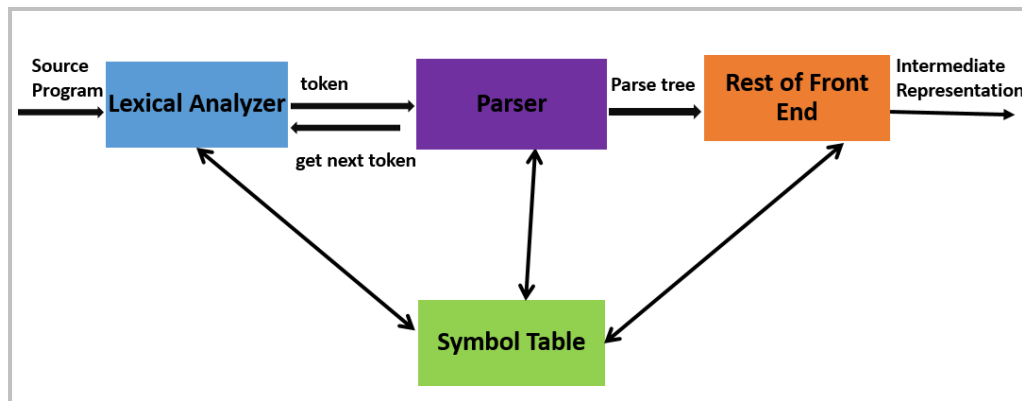


Figure 13: parsing container and its layers of abstraction

II. Technology stack

In computing, a technology/solution/software stack is a set of software subsystems or components needed to create a complete platform such that no additional software is needed to support applications [89].

In software engineering, the terms front end and back end apply to the separation of concerns between the presentation layer (front end) and the data access layer (back end) of the software [90].

1. Backend stack

The system is compiled as a Docker image that can run on any Docker container runtime, which allows the system to be run in virtually any operating system [91].

NodeJS serves as the application runtime environment, Node.js is an open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command-line tools and for server-side scripting programs [92].

Typescript is used to overload native JS capabilities; it is an open-source programming language developed and maintained by Microsoft. It is a strict syntactical superset of JavaScript and adds optional static typing to the language. TypeScript is designed for the development of large

applications and transcompiles to JavaScript [93].

ExpressJS was used as the server web framework for Node.js, released as free and open-source software under the MIT License [94].

YAML and JSON data serialization are the primary serialization framework used in the system. Numerous security frameworks were used in the back-end, such as JSON web token, Bcrypt, JWK, JWA, and JWS.

meta suite dependencies were used interchangeably in the backend and the frontend.

Table IV: *meta* suite dependencies

Dependency	Description
<i>metaparser</i>	Serves as the parser container, a LL, parser combinator, and context-free grammar parser
<i>metatype</i>	Datatype container, and engine for datatype parsing, checking, casting, faking, inference, arithmetic, and validation for primitives, blobs, compounds, and merges
<i>metalambda</i>	Serves as the expression's container, a procedural, strongly-typed, statically typed programming language
<i>metatext</i>	Complete internationalization, globalization and localization ecosystem and runtime
<i>metacontent</i>	Universal content management system and universal data management system provider

2. Frontend stack

The programming language used in front-end development is JavaScript overloaded with Typescript capabilities, *meta* suite was also heavily used.

LESS language was used at the styling layer; it is a dynamic preprocessor style sheet language that can be compiled into Cascading Style Sheets and run on the client-side or server-side. Designed by Alexis Sellier [95].

React was used as the presentation engine; it is an open-source JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual

developers and companies [96].

ApexCharts was used to render charting plots; it is a free and open-source modern charting library that helps developers to create beautiful and interactive visualizations for web pages [97].

3. Development stack

Several technologies were used in the development stack to govern the development process taking into account SDLC and security considerations.

Mocha was used to drive the TDD; it is a feature-rich JavaScript test framework running on Node.js and in the browser, making asynchronous testing simple and fun. tests run serially, allowing for flexible and accurate reporting while mapping uncaught exceptions to the correct test cases [98].

Istanbul was used to keep track of code coverage, it instruments ES5, and ES2015+ JavaScript code with line counters [99].

Nodemon was used as a live-reload server; it is a tool that helps develop Node.js based applications by automatically restarting the node application when file changes in the directory are detected [100].

Webpack was used as the module bundler; it is an open-source JavaScript module bundler. It is made primarily for JavaScript, but it can transform front-end assets such as HTML, CSS, and images if the corresponding loaders are included [101].

Typescript and TSLint enhance the development experience, assert static checking, and enforce coding style; it is an extensible static analysis tool that checks the TypeScript code for readability, maintainability, and functionality errors. It is widely supported across modern editors & build systems and can be customized with your own lint rules, configurations, and formatters [102].

metatext ecosystem was used to govern and manage the internationalization, globalization, and localization processes.

III. System containers

1. Electronic medical record

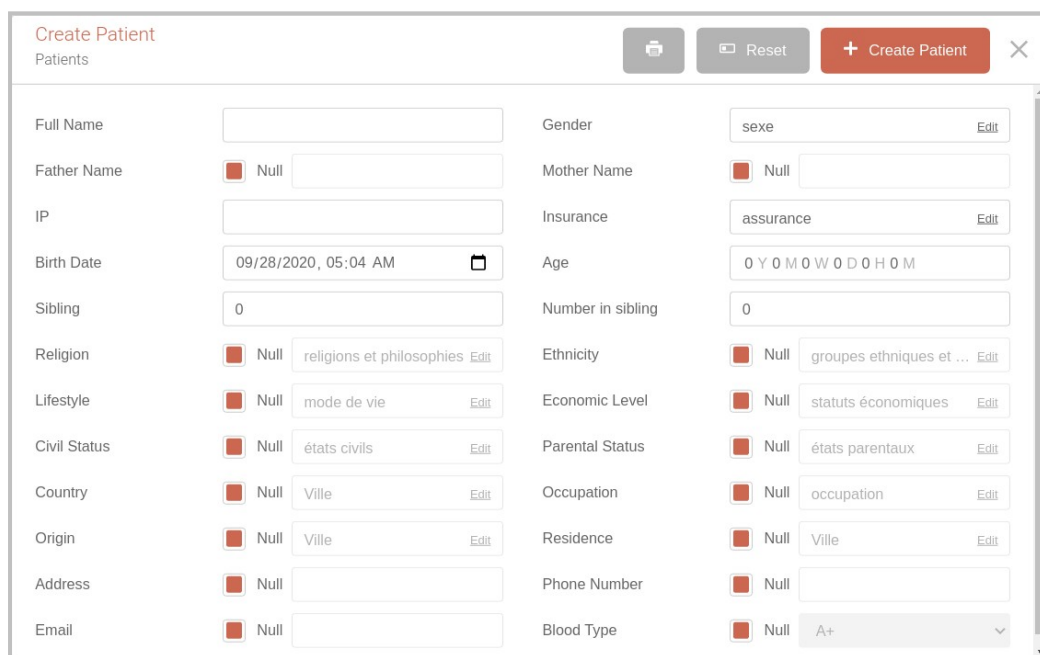
An electronic health record (EHR) is the systematized collection of patients records in digital format. These documents can be exchanged through healthcare environments [37,103].

A. Patient demographics

Patients demographics are also known as patient identity in the paper form counterpart; all relevant information must be collected to carry an upright examination.

Demographics include non-medical patient information. It includes information to locate the patient, including identifying numbers, addresses, and contact numbers. It contains information about race and religion as well as the workplace and professional occupation.

Every concept in the demographic's ontology is guaranteed to be complete and exhaustive at the time of writing this dissertation.



The screenshot shows a web-based form titled "Create Patient" with a sub-header "Patients". The form contains two columns of input fields. The left column includes: Full Name, Father Name (with a "Null" checkbox), IP, Birth Date (with a date picker showing "09/28/2020, 05:04 AM"), Sibling (with a value of "0"), Religion (with a "Null" checkbox and a dropdown menu showing "religions et philosophies"), Lifestyle (with a "Null" checkbox and a dropdown menu showing "mode de vie"), Civil Status (with a "Null" checkbox and a dropdown menu showing "états civils"), Country (with a "Null" checkbox and a dropdown menu showing "Ville"), Origin (with a "Null" checkbox and a dropdown menu showing "Ville"), Address (with a "Null" checkbox), and Email (with a "Null" checkbox). The right column includes: Gender (with a dropdown menu showing "sexe"), Mother Name (with a "Null" checkbox), Insurance (with a dropdown menu showing "assurance"), Age (with a dropdown menu showing "0 Y 0 M 0 W 0 D 0 H 0 M"), Number in sibling (with a value of "0"), Ethnicity (with a "Null" checkbox and a dropdown menu showing "groupes ethniques et ..."), Economic Level (with a "Null" checkbox and a dropdown menu showing "statuts économiques"), Parental Status (with a "Null" checkbox and a dropdown menu showing "états parentaux"), Occupation (with a "Null" checkbox and a dropdown menu showing "occupation"), Residence (with a "Null" checkbox and a dropdown menu showing "Ville"), Phone Number (with a "Null" checkbox), and Blood Type (with a "Null" checkbox and a dropdown menu showing "A+"). At the top right of the form are three buttons: a printer icon, a "Reset" button, and a "+ Create Patient" button. A close button (X) is also present.

Figure 14: captured patient demographic information

A click-to-select interface connects patient records with admission records; patient records can be created on request or retrieved and chosen.

Entity: Patient
Reference: 19
Created At: 9/6/20
Updated At: 06:01

Variant: Patients Action: Update Patient

Execute Choose

Figure 15: patient record relation with admission record

B. Admission status

Admission status is tracked by a dedicated component; parameters include date of admission, discharge date, chief complaint (symptom), cause (diagnosis), service (department), gender, bed number, IP reference and admission status.

Date: 06/13/2020, 06:09 PM

Discharge: ☐ Null 07/06/2020, 06:12 PM

Complaint: symptôme

Reason: insuffisance respiratoire du nouveau-né

Status: sortie, vivant

Service: service de reanimation neonatale

Gender: feminin

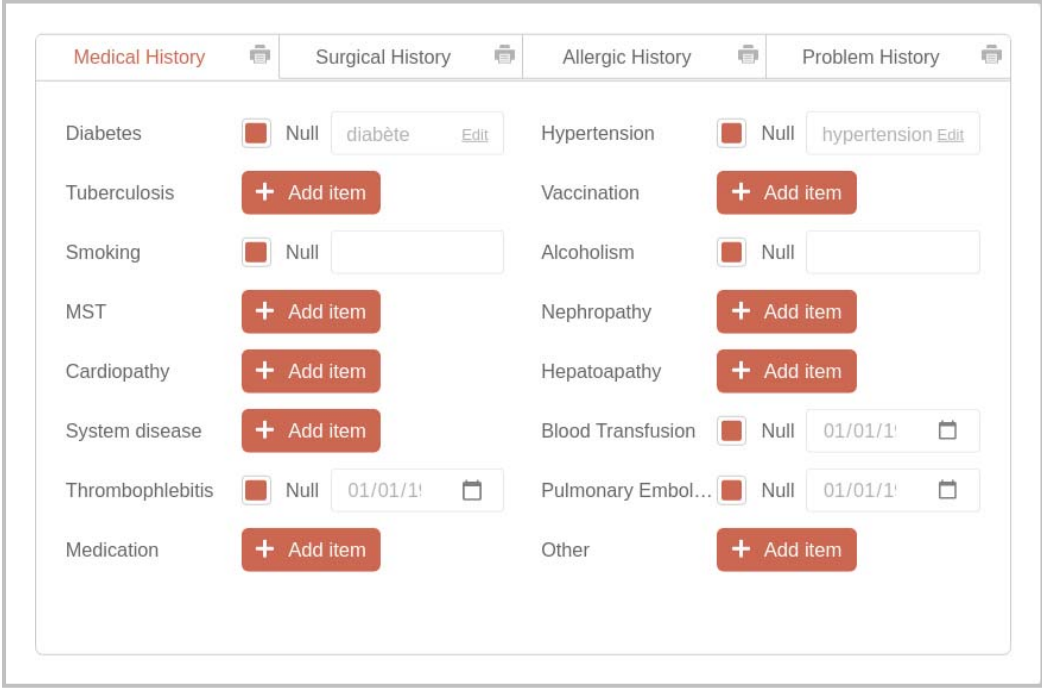
Bed: 7

Figure 16: admission status component

C. History component

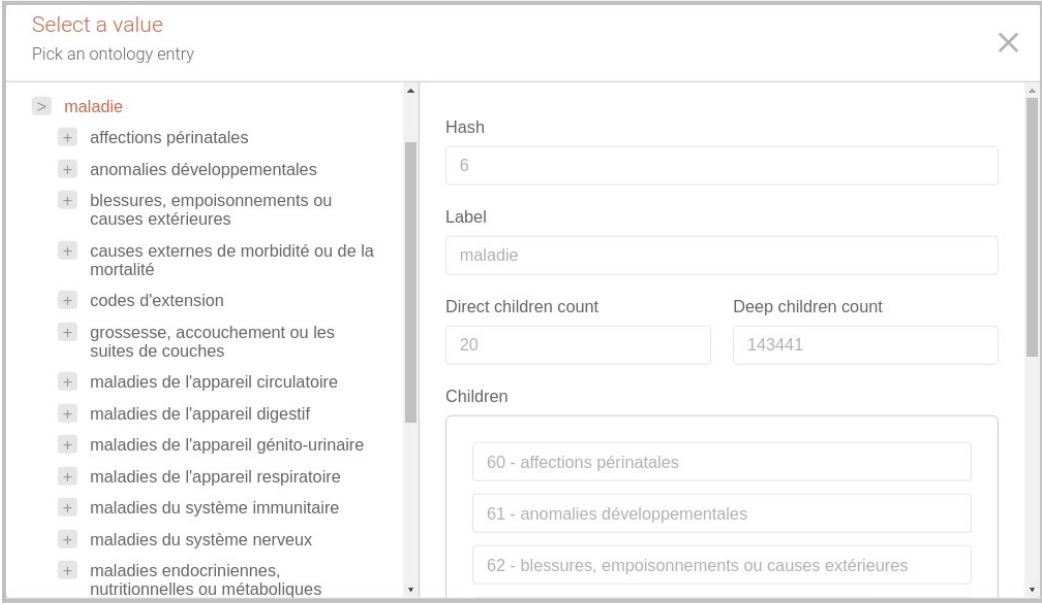
The medical history is a longitudinal record of what has happened to the patient since birth. It chronicles diseases, major and minor illnesses, as well as growth landmarks. It gives the clinician a feel for what has happened before to the patient.

The patient history scheme is a flexible scheme subject to the admission department, patient gender, and patient age.



The Medical History component is a web interface for managing patient medical records. It features four tabs: Medical History, Surgical History, Allergic History, and Problem History. The Medical History tab is active, showing a list of medical conditions and their status. Each condition has a checkbox, a status (Null or a value), and an Edit button. Some conditions have a red '+ Add item' button. The conditions listed are: Diabetes (Null, diabète, Edit), Tuberculosis (+ Add item), Smoking (Null, empty input), MST (+ Add item), Cardiopathy (+ Add item), System disease (+ Add item), Thrombophlebitis (Null, 01/01/1, calendar icon), Medication (+ Add item), Hypertension (Null, hypertension, Edit), Vaccination (+ Add item), Alcoholism (Null, empty input), Nephropathy (+ Add item), Hepatoapathy (+ Add item), Blood Transfusion (Null, 01/01/1, calendar icon), Pulmonary Embol... (Null, 01/01/1, calendar icon), and Other (+ Add item).

Figure 17: medical history component



The Disease ontology picker is a web interface for selecting a disease from an ontology. It has a title 'Select a value' and a subtitle 'Pick an ontology entry'. On the left, there is a list of ontology entries under the 'maladie' category, including: affections périnatales, anomalies développementales, blessures, empoisonnements ou causes extérieures, causes externes de morbidité ou de la mortalité, codes d'extension, grossesse, accouchement ou les suites de couches, maladies de l'appareil circulatoire, maladies de l'appareil digestif, maladies de l'appareil génito-urinaire, maladies de l'appareil respiratoire, maladies du système immunitaire, maladies du système nerveux, and maladies endocriniennes, nutritionnelles ou métaboliques. On the right, there is a form with fields for Hash (6), Label (maladie), Direct children count (20), Deep children count (143441), and a list of Children (60 - affections périnatales, 61 - anomalies développementales, 62 - blessures, empoisonnements ou causes extérieures).

Figure 18: disease ontology picker

Medical History	Surgical History	Allergic History	Problem History
Adenoidectomy	☒ Null 01/01/1!	Cholecystectomy	☒ Null 01/01/1!
Appendectomy	☒ Null 01/01/1!	Mastectomy	☒ Null 01/01/1!
Hysterectomy	☒ Null 01/01/1!	Colectomy	☒ Null 01/01/1!
Hernia	☒ Null 01/01/1!	Phlebotomy	☒ Null 01/01/1!
Splenectomy	☒ Null 01/01/1!	Other	+ Add item

Figure 19: surgical history component

Medical History	Surgical History	Allergic History	Problem History
Betalactamines	☐ Yes ☒ No	Cyclines	☐ Yes ☒ No
Macrolides	☐ Yes ☒ No	Aminosides	☐ Yes ☒ No
Gluten	☐ Yes ☒ No	Eggs	☐ Yes ☒ No
Peanut	☐ Yes ☒ No	Milk	☐ Yes ☒ No
Hymenoptera	☐ Yes ☒ No	Other	+ Add item

Figure 20: allergic history component

Obstetric History	Gynecologic History	Delivery History	Familial History
LMP	09/06/2019, 07:1	EDC	52 weeks 2 days
Followed Pregna...	☒ Yes ☐ No	Consanguinity	☐ Yes ☒ No
Gestation	2	Parity	2
FDIU	0	Miscarriages	0
Exceeded Term	0	Premature Delivery	0
Preterm Labour	1	PRM	0
Multiple Pregnancy	0	Preeclampsia	☐ Yes ☒ No
Sever Preeclamp...	☐ Yes ☒ No	Eclampsia	☐ Yes ☒ No
HELLP Syndrome	☐ Yes ☒ No	Ectopic Pregnancy	☐ Yes ☒ No

Figure 21: obstetric history component

Obstetric History	Gynecologic History	Delivery History	Familial History
LMP	09/06/2019, 07:14 PM		
EDC	52 weeks 2 days		
Age	21 years		
Menorrhagia duration	2 days 0 hour		
Cycle duration	28 days 0 hour		
Cycle regularity	régulier Edit		
Other	+ Add item		

Figure 22: gynecologic history component

The screenshot shows a web interface with four tabs: 'Obstetric History', 'Gynecologic History', 'Delivery History' (which is active and highlighted in red), and 'Familial History'. Each tab has a printer icon. The 'Delivery History' form contains the following fields:

- Date: 06/12/2020, 06:14 PM (with a calendar icon)
- Delivery Route: voie basse (with an 'Edit' link)
- Delivery Assistance: + Add item (with a red button)
- Presentation: sommet (with an 'Edit' link)
- Instructor: Gynécologue (with an 'Edit' link)
- Delivery Place: niveau 3 (with an 'Edit' link)
- Labour duration: 0 day 5 hours 0 minute
- Rupture of membranes: 0 day 1 hour 0 minute
- Amniotic fluid: + Add item (with a red button)

Figure 23: delivery history component

The screenshot shows the same web interface with the 'Familial History' tab active and highlighted in red. The form contains the following fields:

- Genetic Disease: ☒ Null (with a text input field)
- Cancer: ☒ Null (with a text input field)
- Hypertension: ☒ Null (with a text input field)
- Suicide: ☒ Null (with a text input field)
- Other: + Add item (with a red button)
- Asthma: ☒ Null (with a text input field)
- Diabetes: ☒ Null (with a text input field)
- Mental Disorder: ☒ Null (with a text input field)
- Similar: ☐ Yes ☒ No

Figure 24: family history component

The screenshot displays a web-based form titled "Problem History" within a tabbed interface. The tabs include "Medical History", "Surgical History", "Allergic History", and "Problem History". The "Problem History" tab is active. The form contains several input fields and buttons:

- Date:** A text field containing "09/28/2020, 05:14 AM" with a calendar icon on the right.
- Temporality:** A text field containing "temporalité" with an "Edit" link on the right.
- Severity:** A text field containing "gravité" with an "Edit" link on the right.
- Evolution:** A text field containing "évolution" with an "Edit" link on the right.
- Positive Signs:** A red button with a plus sign and the text "+ Add item".
- Negative Signs:** A red button with a plus sign and the text "+ Add item".
- Fever:** A red square checkbox followed by the text "Null" and an empty text field.
- Weight Loss:** A red square checkbox followed by the text "Null" and an empty text field.
- Asthenia:** Two radio buttons, one labeled "Yes" (unselected) and one labeled "No" (selected).

Figure 25: problem history component

D. Examination component

Clinical examination or system review (ROS), is a methodology used by healthcare providers to assess organs/systems to collect physical symptoms in contrast to subjective symptoms collected during the medical history [104].

Notice that clinical examination is a dynamic scheme subject and flexes to the attributes of admission department, patient gender, and patient age.

General	Pulmonary	Cardiac	Abdominal	Urologic	Neurologic	Articular
Discolored Conju...	☐ Yes ☒ No	Jundice Conjuncti...	☐ Yes ☒ No			
Hemorrhagic Conj...	☐ Yes ☒ No	Cyanosis	☐ Yes ☒ No			
Mottles	☐ Yes ☒ No	OMI	☐ Yes ☒ No			
PB Systolic (mm...	120	PB Diastolic (m...	60			
Saturation	95	Pluse Rate (BPM)	60			
Breathing Rate (...)	24	Temperature (C°)	☒ Null			
Weight (kg)	3	Weight (SD)	0			
		Help				
Height (cm)	100	Height (SD)	0			
		Help				
BMI	3	BSA	2.15			

Figure 26: general examination component

General	Pulmonary	Cardiac	Abdominal	Urologic	Neurologic	Articular
Pulmonary Exam						
Shape	normal Edit	Ampliation	4			
Symmetry	symétrique Edit	Dyspnea	☒ Null dyspnée Edit			
Digital Hippocratism	☐ Yes ☒ No	Parietal Mass	☐ Yes ☒ No			
Collateral Circulat...	☐ Yes ☒ No					
Regional Exam						
Top Left			Top Right			
Tonality	normale Edit	Tonality	normale Edit			
Vesicular Mur...	bien perçu Edit	Vesicular Mur...	bien perçu Edit			
Vocal Vibration	bien perçu Edit	Vocal Vibration	bien perçu Edit			

Figure 27: pulmonary examination component

The screenshot shows the 'Cardiac' tab selected in a navigation bar. Below the tab is the 'Cardiovascular Exam' section. It contains two columns of symptoms, each with 'Yes' and 'No' radio buttons. The 'No' buttons are selected for all symptoms. The symptoms are: Blockpnea, Chest Pain, Syncope, Hepatalgie, Punctiform Point ..., B1 Noise, Mitral Noise, Mitral Eradiation, Aortic Noise, Orthpnea, Lipothmia, Palpitations, Visible Point Shock, HJR, B2 Noise, Tricuspid Noise, Tricuspid Eradiation, and Pulmonary Noise. Each symptom has a corresponding 'Edit' button. The 'B1 Noise' and 'B2 Noise' fields are set to 'normal'.

Symptom	Yes	No	Edit
Blockpnea	☐	☒	
Chest Pain	☐	☒	
Syncope	☐	☒	
Hepatalgie	☐	☒	
Punctiform Point ...	☐	☒	
B1 Noise			normal Edit
Mitral Noise	☒	☐	bruit cardi... Edit
Mitral Eradiation	☒	☐	irradiation ...Edit
Aortic Noise	☒	☐	bruit cardi... Edit
Orthpnea	☐	☒	
Lipothmia	☐	☒	
Palpitations	☐	☒	
Visible Point Shock	☐	☒	
HJR	☐	☒	
B2 Noise			normal Edit
Tricuspid Noise	☒	☐	bruit cardi... Edit
Tricuspid Eradiation	☒	☐	irradiation ...Edit
Pulmonary Noise	☒	☐	bruit cardi... Edit

Figure 28: cardiovascular examination component

The screenshot shows the 'Abdominal' tab selected in a navigation bar. Below the tab is the 'Abdominal Exam' section. It contains two columns of symptoms, each with 'Yes' and 'No' radio buttons. The 'No' buttons are selected for all symptoms. The symptoms are: Pyrosis, Colon Pain, Shape, Splenomegaly, Glaucon Sign, Murphy Sign, Stomach Pain, Perineal Pain, Liver Arrow, Hernia, and Flot Sign. Each symptom has a corresponding 'Edit' button. The 'Shape' field is set to 'normal'. The 'Liver Arrow' field is set to '8'. The 'Hernia' field has a '+ Add item' button. Below the 'Abdominal Exam' section is the 'Regional Exam' section, which contains three tabs: HCG, Epigastric, and HCD.

Symptom	Yes	No	Edit
Pyrosis	☐	☒	
Colon Pain	☐	☒	
Shape			normal Edit
Splenomegaly	☐	☒	
Glaucon Sign	☐	☒	
Murphy Sign	☐	☒	
Stomach Pain	☐	☒	
Perineal Pain	☐	☒	
Liver Arrow			8
Hernia			+ Add item
Flot Sign	☐	☒	

Regional Exam

Region	Edit
HCG	
Epigastric	
HCD	

Figure 29: abdominal examination component

The screenshot shows the 'Urologic' tab selected in a medical examination interface. The interface includes a top navigation bar with tabs for General, Pulmonary, Cardiac, Abdominal, Urologic (active), Neurologic, and Articular. The main content area is divided into two columns of questions and answers. The left column contains: Pollakuria (Yes/No), Bladder Globe (Yes/No), Micturition Count (0), Colour (orange clair, Edit), Pyuria (Yes/No), and Fecaluria (Yes/No). The right column contains: Disuria (Yes/No), Micturition Burn (Yes/No), Diurese (0), Continence (continence normale, Edit), and Pneumaturia (Yes/No). Below these, there is a 'Regional Exam' section with two sub-tabs: 'Left Flanc' and 'Right Flanc'. Each sub-tab has a 'Pain' field with a radio button (selected) and a 'Null' field.

Figure 30: genitourinary examination component

The screenshot shows the 'Neurologic' tab selected in the same medical examination interface. The 'Neurologic Exam' section is active. The interface is divided into two columns of questions and answers. The left column contains: Hand Side (droite, Edit), Drunk Walking (Yes/No), Latero Deviation (Yes/No), Barre Manoeuvre (Yes/No), Finger-Nose Man... (normale, Edit), Motor Aphasia (Yes/No), Apraxia (Yes/No), Short-Term Amne... (Yes/No), and Thermo-Algie (normale, Edit). The right column contains: Spine Stiffness (Yes/No), Decomposed U T... (Yes/No), Tendons Dance (Yes/No), Mingazzini Mano... (Yes/No), Heel-Knee Mano... (normale, Edit), Understanding A... (Yes/No), Agnosia (+ Add item), Long-Term Amnesia (Yes/No), and Epieritic Superficial (normale, Edit). A red '+ Add item' button is visible next to the 'Agnosia' field.

Figure 31: neurologic examination component

General Pulmonary Cardiac Abdominal Urologic Neurologic **Articular**

Articular Exam

Myalgia ☐ Yes ☒ No Sonnet Sign ☐ Yes ☒ No

Leri Sign ☐ Yes ☒ No Lasegue Sign ☐ Yes ☒ No

Finger-to-ground ... 0 Schober Index (cm) 0

Regional Exam

Side	Region	Edit
côté	joints et ligam...	Edit
Inflammation	Hot Articulation	☐ Yes ☒ No
Effusion	Active Mobili...	☐ Yes ☒ No
Passive Mob...	Against Mobi...	☐ Yes ☒ No

Figure 32: rhumatologic examination component

Newborn ... Newborn ... Apparent... Inappare... Pulmonary Cardiac Abdominal

Symmetric Gestures ☒ Yes ☐ No

Vigorous Cries ☒ Yes ☐ No

Recoloration Time 3 seconds

Saturation 95

Farr (Score) 15

Help

Farr (AW) 33 weeks 3 days

Weight (kg) 1.7

Weight (SD) 0

Help

Height (cm) 42

Figure 33: newborn examination component

Question	Yes	No
Scarf sign	☒	☐
Upper limbs reflexion	☒	☐
Parachute manoeuver	☒	☐
Lower limbs recovery	☒	☐
Popliteal angle	90	
Dorsiflexion angle	20	
Grasping reflex	☒	☐
Moro reflex	☒	☐
Suction reflex	☒	☐

Figure 34: newborn reflexes examination component

Question	Yes	No
Hydrocephaly	☐	☒
Anencephaly	☐	☒
Scaphocephaly	☐	☒
Trigonocephaly	☐	☒
Plagiocephaly	☐	☒
Brachycephaly	☐	☒
Oxycephaly	☐	☒
Pansynostosis	☐	☒
Spina Bifida	☐	☒
Choanal Atresia	☐	☒
Microtia	☐	☒
Anotia	☐	☒
Leucokoria	☐	☒
Scoliosis	☐	☒
Torticollis	☐	☒
Buphthalmos	☐	☒
Anal Inperforation	☐	☒
Urogenital Fistula	☐	☒

Figure 35: newborn apparent malformations examination component

Figure 36: newborn unapparent malformations examination component

E. Assessment component

This component was implemented to track patient evolution over specific scales, indices, scores, or classifications, with nearly 700 assessments supported.

Figure 37: assessment component

F. Intervention component

This component was implemented to track and instruct interventions for patients.

The screenshot shows a web form titled "Intervention". It contains several input fields and a list of interventions. The "Date" field is set to "06/13/2020, 05:19 PM". The "Type" field is set to "soins infirmiers". The "Motive" field is set to "thérapeutique". The "Interventions" field is a list with six items: "hospitalisation", "rechauffement", "monitorage", "position proclive", "apports de base", and "soins des yeux". Each item has a red plus icon and a red trash icon to its left. There are also red plus and trash icons at the top left of the form.

Figure 38: intervention component

The screenshot shows a web form titled "Select a value" with the subtitle "Pick an ontology entry". It contains a search bar and a list of categories. The "categorie" category is expanded, showing a list of intervention types: "interventions chirurgicales", "interventions de radiotherapie", "interventions medicales", "interventions odontologiques", "interventions physiotherapiques", "interventions psychologiques", "procedures medico-administratives", and "soins infirmiers". The "Hash" field is set to "50". The "Label" field is set to "categorie". The "Direct children count" field is set to "8" and the "Deep children count" field is set to "8". The "Children" field is a list with three items: "500 - interventions chirurgicales", "501 - interventions de radiotherapie", and "502 - interventions medicales".

Figure 39: intervention types ontology picker

G. Diagnosis component

This section keeps track of patient diagnoses and diseases along with arguments that are in favor or are against the diagnosis in addition to its status as retained or not.

The ontology is exhaustive. It contains nearly 170,000 diagnoses and was imported from ICD-11.

The screenshot displays a web interface for managing diagnoses. It features two main entry forms, one for 'atélectasie primitive du nouveau-né' and another for 'nouveau-né prématuré, âge gestationnel 33 semaines ter...'. Each entry includes a 'Disease' field with a search icon, a 'Retained' status with 'Yes' and 'No' radio buttons, and buttons for 'Supported By' and 'Opposed By' with '+ Add item' labels. The interface is clean and uses a light gray color scheme.

Figure 40: diagnosis component

H. Monitoring component

When a patient is hospitalized, daily updates are entered into the medical record documenting clinical changes, new information, etc. These are entered by all healthcare team (doctors, nurses, physical therapists, dietitians, clinical pharmacists, respiratory therapists, etc.). They are kept in chronological order and document the sequence of events leading to the current state of health.

Monitoring

Vitals

☒ Yes ☐ No

Ventilation

☐ Yes ☒ No

Ionogram

☐ Yes ☒ No

CBUE

☐ Yes ☒ No

Blood

☐ Yes ☒ No

Obstetric

☐ Yes ☒ No

Scope

☐ Yes ☒ No

Gasometry

☐ Yes ☒ No

Kidney

☐ Yes ☒ No

Hemoculture

☐ Yes ☒ No

Biochemistry

☐ Yes ☒ No

Anthropometrics

☐ Yes ☒ No

Vitals Monitoring

	Date	BPS (mmHg)	BPD (mmHg)	BPM	BR
+	09/28/2020,	120	60	60	24

Figure 41: vitals monitoring component

Monitoring

Vitals

☐ Yes ☒ No

Ventilation

☐ Yes ☒ No

Ionogram

☐ Yes ☒ No

CBUE

☐ Yes ☒ No

Blood

☐ Yes ☒ No

Obstetric

☐ Yes ☒ No

Scope

☒ Yes ☐ No

Gasometry

☐ Yes ☒ No

Kidney

☐ Yes ☒ No

Hemoculture

☐ Yes ☒ No

Biochemistry

☐ Yes ☒ No

Anthropometrics

☐ Yes ☒ No

Scope Monitoring

	Date	TRC (s)	SaO (%)	Temp (C°)
+	09/28/2020,	3	95	37

Figure 42: scope monitoring component

Monitoring

Vitals

☐ Yes
☒ No

Ventilation

☒ Yes
☐ No

Ionogram

☐ Yes
☒ No

CBUE

☐ Yes
☒ No

Blood

☐ Yes
☒ No

Obstetric

☐ Yes
☒ No

Scope

☐ Yes
☒ No

Gasometry

☐ Yes
☒ No

Kidney

☐ Yes
☒ No

Hemoculture

☐ Yes
☒ No

Biochemistry

☐ Yes
☒ No

Anthropometrics

☐ Yes
☒ No

Ventilation Monitoring

	Date	Mode	FR	TI (s)	TE (s)	PEP (mmHg)	PI (mmHg)
+	09/28/2020,	ventilation	20	0.4	0.6	5	16

Figure 43: ventilation monitoring component

Monitoring

Vitals

☐ Yes
☒ No

Ventilation

☐ Yes
☒ No

Ionogram

☐ Yes
☒ No

CBUE

☐ Yes
☒ No

Blood

☐ Yes
☒ No

Obstetric

☐ Yes
☒ No

Scope

☐ Yes
☒ No

Gasometry

☒ Yes
☐ No

Kidney

☐ Yes
☒ No

Hemoculture

☐ Yes
☒ No

Biochemistry

☐ Yes
☒ No

Anthropometrics

☐ Yes
☒ No

Gasometry Monitoring

	Date	pH	PaO2	PaCO2	HCO3	BE	Lactate
+	09/28/2020,	7.36	80	40	25	0	1

Figure 44: blood gas monitoring component

Monitoring

Vitals

☐ Yes
 ☒ No

Ventilation

☐ Yes
 ☒ No

Ionogram

☒ Yes
 ☐ No

CBUE

☐ Yes
 ☒ No

Blood

☐ Yes
 ☒ No

Obstetric

☐ Yes
 ☒ No

Scope

☐ Yes
 ☒ No

Gasometry

☐ Yes
 ☒ No

Kidney

☐ Yes
 ☒ No

Hemoculture

☐ Yes
 ☒ No

Biochemistry

☐ Yes
 ☒ No

Anthropometrics

☐ Yes
 ☒ No

Ionogram Monitoring

	Date	NA	K	Cl	Ca
+	09/28/2020,	120	4	110	85

Figure 45: ionogram monitoring component

Monitoring

Vitals

☐ Yes
 ☒ No

Ventilation

☐ Yes
 ☒ No

Ionogram

☐ Yes
 ☒ No

CBUE

☐ Yes
 ☒ No

Blood

☐ Yes
 ☒ No

Obstetric

☐ Yes
 ☒ No

Scope

☐ Yes
 ☒ No

Gasometry

☐ Yes
 ☒ No

Kidney

☒ Yes
 ☐ No

Hemoculture

☐ Yes
 ☒ No

Biochemistry

☐ Yes
 ☒ No

Anthropometrics

☐ Yes
 ☒ No

Kidney Monitoring

	Date	Urea	Creat	DFG	Diurese
+	09/28/2020,	0.5	12	120	3

Figure 46: kidney monitoring component

Monitoring

Vitals

☐ Yes
☒ No

Ventilation

☐ Yes
☒ No

Ionogram

☐ Yes
☒ No

CBUE

☒ Yes
☐ No

Blood

☐ Yes
☒ No

Obstetric

☐ Yes
☒ No

Scope

☐ Yes
☒ No

Gasometry

☐ Yes
☒ No

Kidney

☐ Yes
☒ No

Hemoculture

☐ Yes
☒ No

Biochemistry

☐ Yes
☒ No

Anthropometrics

☐ Yes
☒ No

CBUE Monitoring

	Date	Leucocytes	Bacteria	Result	Organism	
trash + print	09/28/2020, calendar	10000	10000	resultat de culture	organisme	antiinfectieux ge

Figure 47: CBUE monitoring component

Monitoring

Vitals

☐ Yes
☒ No

Ventilation

☐ Yes
☒ No

Ionogram

☐ Yes
☒ No

CBUE

☐ Yes
☒ No

Blood

☐ Yes
☒ No

Obstetric

☐ Yes
☒ No

Scope

☐ Yes
☒ No

Gasometry

☐ Yes
☒ No

Kidney

☐ Yes
☒ No

Hemoculture

☒ Yes
☐ No

Biochemistry

☐ Yes
☒ No

Anthropometrics

☐ Yes
☒ No

Hemoculture Monitoring

	Date	Result	Organism	Antibiotic
trash + print	09/28/2020, calendar	resultat de culture	organisme	antiinfectieux g�n�raux � usage syst�

Figure 48: blood culture monitoring component

Monitoring

Vitals

☐ Yes
☒ No

Ventilation

☐ Yes
☒ No

Ionogram

☐ Yes
☒ No

CBUE

☐ Yes
☒ No

Blood

☒ Yes
☐ No

Obstetric

☐ Yes
☒ No

Scope

☐ Yes
☒ No

Gasometry

☐ Yes
☒ No

Kidney

☐ Yes
☒ No

Hemoculture

☐ Yes
☒ No

Biochemistry

☐ Yes
☒ No

Anthropometrics

☐ Yes
☒ No

Blood Monitoring

	Date	HB	Ht	GB	Platelets	Lympho	Neutro	Baso
+	09/28/2020,	12	45	12000	2000	2000	4000	100

Figure 49: blood cell count monitoring component

Monitoring

Vitals

☐ Yes
☒ No

Ventilation

☐ Yes
☒ No

Ionogram

☐ Yes
☒ No

CBUE

☐ Yes
☒ No

Blood

☐ Yes
☒ No

Obstetric

☐ Yes
☒ No

Scope

☐ Yes
☒ No

Gasometry

☐ Yes
☒ No

Kidney

☐ Yes
☒ No

Hemoculture

☐ Yes
☒ No

Biochemistry

☒ Yes
☐ No

Anthropometrics

☐ Yes
☒ No

Biochemistry Monitoring

	Date	GAJ	CRP	ALAT	ASAT
+	09/28/2020,	0.90	4	20	20

Figure 50: biochemistry monitoring component

Monitoring

Vitals

☐ Yes
☒ No

Ventilation

☐ Yes
☒ No

Ionogram

☐ Yes
☒ No

CBUE

☐ Yes
☒ No

Blood

☐ Yes
☒ No

Obstetric

☒ Yes
☐ No

Scope

☐ Yes
☒ No

Gasometry

☐ Yes
☒ No

Kidney

☐ Yes
☒ No

Hemoculture

☐ Yes
☒ No

Biochemistry

☐ Yes
☒ No

Anthropometrics

☐ Yes
☒ No

Obstetric Monitoring

	Date	FCB	Dilatation	Membranes Rupture
trash + print	09/28/2020, calendar	120	4	+ dropdown

Figure 51: obstetric monitoring component

Monitoring

Vitals

☐ Yes
☒ No

Ventilation

☐ Yes
☒ No

Ionogram

☐ Yes
☒ No

CBUE

☐ Yes
☒ No

Blood

☐ Yes
☒ No

Obstetric

☐ Yes
☒ No

Scope

☐ Yes
☒ No

Gasometry

☐ Yes
☒ No

Kidney

☐ Yes
☒ No

Hemoculture

☐ Yes
☒ No

Biochemistry

☐ Yes
☒ No

Anthropometrics

☒ Yes
☐ No

Anthropometrics Monitoring

	Date	Height (cm)	Weight (kg)	BMI
trash + print	09/28/2020, calendar	100	0	0

Figure 52: anthropometrics monitoring component

I. Summary component

This component displays patient admission summaries, including his identity, clinical conclusion, diagnosis, procedures, interventions, and treatments. These summaries are captured through a rich editor.

Identity	Enter contents
Conclusion	Enter contents
Diagnosis	Enter contents
Procedure	Enter contents
Interventions	Enter contents
Treatments	Enter contents

Figure 53: summary component

2. Hospital management system

Also called hospital information system (HIS), it is an element of health informatics that focuses on administrative needs of hospitals. A HIS is a comprehensive, integrated information system with many implementations designed to handle all facets of hospital operation, such as medical , administrative, financial , and legal issues, and the related processing of services [105].



Figure 54: hospital management system

Potential benefits of hospital information systems include [105]:

- Efficient and accurate administration of finance, the diet of the patient, engineering, and distribution of medical aid. It helps to view a broad picture of hospital growth.
- Improved monitoring of drug usage and study of effectiveness. This leads to the reduction of adverse drug interactions while promoting more appropriate pharmaceutical utilization.
- Enhances information integrity, reduces transcription errors, and reduces duplication of information entries.
- Hospital software is easy to use and eliminates errors caused by handwriting. New technology computer systems give perfect performance to pull up information from servers or cloud servers.

Patients are at the center of every HMS; HMS must register patients, accept patients, transfer / discharge patients, ensure patient follow-up via appointments, and medical staff must be able to request and respond to treatments, medications, and consulting.

A. Patient management

Patient management follows the CRUD approach; patient records are created, updated, deleted, or viewed.

Captured demographics include identity, origin, residence, parental status, socioeconomic status, lifestyle, ethnicity, address ... etc.

Records listing are ordered by creation date from newer to older, listing displays the full name, gender, age, birth date

Listing is available through the three conventional view types (Grid, Row, List), with the grid view being the default view.

Records can be filtered, reordered, and navigated in real-time; Actions can be carried on single items as well as on a bulk of items.

Only one listing variant is available for patient records, as the patient record is at the core of the system, and it is added before any related entity.

Create Patient

Patients

+ Create Patient

✕

Full Name		Gender	sexe Edit
Father Name	Null	Mother Name	Null
IP		Insurance	assurance Edit
Birth Date	09/27/2020, 11:41 PM	Age	0 Y 0 M 0 W 0 D 0 H 0 M
Sibling	0	Number in sibling	0
Religion	Null religions et philosophies Edit	Ethnicity	Null groupes ethniques et ... Edit
Lifestyle	Null mode de vie Edit	Economic Level	Null statuts économiques Edit
Civil Status	Null états civils Edit	Parental Status	Null états parentaux Edit
Country	Null Ville Edit	Occupation	Null occupation Edit
Origin	Null Ville Edit	Residence	Null Ville Edit
Address	Null	Phone Number	Null
Email	Null	Blood Type	Null A+ ▼

Figure 55: create patient action

Patients

+

Create Patient

Filters

No filter

Grid

Check all

No item

Name	Gender	Age	Birth Date	More
	feminin	3 months	12/6/20	
	feminin	2 months	2/7/20	
	masculin	2 months	7/7/20	
	feminin	5 months	28/4/20	

Figure 56: patients listing interface

B. Admission management

Similarly, admissions management follows the CRUD approach; admission records are created, updated, deleted, or viewed and requested, accepted, rejected, discharged, or transferred.

In some cases, it would be counter-intuitive to load the whole admission record just to read the summary.

Every admission record is directly related to a patient record and linked with other EHR record entities.

Captured entities are all components of an EHR record, including admission status, history, examination, evaluations, procedures, interventions, prescriptions, monitoring.

Table V: admission unconventional CRUD actions

Action	Description
Request	Request an admission
Accept	Accept an admission request
Reject	Reject an admission request
Discharge	Discharge an admission
Status	View admission status details
Summary	View admission summary

Records listing are ordered by creation date from newer to older, listing displays the full name, service/department, age, gender, and admission status.

Listing is available through the three conventional view types (Grid, Row, List), with the grid view being the default view.

Create Admission

Admissions

Reset

+ Create Admission

×

Patient		
Admission	Date	09/27/2020, 11:44 PM
History	Discharge	☐ Null 01/01/1970, 12:00 AM
Examination	Complaint	symptôme Edit
Assessment	Reason	maladie Edit
Consent	Status	demandé Edit
Intervention	Service	services d'un hôpital Edit
Procedure	Gender	sexe Edit
Prescription	Bed	
Consulting		
Diagnosis		
Monitoring		
Summary		

Figure 57: create admission action

Request Admission

Admissions

Reset

+ Request Admission

×

Patient

Click to attach

Variant

Patients

Choose

Complaint	symptôme	Edit
Reason	maladie	Edit
Service	services d'un hôpital	Edit
Bed		

Figure 58: request admission action

Table VI: admission record components

Component	Description
Patient	Patient record relation linking
Status	Information related to the admission that is HMS related such as department, bed number, gender, admission/discharge date, and status
History	Patient history; custom forms will be displayed depending on the admission department; supported history types are medical, surgical, allergic, obstetric, gynecologic, delivery, and family histories.
Examination	Clinical examination, all regions are supported, pediatric and newborn examinations are supported; exams displayed will change according to the admission's department.
Assessment	Medical scores, scaling, evaluations will go here
Consent	Patient consents go here
Intervention	Interventions requested for the patient
Procedure	Patient procedures go here; imaging, biological, and exploration procedures are supported
Prescription	Patient prescription go here, doses and drug formulations are calculated in real-time
Consulting	Patient inter-disciplinary medical consults
Diagnosis	All suggested patient diagnoses are saved in this component, whether retained or not, including arguments that are in favor or that are against
Monitoring	Patient monitoring tables and charts
Summary	Admission summary

Six record listing variants are available, filtered by status; Yet listing variants can be further filtered via real-time filtering.

Records can be filtered, reordered, and navigated in real-time; actions can be carried on single items as well as on a bulk of items.

Table VII: admission listing variants

Variant	Description
All	Listing of all admissions
Pending	Listing of ongoing admissions
Requested	Listing of requested admissions
Rejected	Listing of rejected admissions
Discharged	Listing of discharged admissions
Transferred	Listing of transferred admissions

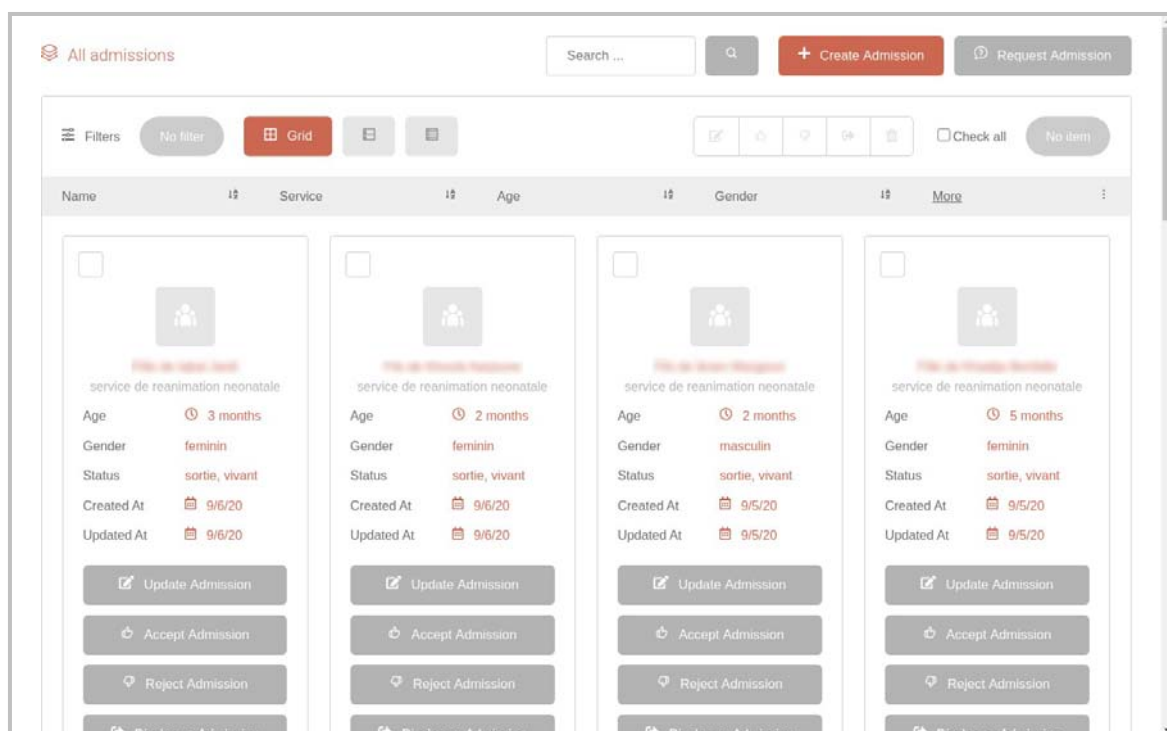


Figure 59: admission records listing

C. Procedure management

Similarly, procedures management follows the CRUD approach, procedure records are created, updated, deleted, or viewed, and yet procedures can be requested, accepted, rejected, responded, or ended.

Table VIII: procedure unconventional CRUD actions

Action	Description
Request	Request a procedure
Accept	Accept a procedure request
Reject	Reject a procedure request
End	End a procedure

Each procedure is directly linked to a patient record and may be indirectly linked to an admission record.

Procedures are organized by department, and can comprise as many procedures as long as they fit into the same department; procedure motives are also captured.

We compiled a list of the most requested procedures and made preconfigured input templates to ease data capture; added frequent biochemistry, microbiology, hematology, and imaging report templates along with four generic report templates.

Table IX: supported procedure report types

Report	Description
Blood count	Blood count and hemogram
Ionogram	Major electrolytes concentration in blood
Gasometry	Blood gases and acid–basic equilibrium
Biochemistry	Commonly requested biochemistry procedures that doesn't fit in other templates
Kidney	Kidney function and excretion monitoring
Liver	Liver function and excretion monitoring
Heart	Hearth acute and chronic failures monitoring
Hematology	Coagulation and hematologic functions
Iron	Iron metabolism panel
Lipid	Lipids metabolism panel
Urine examination	Cytology–bacterial urine examination panel
Microbiology	Common bacteriology panels
Lumbar punction	Cytology–bacterial–biochemistry cerebral–spinal fluid panel
Imaging	Imaging procedures
Table	Parameter–value–unit–interpretation tuple table suitable for biochemistry findings
Text	Suitable for textual reports
Key/value	Suitable for key/value tuple pairs
X/Y	Suitable for numeric mappings or mathematical functions mapping

Procedure Report

Blood

☐ Yes ☒ No

Gasometry

☐ Yes ☒ No

Kidney

☐ Yes ☒ No

Cardiac

☐ Yes ☒ No

Iron

☐ Yes ☒ No

Urine Examination

☐ Yes ☒ No

Lumbar punction

☐ Yes ☒ No

Table

☐ Yes ☒ No

Key/Value

☐ Yes ☒ No

Ionogram

☐ Yes ☒ No

Biochemistry

☐ Yes ☒ No

Liver

☐ Yes ☒ No

Hematology

☐ Yes ☒ No

Lipid

☐ Yes ☒ No

Microbiology

☐ Yes ☒ No

Imaging

☐ Yes ☒ No

Text

☐ Yes ☒ No

X/Y

☐ Yes ☒ No

Figure 60: supported procedure templates

Blood Report

HB (g/dL)

☒ Null

MCV (fL)

☒ Null

MCH (pg)

☒ Null

White cells (g/dL)

☒ Null

Lymphocytes (count)

☒ Null

Basophiles (count)

☒ Null

Ht

☒ Null

MCHC (g/dL)

☒ Null

Reticulocytes (%)

☒ Null

Platelets (count)

☒ Null

Neutrophiles (count)

☒ Null

Eosinophiles (count)

☒ Null

Figure 61: blood count panel report

Ionogram Report

Calcium (mg/L)

☒ Null

Chlore (mmol/L)

☒ Null

Phosphates (mmol/L)

☒ Null

Ammoniac (mmol/L)

☒ Null

Zinc (mmol/L)

☒ Null

Nickel (mmol/L)

☒ Null

Sodium (mmol/L)

☒ Null

Potassium (mmol/L)

☒ Null

Magnesium (mmol/L)

☒ Null

Copper (mmol/L)

☒ Null

Mercury (mmol/L)

☒ Null

Oxalate (mmol/L)

☒ Null

Figure 62: ionogram panel report

Kidney Report

Urea (g/L)

Null

Creatine (mg/L)

Null

GFR (mL/min)

Null

Volume (mL/kg/h)

Null

Calcium (mg/L)

Null

Sodium (mmol/L)

Null

Chlore (mmol/L)

Null

Potassium (mmol/L)

Null

Oxalate (mg/L/d)

Null

Uric acid (mg/L/d)

Null

Protein (mg/d)

Null

Microalbumin (mg/d)

Null

Osmolality (msom/kg)

Null

Osmolarity (mosm/L)

Null

Figure 63: kidney function panels report

Liver Report

GGT (UI/L)

Null

LAP (ug/L)

Null

ALAT (UI/L)

Null

ASAT (UI/L)

Null

Amylase (UI/L)

Null

Lipase (UI/L)

Null

LDH (UI/L)

Null

Total bilirubin (UI/L)

Null

Free bilirubin (UI/L)

Null

Unconjugated bilirubin...

Null

Protein (UI/L)

Null

Albumin (UI/L)

Null

Figure 64: liver function panels report

Cardiac Report

Troponin I (ng/mL)

Null

Troponin C (ng/mL)

Null

Troponin T (ng/mL)

Null

CK-MB (UI/L)

Null

BNP (pg/mL)

Null

NT-proBNP (pg/mL)

Null

Myoglobin (ng/mL)

Null

Figure 65: cardiac function panels report

Gasometry Report

☐ pH ☐ Null
☐ Bases exces ☐ Null

☐ pO2 (mmHg) ☐ Null
☐ pCO2 (mmHg) ☐ Null

☐ SO2 (%) ☐ Null
☐ HCO3 (mmol/L) ☐ Null

☐ Lactate (mmol/L) ☐ Null
☐ Pyruvic acid (mmol/L) ☐ Null

☐ Ketones (mmol/L) ☐ Null
☐ HbCO (%) ☐ Null

Figure 66: gasometry panel report

Hematology Report

☐ Blood Group ☐ Null
☐ Rhesus ☐ Null ☐ Yes ☐ No

☐ RAI
☐ Direct Coombs ☐ Null ☐ Yes ☐ No

☐ Indirect Coombs ☐ Null ☐ Yes ☐ No

☐ BT (minute) ☐ Null
☐ PT (second) ☐ Null

☐ TT (second) ☐ Null
☐ ACT (second) ☐ Null

☐ Fibrinogen (g/L) ☐ Null
☐ D-Dimer (ng/mL) ☐ Null

☐ FDP (ug/mL) ☐ Null

Figure 67: hematology panel report

Iron Report

☐ Iron (ug/dL) ☐ Null
☐ Ferritin (ng/dL) ☐ Null

☐ Transferrin (mg/dL) ☐ Null
☐ TIBC (ug/dL) ☐ Null

☐ TS (ug/dL) ☐ Null

Figure 68: iron metabolism panel report

Lipid Report

☐ Total cholestrol (g/L) ☐ Null
☐ HDL cholestrol (g/L) ☐ Null

☐ LDL cholestrol (g/L) ☐ Null
☐ Triglycerid (g/L) ☐ Null

☐ Free fatty acids (g/L) ☐ Null
☐ Apo A (g/L) ☐ Null

☐ Apo B (g/L) ☐ Null
☐ Lipo (a) (g/L) ☐ Null

Figure 69: lipid metabolism panel report

Biochemistry Report

Glucose fasting (g/L)

☐ Null

Glucose non-fasting (...)

☐ Null

HBA1C (%)

☐ Null

Fructosamine (umol/L)

☐ Null

CRP (mg/L)

☐ Null

Procalcitonin (ng/mL)

☐ Null

Uric acid (mg/L)

☐ Null

Sedimenation rate (m...

☐ Null

Figure 70: common biochemistry panels report

Urine Examination Report

White cells (count)

☐ Null

Red cells (count)

☐ Null

Bacteria

☐ Null

Result

☐ Null

resultat de culture Edit

Organism

☐ Null

organisme Edit

Antibiotic

☐ Null

antiinfectieux gén... Edit

Sensitivity

☐ Null

sensibilité d'antibi... Edit

Figure 71: cyto-bacterial urine examination procedure report

Microbiology Report

Bacteria

☐ Null

Result

☐ Null

resultat de culture Edit

Organism

☐ Null

organisme Edit

Antibiotic

☐ Null

antiinfectieux gén... Edit

Sensitivity

☐ Null

sensibilité d'antibi... Edit

Figure 72: microbiology panels report

Lumbar Punction Report

White cells (count)

☐ Null

Red cells (count)

☐ Null

Glucose (g/L)

☐ Null

Protein (g/L)

☐ Null

Bacteria

☐ Null

Result

☐ Null

resultat de culture Edit

Organism

☐ Null

organisme Edit

Antibiotic

☐ Null

antiinfectieux gén... Edit

Sensitivity

☐ Null

sensibilité d'antibi... Edit

Figure 73: lumbar punction panel report

Imaging Report

Medias

+ Add item

Report

+ Enter contents

Conclusion

+ Enter contents

Diagnoses

+ Add item

Figure 74: imaging procedures report

Table Report

	Key	Value	Unit	Result
+	paramètre	1	unité	constatations de valeur

Figure 75: generic biochemistry table report

Text Report

Report

+ Enter contents

Figure 76: generic text report

Key/Value Report

	Key	Value
+		

Figure 77: generic key/value report

X/Y Report

	X	Y
+	0	0

Figure 78: generic x/y report

Records listing are ordered by creation date from newer to older, listing displays department, patient name, patient gender, status, and if it has received a response or not.

Listing is available through the three conventional view types (Grid, Row, List), with the grid view being the default view.

Five record listing variants are available, filtered by status; Yet listing variants can be further filtered via real-time filtering.

Table X: procedure listing variants

Variant	Description
All	Listing of all procedures
Pending	Listing of ongoing procedures
Requested	Listing of requested procedures
Rejected	Listing of rejected procedures
Finished	Listing of finished procedures

The screenshot shows a web application interface for creating a procedure. At the top, there's a header bar with the title 'Create Procedure' and a subtitle 'Procedures'. On the right of the header are two buttons: 'Reset' and '+ Create Procedure', along with a close icon. The main form is divided into two sections: 'Patient' and 'Procedure'. The 'Patient' section has a text input field labeled 'Patient' with a hint 'Click to attach', a dropdown menu for 'Variant' currently set to 'Patients', and a 'Choose' button. The 'Procedure' section contains several fields: 'Date' with a date-time picker showing '09/27/2020, 11:52 PM', 'Type' with a text input 'categorie' and an 'Edit' link, 'Motive' and 'Procedures' each with a '+ Add item' button, and 'Status' with a text input 'demandé' and an 'Edit' link. A vertical scrollbar is visible on the right side of the form.

Figure 79: create procedure action

Reporting

Text

☐ Yes
☒ No

Imaging

☐ Yes
☒ No

Table

☒ Yes
☐ No

X/Y

☐ Yes
☒ No

Key/Value

☐ Yes
☒ No

Generic

☐ Yes
☒ No

	Key	Value	Unit	Result
  	Organisme	0	unité	normale
  	Globules blancs	1	unité	normale
  	Glucose	0.63	unité	normale
  	Protéines	1.21 	unité	normale

Figure 80: procedure reporting interface

All procedures

Search ...

+ Create Procedure

Request Procedure

Filters

No filter

Grid

☐ Check all

No item

Type	Name	Gender	Status	More
microbiologie Gender feminin Status terminé, toutes... Responded Yes Created At 9/6/20 Updated At 9/6/20 Update Procedure Accept Procedure Reject Procedure End Procedure	biochimie Gender feminin Status terminé, toutes... Responded Yes Created At 9/6/20 Updated At 9/6/20 Update Procedure Accept Procedure Reject Procedure End Procedure	hematologie Gender feminin Status terminé, toutes... Responded Yes Created At 9/6/20 Updated At 9/6/20 Update Procedure Accept Procedure Reject Procedure End Procedure	biochimie Gender masculin Status terminé, toutes... Responded Yes Created At 9/5/20 Updated At 9/5/20 Update Procedure Accept Procedure Reject Procedure End Procedure	

Figure 81: procedure records listing

D. Prescription management

Electronic prescribing (e-prescribing or e-Rx) is the computer-based electronic generation, transmission, and filling of a medical prescription [106,107,108].

Similarly, prescriptions management follows the CRUD approach, prescription records are created, updated, deleted, or viewed, and yet prescriptions can be requested, accepted, rejected, responded, disposed, collected, or ended.

Table XI: prescription unconventional CRUD actions

Action	Description
Request	Request a prescription
Accept	Accept a prescription request
Reject	Reject a prescription request
Dispose	Dispose a prescription request
End	End a prescription

Our prescription management model is considered to be a qualified e-prescribing software as it is capable of performing all of the following functions [106,107,108]:

- Patient identification.
- Generating a complete active medication list.
- Access to patient historical data.
- Printing prescriptions.

Compared to paper-based prescribing, e-prescribing can improve health and reduce costs because it can [106]:

- Reduce prescribing and dispensing errors.
- Decrease the work needed to execute a prescription.
- Speed receipt of prescribed drugs.
- Improve medication compliance (taking the prescribed medications on time) by reducing lost and unfilled prescriptions and minimizing patient costs.
- Reduce the incidence of drug diversion (drug abuse) by alerting providers and pharmacists of duplicate prescriptions for controlled substances.

Each prescription is directly linked to a patient record and may be indirectly linked with an admission record.

Table XII: prescription record components

Component	Description
Patient	Patient record relation
Intake	Patient general information such as weight, height in addition to intake information
Enteral feeding	Enteral feeding prescription by weight, delivered in portions
Electrolyte perfusion	Electrolyte perfusion by mmol by weight delivered in portions
Par-enteral perfusion	Treatment perfusion by mg by weight delivered in portions
Auto perfusion	Drug auto infusions by $\mu\text{g}/\text{kg}/\text{min}$ delivery by auto perfusion devices
Medication	Medications that cannot be delivered through perfusion or have specific delivery instructions
Transfusion	Blood derivatives transfusion and treatments

Prescription


Date


Status
[Edit](#)

Motive

+ Add item

Weight (kg)

Height (cm)

BMI

BSA

Intake (ml/kg/d)

Protein (g/kg/d)

Lipids (g/kg/d)

Glucids (g/kg/d)

Figure 82: prescription intake component

Enteral Feeding 

Type	type d'allaitement	Edit
Age	age d'allaitement	Edit
Total (ml/kg/d)	0	
Total (ml)	0	
Doses	☐ 	
Portion (ml)	0	

Figure 83: prescription enteral feeding component

Electrolyte Perfusion 

  	Type	suppléments minéraux Edit	Concentration (m...	0.01
	Total (mmol/kg/d)	0	Total (mmol)	0
	Total (ml)	0	Doses	☐ 
	Portion (ml)	0	Duration	0 hour 0 minute

Figure 84: prescription electrolyte perfusion component

Parenteral Perfusion 

  	Type	médicament Edit	Concentration (m...	0.001
	Total (mg/kg/d)	0	Total (mg)	0
	Total (ml)	0	Doses	☐ 
	Portion (ml)	0	Duration	0 hour 0 minute

Figure 85: prescription parenteral perfusion component

Auto Perfusion 

  	Type	médicament Edit	Concentration (m...	0.00001
	Total (ug/kg/min)	0	Total (mg/h)	0
	Total (ml/h)	0	Duration	0 hour 0 minute

Figure 86: prescription auto perfusion component

Type

médicament

Edit

Name

Form

forme

Edit

Route

route

Edit

Formula

10

Unit

mg

Total (unit/kg/d)

1

Total (unit)

0

Doses

+

Portion (unit)

0

Portion (count)

0

Instruction

N/A

Figure 87: prescription medication component

Transfusion

Type

transfusions sanguin... Edit

Rhesus

☐ Yes ☒ No

Count

1

Phenotyped

☐ Yes ☒ No

Compatibilized

☐ Yes ☒ No

Blood Group

O

RAI

+ Add item

Volume (ml)

100

Irradiated

☐ Yes ☒ No

Leukodepleted

☐ Yes ☒ No

Figure 88: prescription transfusion component

Records listing are ordered by creation date from newer to older, listing displays department, patient name, patient gender, status, and if it has received a response or not.

Listing is available through the three conventional view types (Grid, Row, List), with the grid view being the default view.

Seven record listing variants are available, filtered by status; Yet listing variants can be further filtered via real-time filtering.

Records can be filtered, reordered, and navigated in real-time; Actions can be carried on single items as well as on a bulk of items.

Table XIII: prescription listing variants

Variant	Description
All	Listing of all prescriptions
Pending	Listing of ongoing prescriptions
Requested	Listing of requested prescriptions
Rejected	Listing of rejected prescriptions
Disposed	Listing of disposed prescriptions
Collected	Listing of collected prescriptions
Finished	Listing of finished prescriptions

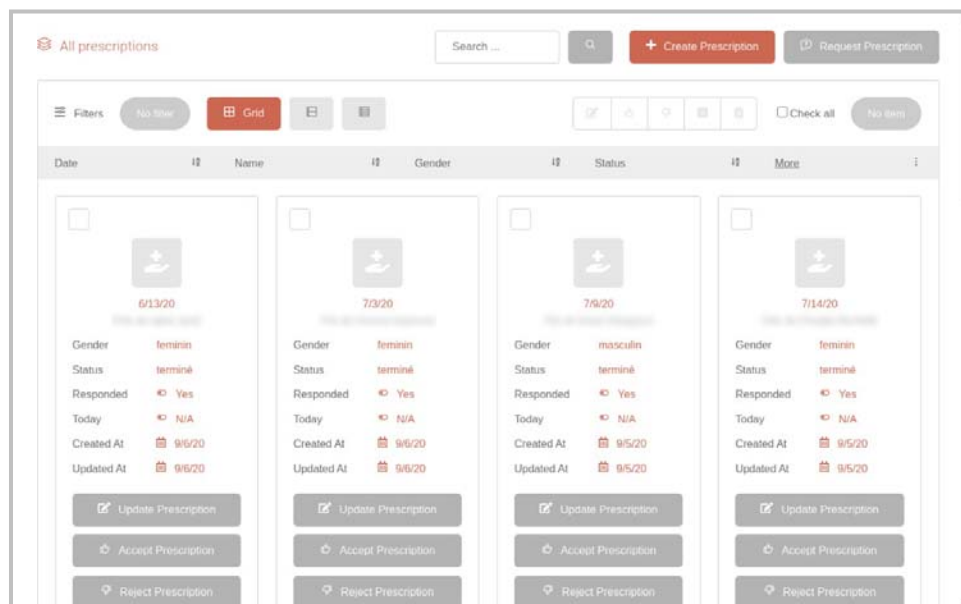


Figure 89: prescription records listing

E. Consulting management

Medical consulting is the interdisciplinary cooperation between different specialties to construct an integrale view of patient records.

Similarly, consulting management follows the CRUD approach; consulting records are created, updated, deleted, or viewed, and yet consulting can be requested, accepted, rejected, responded, or ended.

Table XIV: consulting unconventional CRUD actions

Action	Description
Request	Request a consulting
Accept	Accept a consulting request
Reject	Reject a consulting request
End	End a consulting

Each consulting record is directly linked to a patient record and may be indirectly linked with an admission record.

Clinical information is delivered through a free entry form; physicians can also read admission records by following relation links.

Consulting reports use a generic reporting template that monitors progress, severity, temporality, symptoms, and diagnoses.

The screenshot shows a web interface titled "Report" in red text. On the left, there is a vertical list of categories: Evolution, Temporality, Severity, Symptoms, Diagnoses, and Conclusion. To the right of these categories are input fields and buttons. For "Evolution", "Temporality", and "Severity", there is a red square icon, the word "Null", a text input field containing "évolution", "temporalité", and "gravité" respectively, and an "Edit" link. For "Symptoms" and "Diagnoses", there is a red button with a plus sign and the text "Add item". For "Conclusion", there is a large text input field with a plus sign and the text "Enter contents". A printer icon is located in the top right corner of the form area.

Figure 90: generic consulting reporting component

Records listing are ordered by creation date from newer to older, listing displays department, patient name, patient gender, status, and if it has received a response or not.

Listing is available through the three conventional view types (Grid, Row, List), with the grid view being the default view.

Five record listing variants are available, filtered by status; Yet listing variants can be further filtered via real-time filtering.

Table XV: consulting listing variants

Variant	Description
All	Listing of all consulting
Pending	Listing of ongoing consulting
Requested	Listing of requested consulting
Rejected	Listing of rejected consulting
Finished	Listing of finished consulting

Create Consulting

Consultings

Reset

+ Create Consulting

×

Patient

Patient

Click to attach

Variant

Patients

▼

Choose

Consulting

Date

09/28/2020, 04:47 AM

📅

Service

services d'un hôpital

Edit

Status

demandé

Edit

Motive

+ Add item

Details

+ Enter contents

Figure 91: create consulting action

Figure 92: request consulting action

Service	Name	Gender	Status	More
service d'urologie		feminin	demandé	

Figure 93: consulting records listing

F. Consent management

Informed consent is a process for getting permission before conducting a healthcare intervention on a person or for disclosing personal information. A health care provider may ask a patient to consent to receive therapy before providing it, or a clinical researcher may ask a research participant before enrolling that person into a clinical trial. Informed consent is collected according to guidelines from medical ethics and research ethics [109,110].

Similarly, consents management follows the CRUD approach; consulting records are created,

updated, deleted, or viewed, and yet consulting can be requested, accepted, or rejected.

Table XVI: consent unconventional CRUD actions

Action	Description
Accept	Accept a consent request
Reject	Reject a consent request

For an individual to give valid informed consent, three components must be present: disclosure, capacity, and voluntariness [109,111].

Table XVII: informed consent requirements [109,111]

Requirement	Description
Disclosure	Requires the requestor to supply each prospective subject with the information necessary to make an autonomous decision and ensure that the subject adequately understands the information provided. This latter requirement implies that a written consent form be written in lay language suited for the comprehension skills of the subject population, as well as assessing the level of understanding through conversation (to be informed)
Capacity	Pertains to the ability of the subject to both understand the information provided and form a reasonable judgment based on the potential consequences of his/her decision
Voluntariness	Refers to the subject's right to freely exercise his/her decision making without being subjected to external pressure such as coercion, manipulation, or undue influence

Our model includes an exhaustive list of disclosure acknowledgments, capacity statements, and voluntarism expressions.

Consent response is recorded and can be consolidated with visual proof by uploading the proof image file.

Each consent record is directly linked to a patient record and may be indirectly linked with an admission record.

Records listing are ordered by creation date from newer to older, listing displays type, patient name, patient gender, and status.

Listing is available through the three conventional view types (Grid, Row, List), with the grid view being the default view.

Four record listing variants are available, filtered by status; Yet listing variants can be further filtered via real-time filtering.

Records can be filtered, reordered, and navigated in real-time; actions can be carried on single items as well as on a bulk of items.

Table XVIII: consent listing variants

Variant	Description
All	Listing of all consent
Pending	Listing of ongoing consent
Requested	Listing of requested consent
Rejected	Listing of rejected consent

Create Consent

Consents

Reset

+ Create Consent

×

Variant

Patients

▼

Choose

Consent

Print

Date

09/28/2020, 04:54 AM

📅

Type

consentement

Edit

Instructions

+ Add item

Acknowledgments

+ Add item

Capacities

+ Add item

Status

demandé

Edit

Proxy

☐ Yes ☒ No

Proof

+ Add item

Figure 94: create consent action

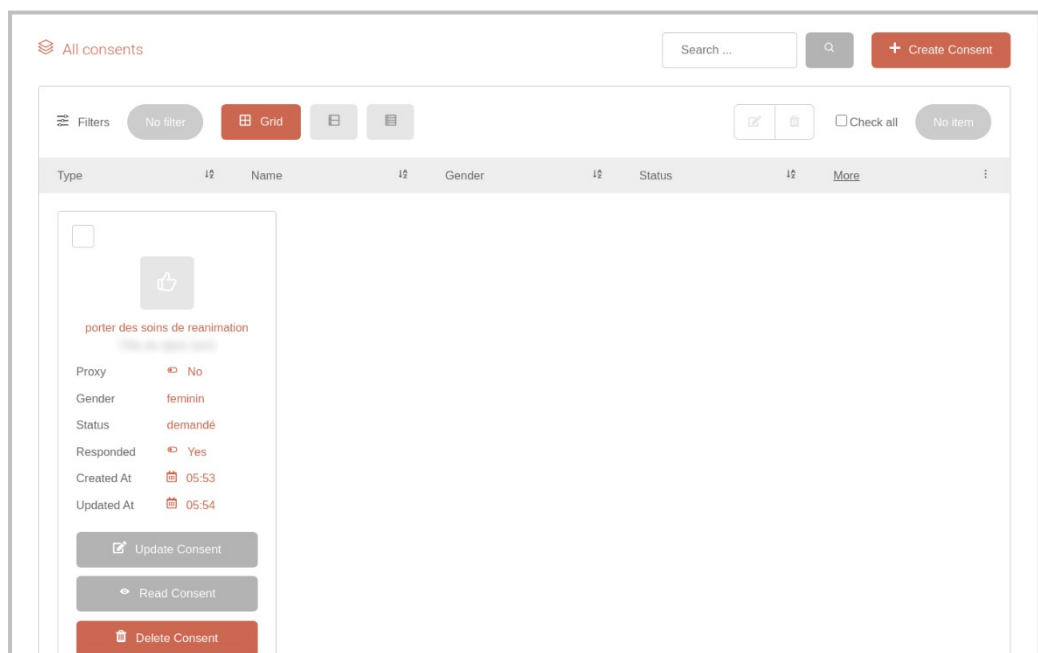


Figure 95: consent records listing

G. Appointment management

Appointments management follows the CRUD approach; appointment records are created, updated, deleted, or viewed, and yet consulting can be responded to, canceled, or ended.

Table XIX: appointment unconventional CRUD actions

Action	Description
Cancel	Cancel an appointment
End	End an appointment

Each appointment record is directly linked to a patient record and may be indirectly linked with an admission record.

Appointment reports use a generic reporting template that monitors progress, severity, temporality, symptoms, and diagnoses.

Figure 96: generic appointment reporting component

Records listing are ordered by creation date from newer to older, listing displays department, patient name, patient gender, status, and response status.

Listing is available through the three conventional view types (Grid, Row, List), with the grid view being the default view.

Five record listing variants are available, filtered by status; Yet listing variants can be further filtered via real-time filtering.

Table XX: appointment listing variants

Variant	Description
All	Listing of all appointments
Unattended	Listing of unattended appointments
Attended	Listing of attended appointments
Passed	Listing of past appointments
Pending	Listing of pending appointments

Records can be filtered, reordered, and navigated in real-time; actions can be carried on single items as well as on a bulk of items.

The 'Create Appointment' form is divided into several sections: 'Patient', 'Appointment', and 'Reporting'. The 'Patient' section includes a text input for the patient's name, a 'Click to attach' link, a 'Variant' dropdown menu set to 'Patients', and a 'Choose' button. The 'Appointment' section contains a 'Date' field with the value '09/28/2020, 04:58 AM', an 'Attended' checkbox with 'Yes' and 'No' options, a 'Service' dropdown menu set to 'services d'un hôpital', and a 'Details' section with a '+ Enter contents' button. The 'Reporting' section is currently empty. At the top right, there are 'Reset' and 'Create Appointment' buttons.

Figure 97: create appointment action

The 'All appointments' listing page features a search bar, a 'Create Appointment' button, and a table of appointments. The table has columns for 'Service', 'Name', 'Gender', 'Date', and 'More'. A sidebar on the left shows a calendar icon, the text 'service d'oncologie', and a list of appointment details: Gender (feminin), Date (05:59), Attended (No), Passed (N/A), Created At (05:59), and Updated At (05:59). Below the details are buttons for 'Update Appointment', 'End Appointment', and 'Cancel Appointment'.

Service	Name	Gender	Date	More
service d'oncologie		feminin	05:59	

Figure 98: appointment records listing

H. Guide management

Guides make good educational material for newly enrolled interns, residents, or students; guides are written in Markdown language, which makes them uniform and easily modifiable.

Our markup editor is both a markup editor and a WYSIWYG editor, which brings the best in

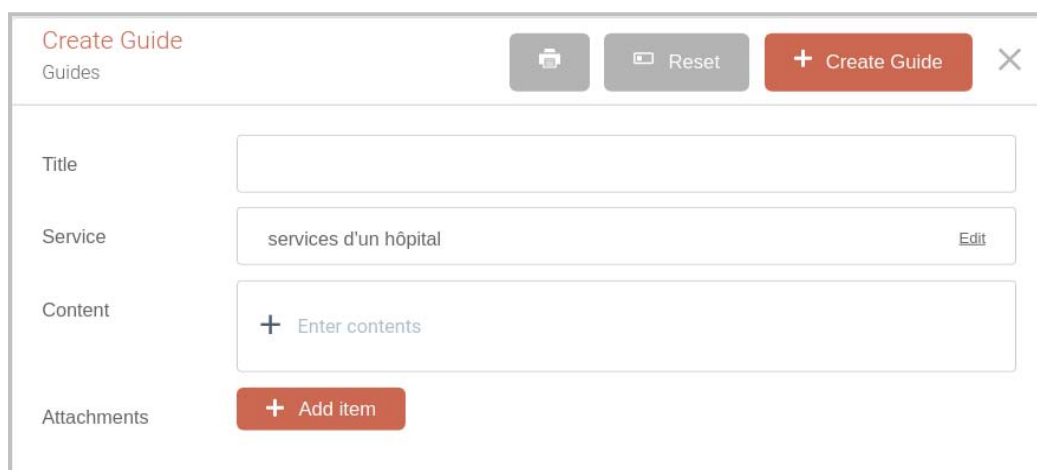
both worlds. It supports all conventional content types in rich text documents such as paragraphs, images, lists, tables, headings, links, comments, blocks, code highlighting ... etc.

Guides management follows the CRUD approach; guides are created, updated, deleted, or viewed.

Records listing are ordered by creation date from newer to older, listing displays department, patient name, patient gender, status, and response status.

Listing is available through the three conventional view types (Grid, Row, List), with the row view being the default view.

Records can be filtered, reordered, and navigated in real-time; actions can be carried on single items as well as on a bulk of items.



The screenshot shows a 'Create Guide' form with a title bar containing 'Create Guide', 'Guides', a printer icon, a 'Reset' button, a '+ Create Guide' button, and a close 'X' icon. The form has four sections: 'Title' with a text input field; 'Service' with a text input field containing 'services d'un hôpital' and an 'Edit' link; 'Content' with a large text area containing a '+ Enter contents' prompt; and 'Attachments' with a '+ Add item' button.

Figure 99: create guide action

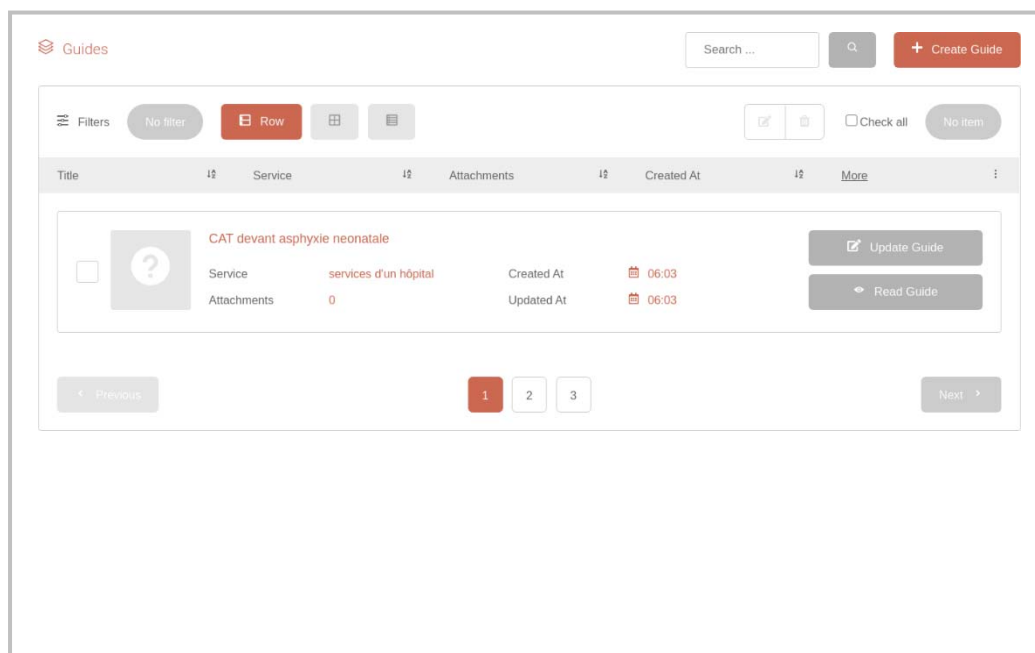


Figure 100: guide records listing

3. Multi-purpose content management system

Content consists of two parts: records and schemes; records are data entered by users, and schemes are the configuration of those data, much like a template or the rules that govern those data.

Therefore, content management is done through two processes, the first process is data definition, and the latter is data entry and manipulation.

From a technical viewpoint, data definition is considered as a configuration capture process, whereas data entry and manipulation are considered as the run-time of those captured configurations.

Data can take several forms, shapes, and volumes; it can be structured and unstructured, relational or not, linear, or non-linear.

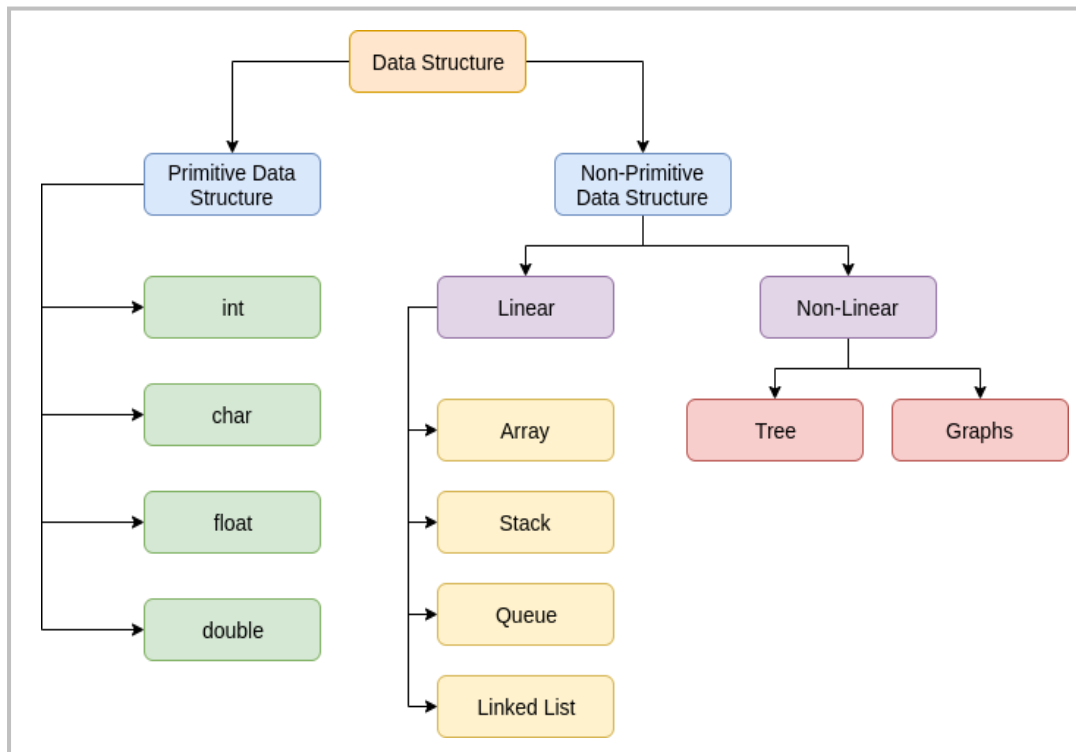


Figure 101: data structures hierarchy

It is mandatory to define what data shapes any content management system can capture or manage.

JavaScript Object Notation is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute-value pairs and array data types (or any other serializable value). It is a universal data format with a diverse range of applications [112].

JSON data structure is proven to be the most widespread data serialization standard; analyzing its data structure, we find that it's composed of primitive data structures and compound data structures [113].

JSON primitive data structures are numbers, strings, booleans, and null values, whereas compound data structures are arrays and objects [113].

Numbers can hold numeric data in the form of a double floating number; strings can hold textual data or binary data, booleans hold true/false values while null values are used to reference undefined values [113].

Arrays are considered to be a list of primitives or other compound values; objects are key-value pairs of primitives or other compound values; recursive references are not allowed JSON serializations [113].

Yet JSON data format has proven to be the most widespread data serialization standard. Still, it doesn't cover sufficiently certain data structures, e.g., files, blobs, date, and duration values are not directly supported.

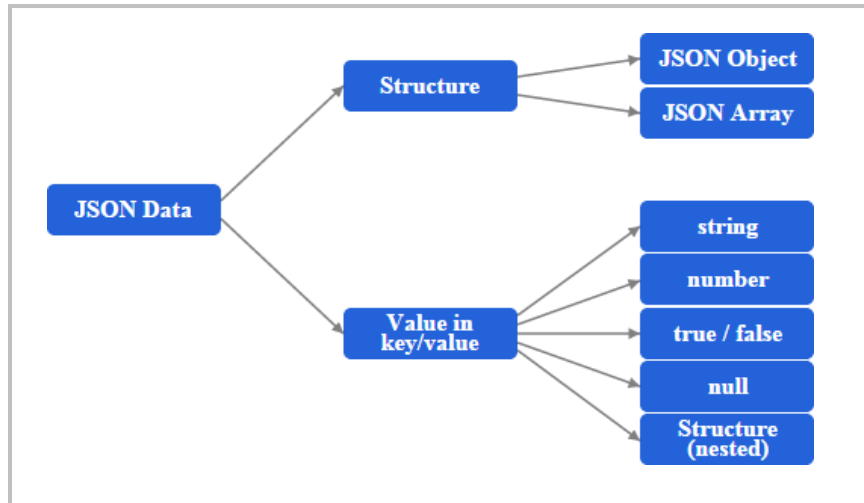


Figure 102: JSON data structures

A. Prerequisites

metatype and *metalambda* dependencies serve as the core of the content management system.

The most simplistic unit of abstraction is properties; properties are equivalent to primitives in the JSON data–structure counterpart. Nonetheless, they might be composed of union types or compound types.

Metadata and inference are essential concepts in which the content management system was built.

Properties are defined mainly by their expression value and their key name; Expressions are provided by *metalambda* and can express the fallback value, infer the property's datatype, and value dependencies.

Property value metadata is poly-morph, and combinations are endless; properties can be hidden, visible, disabled, enabled, read-only, prefilled, filled on time, dynamic, or static.

Datatypes are provided by *metatype*; *metatype* can express an endless combination of datatypes; it supports 14 primitive datatype, array, object, union, and intersection datatypes.

Table XXI: *metatype* primitive datatypes

Primitive	Description
any	Any datatype primitive or compound, be it
null	Accommodate undefined values
number	Accommodate any numeric value
integer	Accommodate whole numbers without the fractional component
float	Pretty much like a number datatype
string	Accommodate textual values
boolean	Accommodate true/false values
date	Accommodate date values
duration	Accommodate duration values
file	Accommodate file blobs
image	Accommodate image blobs
video	Accommodate video blobs
audio	Accommodate audio blobs
document	Accommodate document blobs

Datatype can be fine-grained thanks to attributes, patterns, and flags; nearly 24 attributes, 65 patterns are supported.

Properties can also be preconditioned, meaning that their visibility state is conditioned with an individual expression that evaluates a Boolean value.

metatype metadata data are used to infer datatype appearance and input state; appearance can be further customized thanks to datatype hints, e.g., one can infer a markup editor for a string primitive by instructing “*string#markdown*”.

Expressions allow us to infer many metadata information without having the user specifying them explicitly; They allow real-time synchrony between the fallback value, input type, and visual appearance.

B. Segment entity

Segments are the first layer of abstraction in the elastic scheme system; they are the

equivalent to the object compound data structure in the JSON counterpart.

Segments consist of key–value pairs or properties; properties have these metadata:

Table XXII: segment property metadata

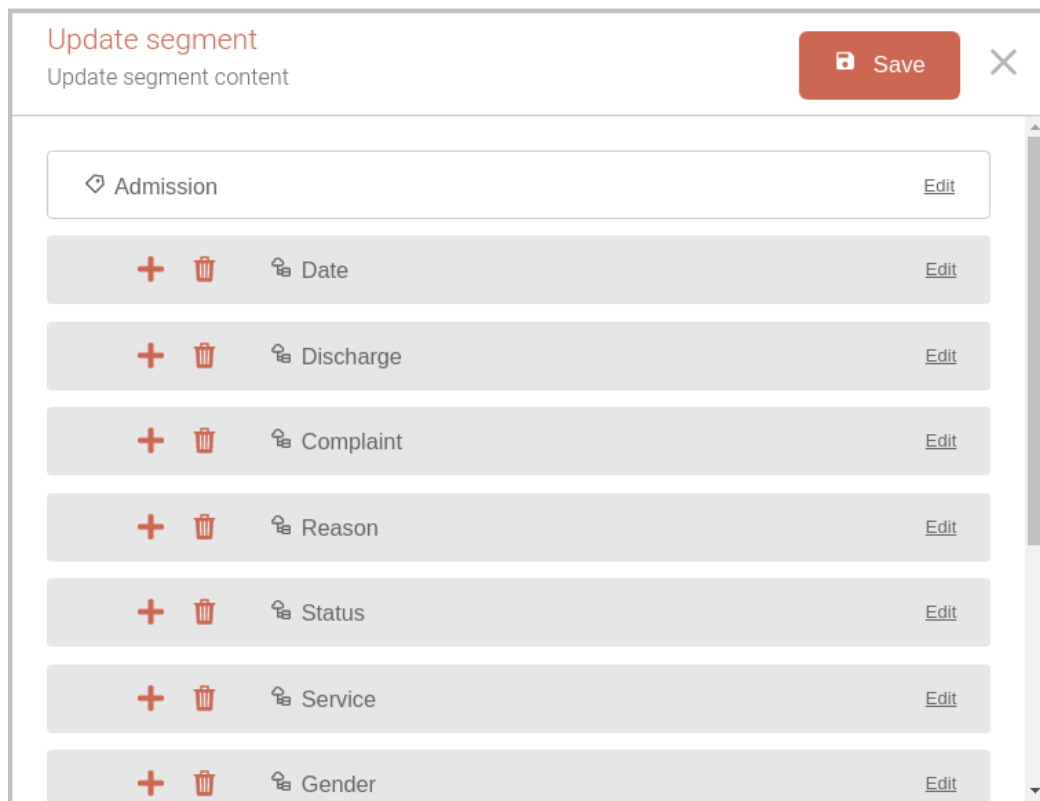
Key	Description
Key	Property's key name and its translations
Value	<i>metalambda</i> expression from which we infer that fallback value, property's datatype, and property dependencies
Appearance	Property's appearance in the run–time editor, appearances as input, hidden, read–only, or disabled are supported
Condition	Optionally a precondition metadata

Properties can be reordered thanks to the drag and drop interaction; a segment must have at least one property to be valid.

Each segment is identified by its key; keys must consist of alphanumeric or underscore characters only and must not be duplicated.

If a segment is referenced elsewhere, its key gets locked and cannot be deleted or renamed until the lock is released.

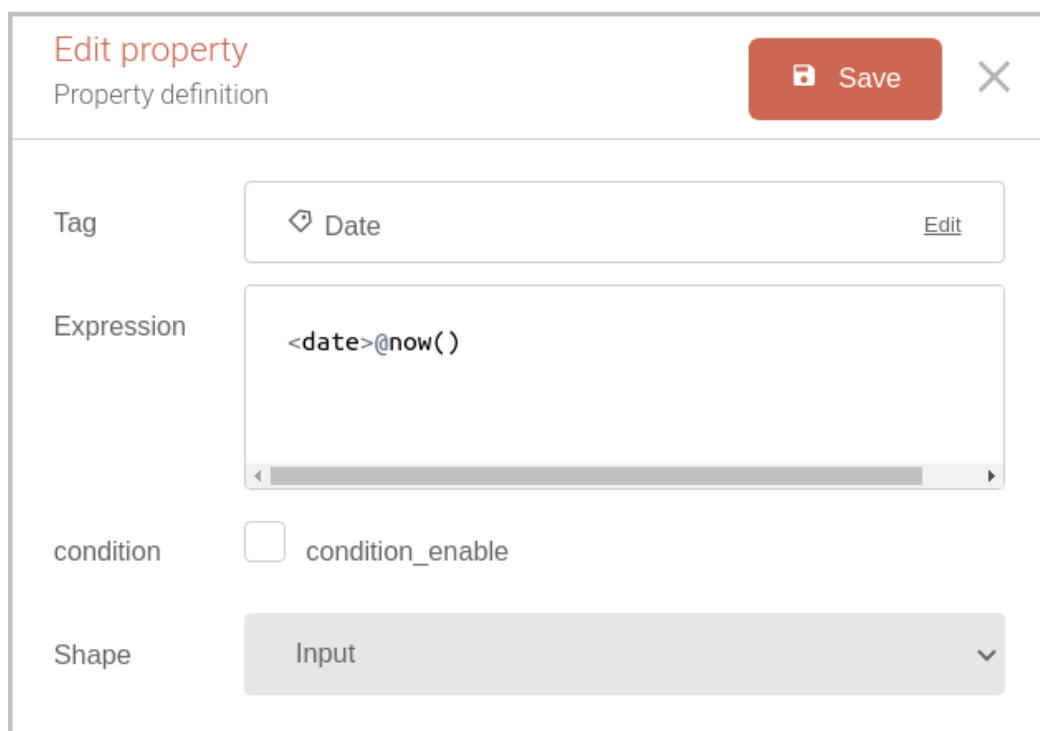
Segments listing display the name and properties count in addition to creation and last update dates; three conventional view variants are available (Grid, Row, and List) with the row variant as the default.



The 'Update segment' dialog box has a title bar with 'Update segment' in red and 'Update segment content' below it. On the right are a red 'Save' button and a close 'X' icon. The main area contains a list of segment entities. The first entity, 'Admission', is highlighted with a white background and has an 'Edit' link. The other entities are in grey bars: 'Date', 'Discharge', 'Complaint', 'Reason', 'Status', 'Service', and 'Gender'. Each entity has a red '+' icon, a red trash icon, a lock icon, and an 'Edit' link.

Entity	Actions
Admission	Edit
Date	+ Trash Lock Edit
Discharge	+ Trash Lock Edit
Complaint	+ Trash Lock Edit
Reason	+ Trash Lock Edit
Status	+ Trash Lock Edit
Service	+ Trash Lock Edit
Gender	+ Trash Lock Edit

Figure 103: segment entity editor



The 'Edit property' dialog box has a title bar with 'Edit property' in red and 'Property definition' below it. On the right are a red 'Save' button and a close 'X' icon. The main area contains four fields: 'Tag' with a dropdown showing 'Date' and an 'Edit' link; 'Expression' with a text area containing '<date>@now()'; 'condition' with a checkbox labeled 'condition_enable'; and 'Shape' with a dropdown menu showing 'Input'.

Tag	Date Edit
Expression	<date>@now()
condition	☐ condition_enable
Shape	Input

Figure 104: segment property editor

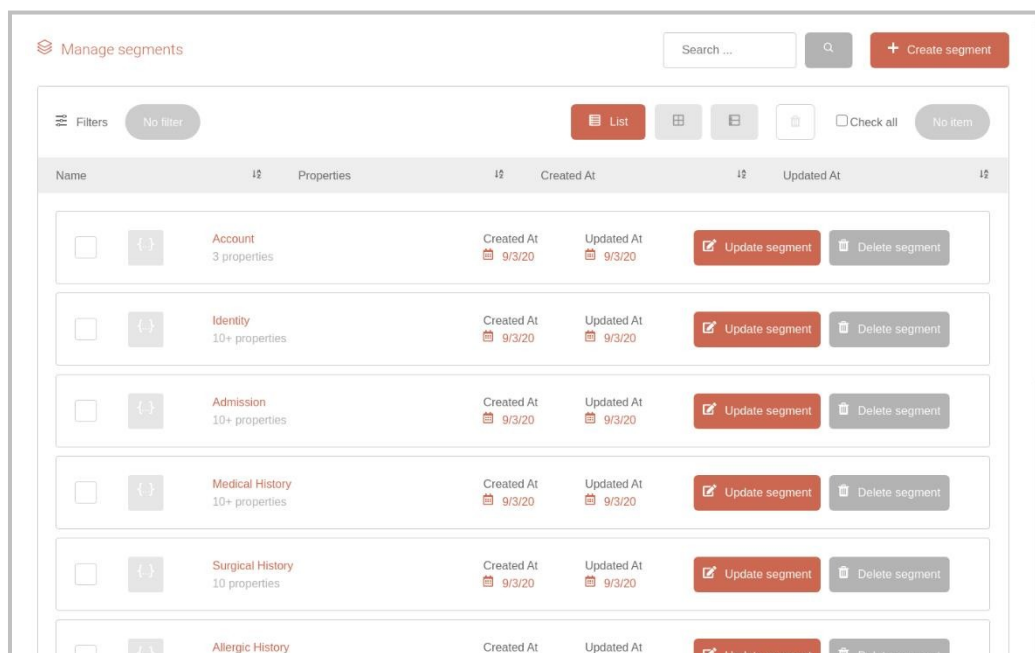


Figure 105: segment entities listing

C. Scheme entity

Schemes are the next layer of abstraction; schemes are considered the template that guideline and govern data towards its final constrained shape.

Five types of schemes are supported, schemes can reference segments or other schemes recursively, cyclic dependencies, and undefined references are not allowed.

Table XXIII: supported scheme types

Type	Description
Segment type	Reference segment entities directly; this type is the end type in the scheme tree
List type	The array equivalent in the JSON counterpart, list metadata are min count and optionally a max item count
Map type	The object equivalent in the JSON counterpart, a key-value association of other sub-schemes
Reference type	This type can reference another sub-scheme recursively
Relation type	Use this type to link between scheme records with a link reference, e.g., the relationship between patient records and admission records

Scheme appearances are hinted via appearance flags, appearances like table, tabs, accordions are supported; over 40 appearance flags are supported.

Each type has additional metadata:

- **Name:** optional or mandatory; if mandatory, then it is to assert scheme name, else then to override the reference name.
- **Appearance:** used to instruct scheme appearance directives.
- **Condition:** optionally a precondition metadata.

The screenshot displays a 'list scheme editor' interface with the following fields and controls:

- Name:** A text input field containing 'undefined_tag' with an 'Edit' link to its right.
- Condition:** A section containing a red square icon, the text 'Null', and a large empty rectangular box.
- Orderable:** A section containing an unchecked checkbox and the text 'Enable'.
- Appearance:** A dropdown menu currently showing 'Box'.
- Reference:** A text input field containing '{-} Segment type' with a red double-headed arrow icon and a red edit icon to its right.
- Min count:** A text input field containing the value '0'.
- Max count:** A section containing an unchecked checkbox, the text 'Infinite', and a text input field containing the value '1'.

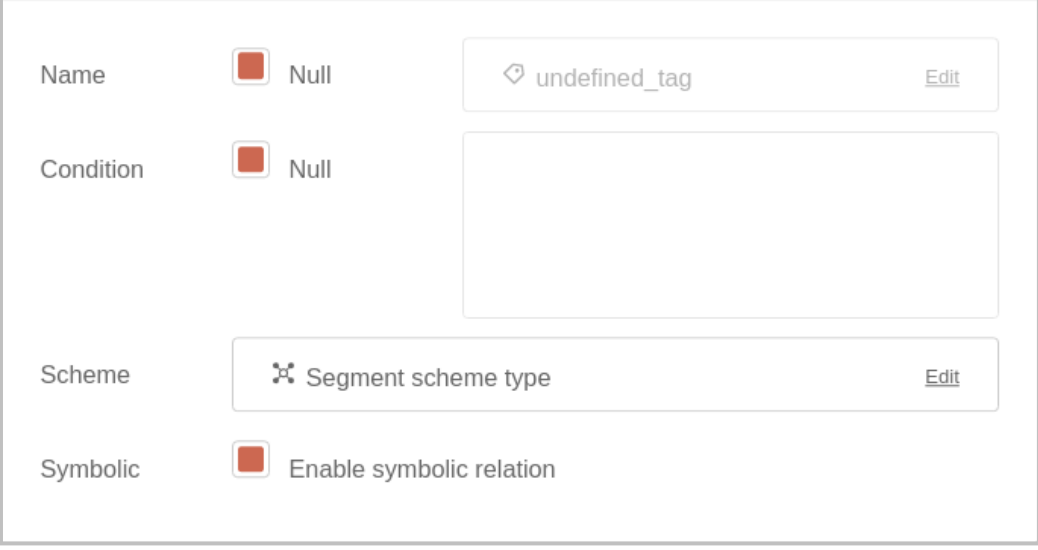
Figure 106: list scheme editor

Name	🔗 undefined_tag Edit	
Condition	☒ Null	
Orderable	☐ Enable	
Appearance	None	
Direction	Vertical	
References	+ 🗑 { } Segment type ↕ ✎ + Add reference	

Figure 107: map scheme editor

Name	☒ Null	🔗 undefined_tag Edit
Condition	☒ Null	
Scheme	🔗 Segment scheme type Edit	
Symbolic	☐ Enable symbolic relation	

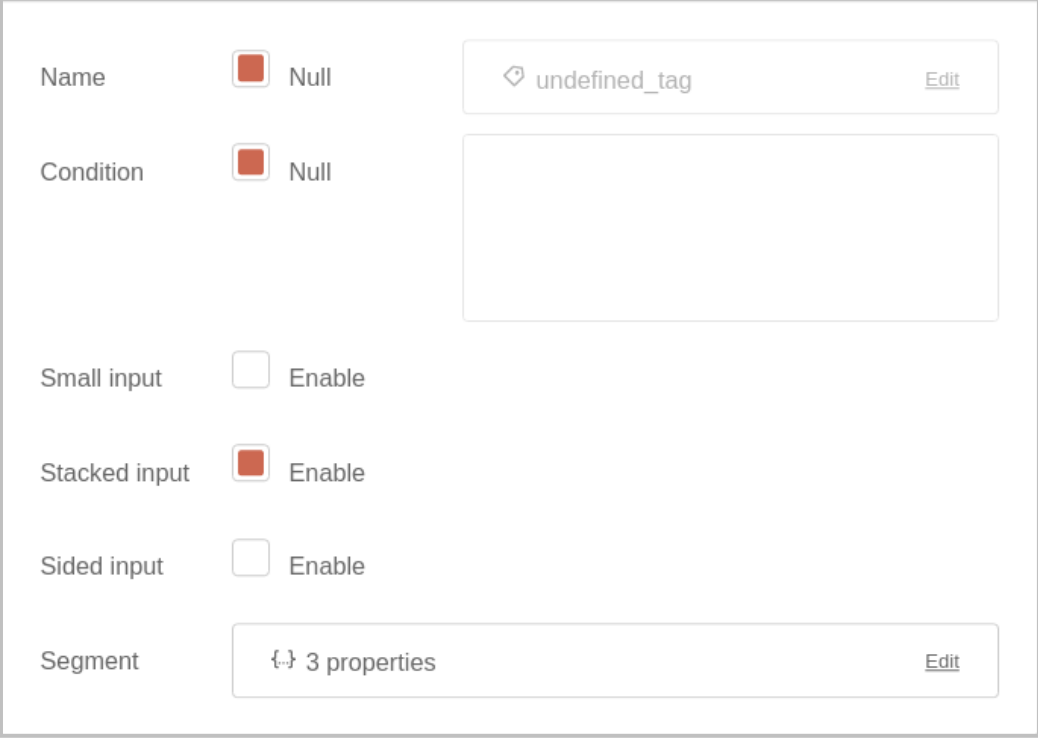
Figure 108: reference scheme editor



The figure shows a 'relation scheme editor' interface. It contains four rows of controls:

- Name:** A red square checkbox followed by the text 'Null'. To the right is a text box containing 'undefined_tag' and an 'Edit' link.
- Condition:** A red square checkbox followed by the text 'Null'. To the right is a large empty rectangular box.
- Scheme:** A button with a cross-hatch icon and the text 'Segment scheme type', followed by an 'Edit' link.
- Symbolic:** A red square checkbox followed by the text 'Enable symbolic relation'.

Figure 109: relation scheme editor



The figure shows a 'segment scheme editor' interface. It contains six rows of controls:

- Name:** A red square checkbox followed by the text 'Null'. To the right is a text box containing 'undefined_tag' and an 'Edit' link.
- Condition:** A red square checkbox followed by the text 'Null'. To the right is a large empty rectangular box.
- Small input:** An unchecked square checkbox followed by the text 'Enable'.
- Stacked input:** A red square checkbox followed by the text 'Enable'.
- Sided input:** An unchecked square checkbox followed by the text 'Enable'.
- Segment:** A button with a curly brace icon and the text '3 properties', followed by an 'Edit' link.

Figure 110: segment scheme editor

Each scheme is identified by its key; keys must consist of alphanumeric or underscore characters only and must not be duplicated.

If a scheme is referenced elsewhere, its key gets locked and cannot be deleted or renamed until the lock is released.

Schemes listing display the name, type, label, and precondition in addition to creation and last update dates; three conventional view variants are available (Grid, Row, and List) with the row variant as the default.

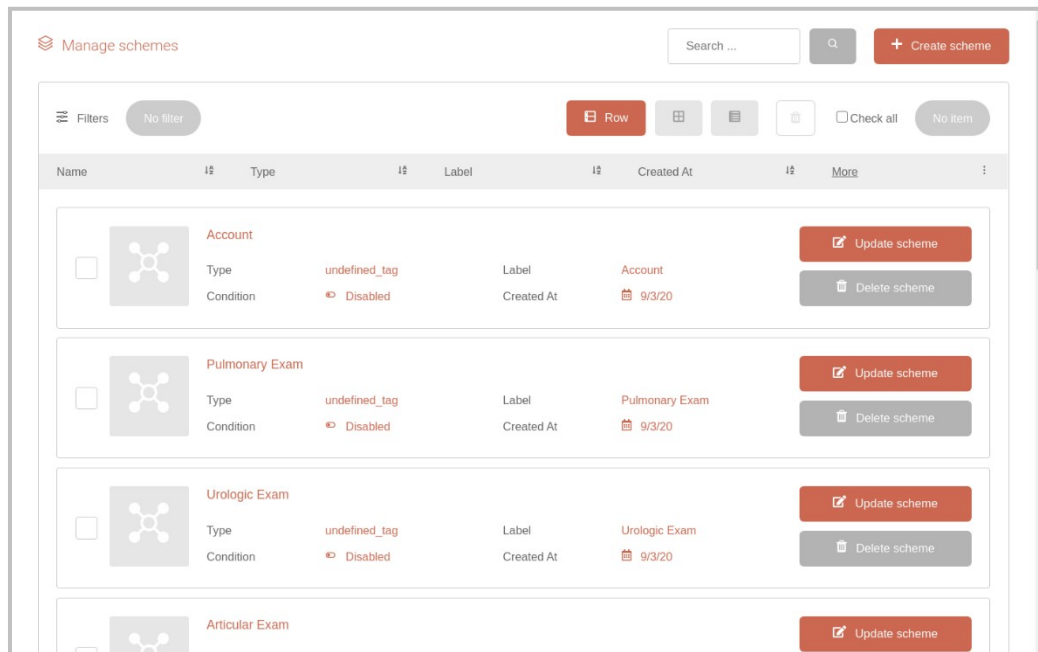


Figure 111: scheme entities listing

D. Action entity

Actions define how scheme records can be created, updated, deleted, or viewed; they instruct the runtime engine to expose consumption endpoints and instruct the front-end to expose the proper editing/reading interface.

Table XXIV: action entity types

Type	Description
Create	Inserting new records, this is a write-only action
Update	Updating an existing record, this is a write and read action
Delete	Delete an existing record; this is a read-write action
Read	Reading of an existing record, this a read-only action

Actions can be carried on a single record and a bulk of records; actions can require user feedback or can be run without user feedback.

The runtime editor takes concern of the display editing/reading interface of an action configuration over a scheme record.

Runtime editor support form resetting, content printing, progress auto-save, content validation, and relationship navigation.

Table XXV: action entity metadata

Key	Description
Height	Editor's window height in pixels
Width	Editor's window width in pixels
Icon	Button's icon art
Variant	Button's style variant
Message	On success callback message
Flags	Override appearance flags on matched paths with target values
Keys	Determine what properties are to be displayed as read-only
Inputs	Determine what properties are editable using match paths
Values	Automatically populate matched property paths with target values
Variant	Button's style variant

In practice, it is counter-intuitive to display "Keys" or "Inputs" for a delete action as the records will be deleted anyway. Therefore, metadata visibility follows an availability matrix that is based mainly on the action type.

Table XXVI: action entity metadata availability matrix

Action	Create	Update	Delete	Read
Key				
Height	Yes	Yes	No	Yes
Width	Yes	Yes	No	Yes
Icon	Yes	Yes	Yes	Yes
Variant	Yes	Yes	Yes	Yes
Message	Yes	Yes	Yes	Yes
Flags	Yes	Yes	No	Yes
Keys	No	Yes	No	Yes
Inputs	Yes	Yes	No	No
Values	Yes	Yes	No	No

Create Patient

Edit

+ Create type

Width

900

▼

Height

500

▼

Icon

Edit media

Edit

Variant

Primary

▼

Message

undefined_tag

Edit

Flags

+ Add flag

Inputs

+ 🗑

Path

**

Values

+ Add value

Figure 112: action create type

Update Patient

Edit

Update type

Width

900

Height

500

Condition

☐ Enable

Icon

Edit media

Edit

Variant

Secondary

Message

undefined_tag

Edit

Flags

+ Add flag

Keys

+ Add key

Inputs

+

Path

**

Values

+ Add value

Figure 113: action update type

✕ Delete Patient

Edit

🗑 Delete type

⬆ ⬇ ✎

Width

400

▼

Height

200

▼

Condition

☐ Enable

Icon

🖼 Edit media

Edit

Variant

Primary

▼

Message

✕ undefined_tag

Edit

Figure 114: action delete type

The screenshot shows a configuration form for an action named 'Read Patient'. The form is organized into sections. The top section is a header bar with the action name and an 'Edit' link. Below this is a 'Read type' section with a toggle icon and a pencil icon. The main configuration area includes: 'Width' and 'Height' dropdown menus set to 900 and 500 respectively; a 'Condition' checkbox labeled 'Enable' which is currently unchecked; a large empty text area; an 'Icon' section with a media icon, the text 'Edit media', and an 'Edit' link; a 'Variant' dropdown menu set to 'Secondary'; a 'Flags' section with a red bar and a '+ Add flag' button; and a 'Keys' section with a '+', a trash icon, a 'Path' label, and a text input field containing '**'.

Figure 115: action read type

Each action is identified by its key; keys must consist of alphanumeric or underscore characters only and must not be duplicated.

If an action is referenced elsewhere, its key gets locked and cannot be deleted or renamed until the lock is released.

Actions listing display the name, type, and keys count in addition to creation and last update dates; three conventional view variants are available (Grid, Row, and List) with the row variant as the default.

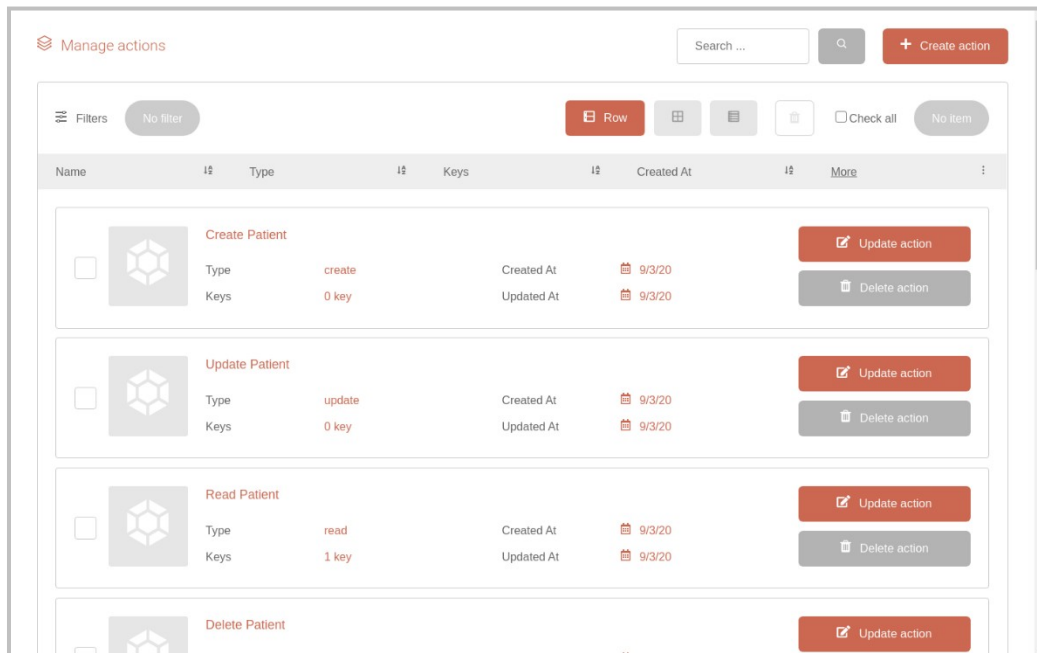


Figure 116: action entities listing

E. Variant entity

A scheme can have as many query variants as required; scheme records for a target scheme are all saved to the same corresponding datatable no matter through what variant were added.

It is up to the content master to filter saved records to obtain a variant data view: Queries support transformation, filtering, ordering, limitation, and grouping operations.

The returned query result must be a list of objects, each object must expose an “id” property, and its value must be unique within the list.

Table XXVII: variant entity metadata

Key	Description
Key	Variant key name and its translations
Scheme	Target records scheme
Vector	Vector media art for empty results
Query	<i>metalambda</i> expression that defines the query request
Paragraph	Paragraph or description tag key and its translations

The screenshot shows a web-based interface for editing a variant entity. At the top, there is a header bar with a back arrow icon, the text "Patients", and an "Edit" link. Below this, the main content area is divided into several sections: "Scheme" with a "Segment scheme type" field and an "Edit" link; "Variant" with a dropdown menu currently showing "No variant"; "Vector" with an "Edit media" field and an "Edit" link; "Query" with a text area containing a JSON query; and "Paragraph" with a text field containing "The list of all patients wheter admitted or not" and an "Edit" link. The query text is:

```
input -> sort("desc", (item) => <integer>item.id) -> map((item) => { id: item
.id, name: item.value.item.full_name, gender: item.value.item.gender ->
interpolate(), birth_date: item.value.item.birth_date, age: <duration>@(now
() - item.value.item.birth_date), create: item.create, update: item.update,
})
```

Figure 117: variant entity editor

Each variant is identified by its key; keys must consist of alphanumeric or underscore characters only and must not be duplicated.

If a variant is referenced elsewhere, its key gets locked and cannot be deleted or renamed until the lock is released.

Variants listing display the name, scheme, and vector art key in addition to creation and last update dates; three conventional view variants are available (Grid, Row, and List) with the row variant as the default.

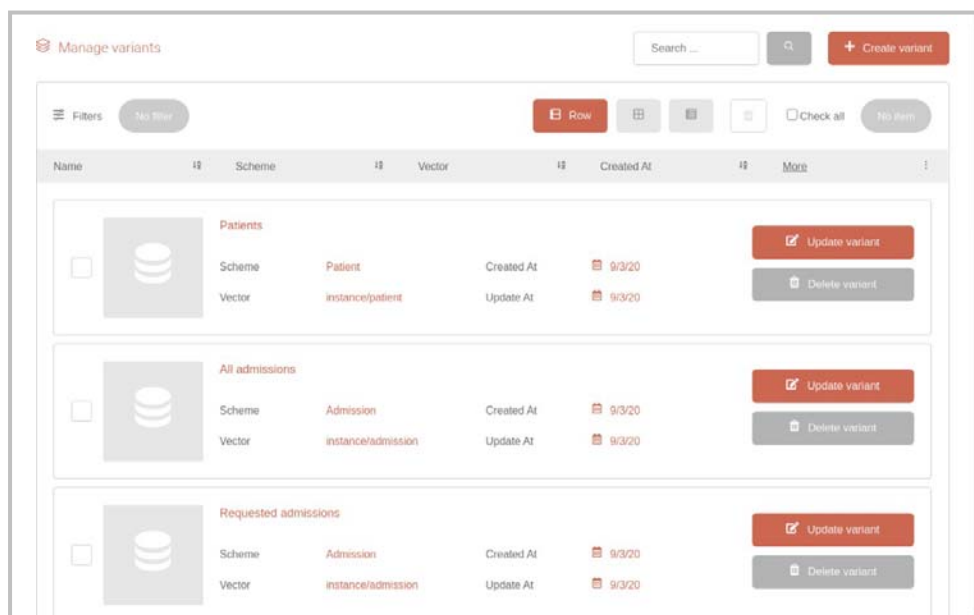


Figure 118: variant entities listing

F. Item entity

Items are single record single action entities, they always have a fallback value, and they are always update-only single records; items can be suitable for configuration endpoints or to make calculator interfaces.

Items follow a hybrid metadata configuration set between variant entities and action entities, they always display on page with full width and height without a popup window, and they always have a success callback message.

Table XXVIII: item entity metadata

Key	Description
Key	Item key name and its translations
Scheme	Target records scheme
Message	On success callback message
Flags	Override appearance flags on matched paths with target values
Keys	Determine what properties are to be displayed as read-only
Inputs	Determine what properties are editable using match paths
Values	Automatically populate matched property paths with target values

The screenshot shows the 'Newborn Score' item entity editor. At the top, there's a header bar with the item name 'Newborn Score' and an 'Edit' link. Below this, the editor is organized into several sections: 'Scheme' with a 'Map scheme type' button and an 'Edit' link; 'Message' with an 'undefined_tag' and an 'Edit' link; 'Flags' with a '+ Add flag' button; 'Keys' with a '+ Add key' button; 'Inputs' which contains a sub-editor with a '+', a trash icon, a 'Path' label, and a text input field containing '**'; and 'Values' with a '+ Add value' button. The interface uses a clean, modern design with a light gray background and orange-red buttons.

Figure 119: item entity editor

Each item is identified by its key; keys must consist of alphanumeric or underscore characters only and must not be duplicated.

If an item is referenced elsewhere, its key gets locked and cannot be deleted or renamed until the lock is released.

Items listing display the name, keys count, flags count, values count, and inputs count in addition to creation and last update dates; three conventional view variants are available (Grid, Row, and List) with the row variant as the default.

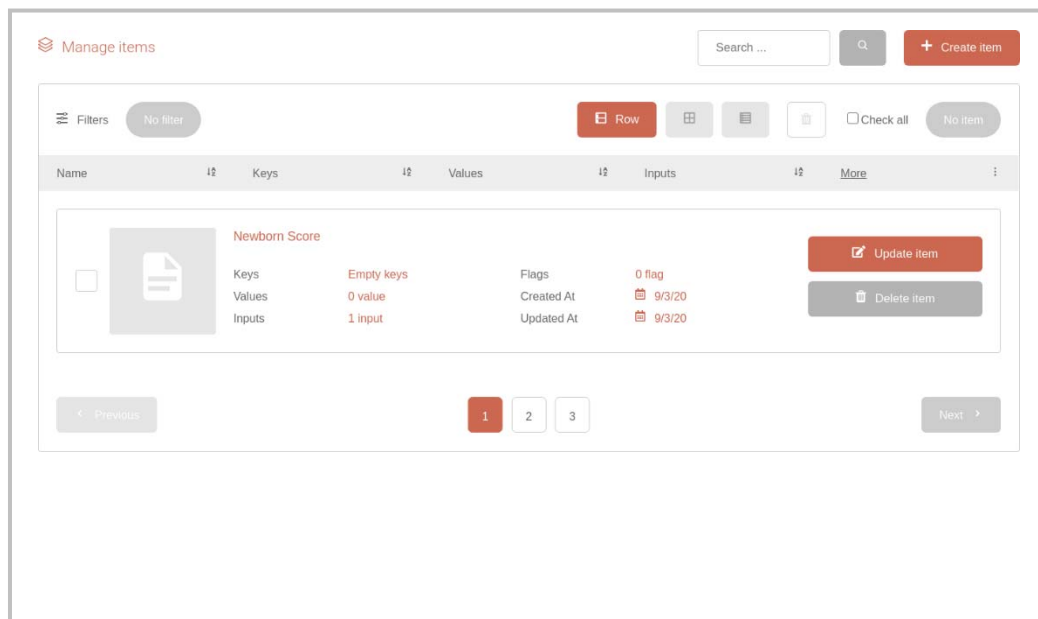


Figure 120: item entities listing

G. View entity

View entity is the uppermost layer of abstraction, but a simple entity that serves a simple purpose: Instruct the listing component and map variants with actions altogether.

This entity is metadata-rich; it captures listing elements positioning, variants mapping, action mapping, types configurations, and viewers configurations.

Table XXIX: view metadata components

Component	Description
Listing elements position	Instruct where and if a listing component element should be displayed or not
Variants mapping	Map a set of variants with a view entity
Actions mapping	Map a set of actions with a view entity
Types configurations	Configure and map result keys into listing card elements
Viewers configurations	Configure card appearances and viewers

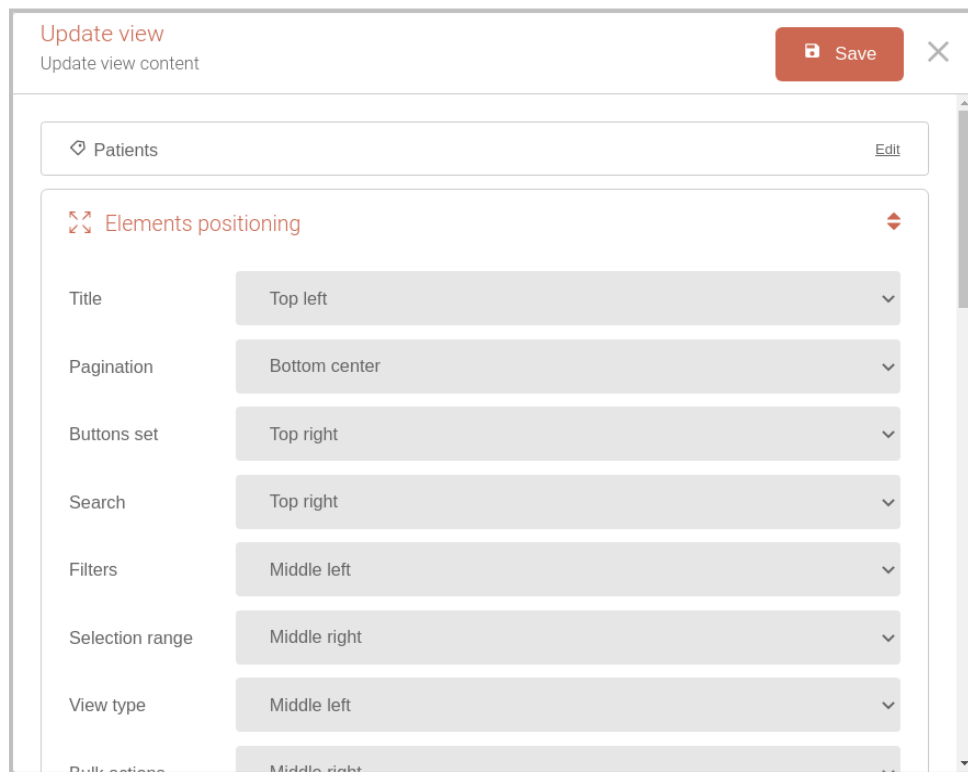


Figure 121: view entity editor

The listing component allows a wide array of operations to be carried on records such as filtering, reordering, pagination ... etc.

Table XXX: listing component supported operations

Operation	Description
Filtering	Keeping only relevant result to a filters query
Ordering	Sorts result against a key-value in a specific order direction
Pagination	Navigates listing records through the pagination system
Checking	Checking a selection of records to carry a bulk action
Search	Runs a full-text search or a search by id.
View switch	Switch display type
Range checking	Checks or uncheck a range of records

Table XXXI: listing component elements

Element	Description
Title	component title
Pagination	Pagination component
Buttons set	Call-to-action or create-actions buttons
Search	Natural language full-text search
Filters	Filtered keys and filters count
Selection range	Range checking component
View type	Switch display type
Bulk actions	Bulk actions that can be carried on a selection of records

Variants mapping enabled the content master to link similar query variants of the same scheme altogether; variants must point to the same scheme and must be unique.

Actions mapping link data manipulation layer with data definition layer, actions must be unique.

Types configurations map variant query's results with listing card elements and define how the listing would appear.

Several types are supported to cover a wide range of possibilities; combinations are endless; each type configuration has its specific metadata to enable further customization.

Table XXXII: listing types configurations

Type	Description
Title type	Instruct that a property is of title type
Subtitle type	Instruct that a property is of subtitle type
Icon	Define the card component's icon media art
Text	Instruct that a property is of textual type
Date	Instruct that a property is of date type, several formats are supported
Duration	Instruct that a property is of duration type, several formats are supported
Number	Instruct that a property is of numeric type, several formats are supported
Boolean	Instruct that a property is of boolean type; True/False, On/Off, Valid/Invalid, Yes/No, and Up/Down formats are supported

The screenshot shows a web-based configuration editor titled "Edit type" with a subtitle "Edit type". In the top right corner, there is a red "Save" button with a floppy disk icon and a close button (X). The main content area displays the configuration for the "Duration type", which is indicated by a curly brace icon and the text "Duration type" at the top of the configuration box. To the right of this header are two small red icons: a double-headed vertical arrow and a pencil icon. Below the header, there are two configuration fields. The first field is labeled "Label" and contains a text input with the value "Age" preceded by a small icon; an "Edit" link is visible to the right of the input. The second field is labeled "Format" and contains a dropdown menu currently showing "Default" with a downward arrow icon.

Figure 122: type configuration editor

Viewers define how the listing component should display card components; three types of viewers are supported.

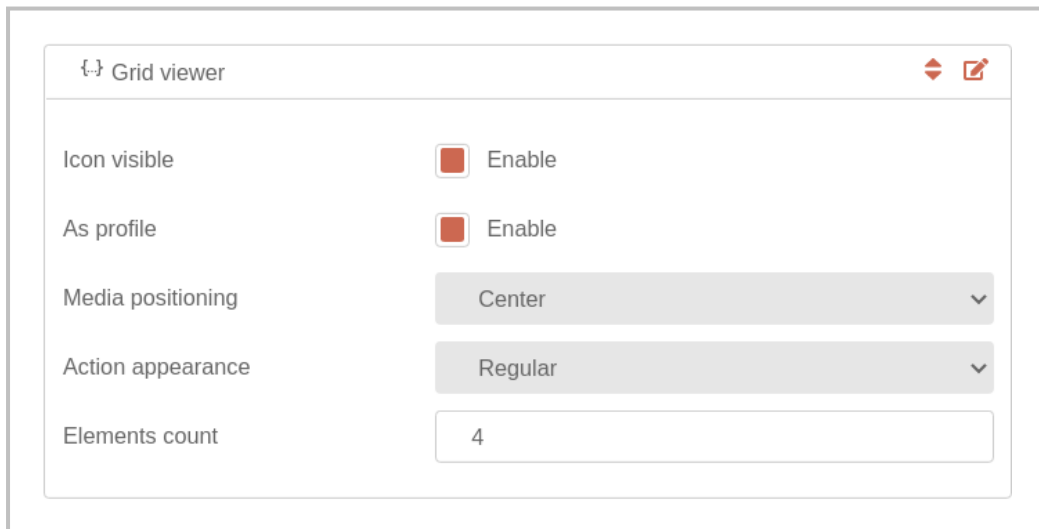
Table XXXIII: supported viewer types

Type	Description
Grid	display card components in a grid manner, four card components in a row, can display many buttons
Row	display card components in a row manner; rows are fatter than lists but can display no more than two buttons
List	display card components in a list manner; lists are thinner than rows but can display many buttons

Each viewer has its configurations; combinations are endless; each configuration requires a certain amount of type configurations and types of certain kinds to be valid.

Table XXXIV: grid viewer metadata

Key	Description
Icons visible	Define the card's icons visibility
As profile	Enable/disable a title/subtitle media pair style
Media position	Define the card's media art positioning
Action appearance	Defines buttons set appearance
Elements count	Defines exact card elements count to be displayed

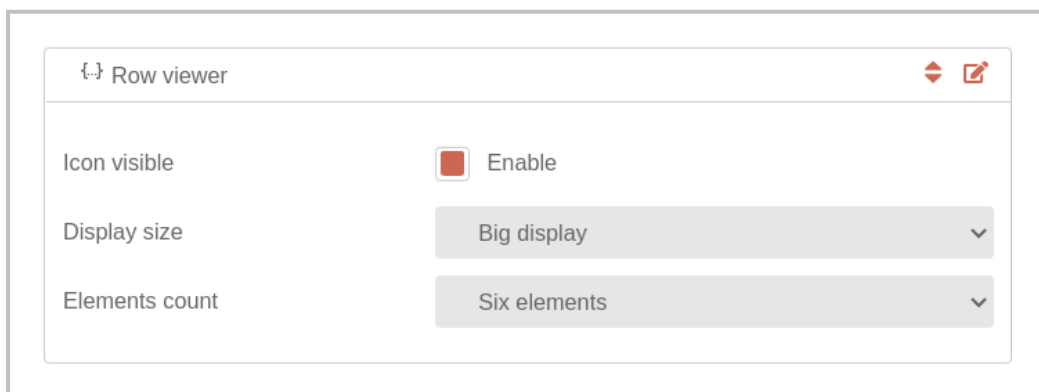


The image shows a 'Grid viewer' editor window. It has a title bar with a toggle icon and a close icon. The main area contains five settings: 'Icon visible' with an 'Enable' checkbox, 'As profile' with an 'Enable' checkbox, 'Media positioning' with a dropdown menu set to 'Center', 'Action appearance' with a dropdown menu set to 'Regular', and 'Elements count' with a text input field containing the number '4'.

Figure 123: grid viewer editor

Table XXXV: row viewer metadata

Key	Description
Icons visible	Define the card's icons visibility
Display size	Defines row size; small, medium, and big sizes are supported
Elements count	Defines exact card elements count to be displayed



The image shows a 'Row viewer' editor window. It has a title bar with a toggle icon and a close icon. The main area contains three settings: 'Icon visible' with an 'Enable' checkbox, 'Display size' with a dropdown menu set to 'Big display', and 'Elements count' with a dropdown menu set to 'Six elements'.

Figure 124: row viewer editor

Table XXXVI: list viewer metadata

Key	Description
Icons visible	Define the card's icons visibility
As profile	Enable/disable a title/subtitle media pair style
Label visible	Define the card elements' label visibility

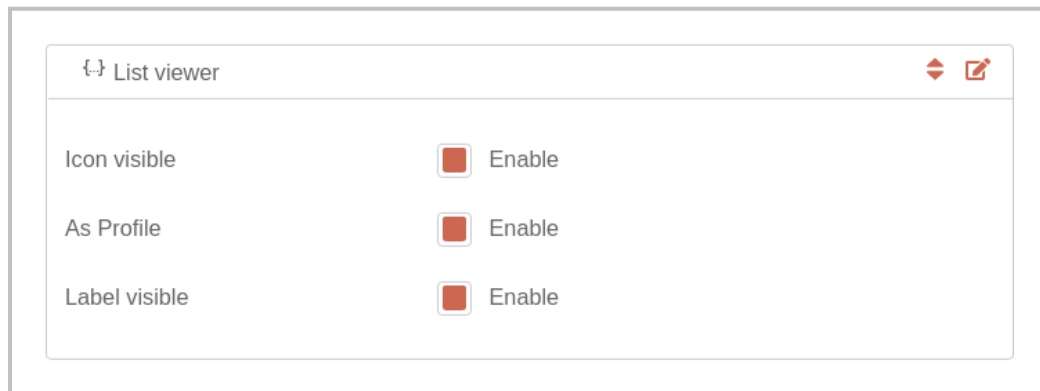


Figure 125: list viewer editor

To make a valid view entity, these conditions must be asserted:

- At least one variant.
- At least one non-create action.
- At least one type configuration.
- At least one viewer configuration.
- Variants must not be duplicated.
- Actions must not be duplicated.
- All variants must point to the same scheme.
- Types configurations must match every viewer configuration.

Each view is identified by its key; keys must consist of alphanumeric or underscore characters only and must not be duplicated.

If a view is referenced elsewhere, its key gets locked and cannot be deleted or renamed until the lock is released.

View listing displays the variants count, actions count, types count, and viewers count in addition to creation and last update dates; three conventional view variants are available (Grid, Row, and List) with row variant as the default.

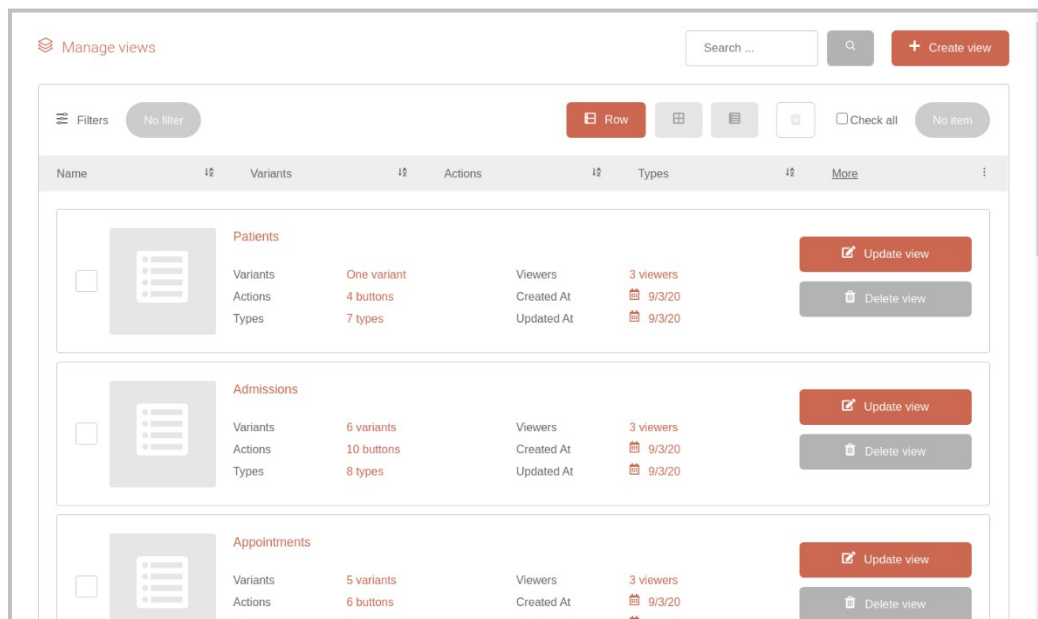


Figure 126: view entities listing

H. Layout entity

Layouts help organize views and items into menu items; similar entities can be grouped together; grouping can be of any flavor.

Links will be created automatically, and menu items will be added according to group contents; links are deduplicated and shortened automatically.

Links shortening uses an algorithm based on the most frequent prefix/suffix sub-string; this sub-string will be stripped out if a predefined frequency threshold was hit.

Table XXXVII: layout entity metadata

Type	Description
Icon	Sidebar icon art
Description	Sidebar item description
Entities	Mapping of items and/or views

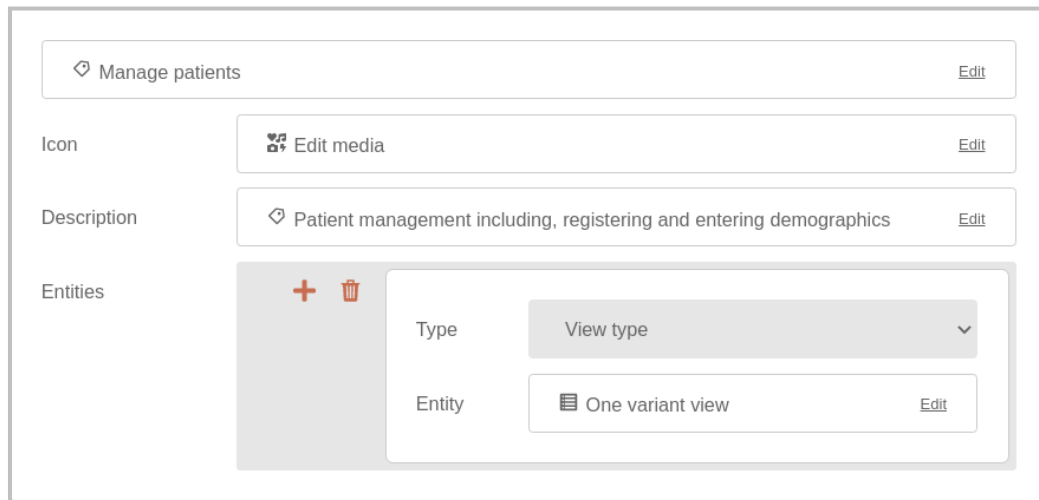


Figure 127: layout entity editor

Each layout is identified by its key; keys must consist of alphanumeric or underscore characters only and must not be duplicated.

If a layout is referenced elsewhere, its key gets locked and cannot be deleted or renamed until the lock is released.

Layouts listing displays the icon art key and entities count. In addition to creation and last update dates, three conventional view variants are available (Grid, Row, and List), with the row variant as the default.

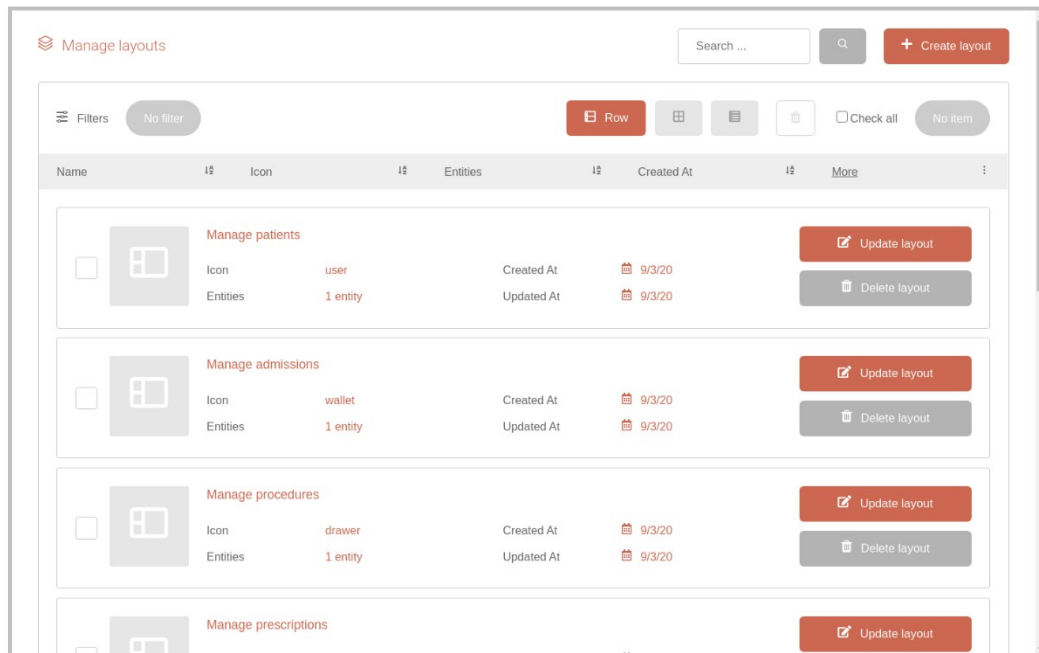


Figure 128: layout entities listing

I. Editor component

After defining the data structure, the front-end runtime component is responsible for displaying action runtimes and records listing.

Action runtime was designed to be as interactive as possible while maintaining fault-tolerance.

Table XXXVIII: user experience principles followed when crafting the editor component

Consideration	Description
Feedback	Immediate validity feedback
Consistency	Visual appearance consistency
Tolerance	Tolerance of faulty inputs
Printing	Every element in the editor is print-ready and have a custom print appearance
Accessibility	Easy navigation between inputs and short filling duration
Checkpoints	Ability to save progress without exiting the editor window
Read-only	Read-only variants for reading actions
Accommodation	Long labels must be accommodated within small spaces

Data entered in the editor are immediately validated, accurate error messages will be displayed on erroneous inputs so users can take immediate action.

Inputs and fields are consistent across datatypes thanks to *metatype* metadata, yet inputs do flex/blend to fit within their containers.

Printability considers that a partial content print may be requested, yet partial printing must not cut off parts of the scheme; Therefore, scheme types are the smallest unit of printable content.

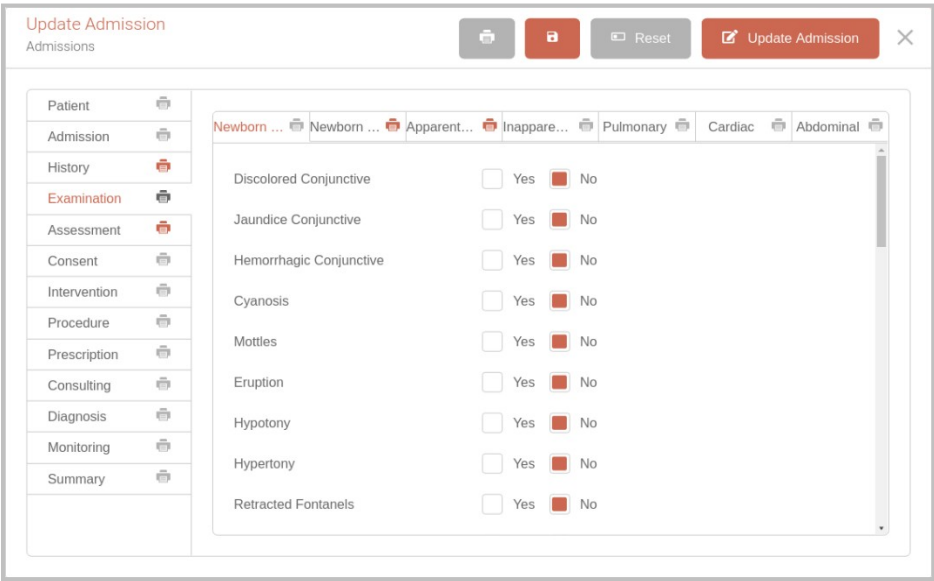


Figure 129: partial printing example

One can easily select to print a segment, a relation, or a whole scheme tree by clicking on the print icon near the scheme unit title.

A QR code will be automatically generated containing metadata that allows the reproduction of exact print results.

Subject (username), role, date, entity, and reference metadata will be automatically added to the print page's header along with the application logo and header text; Footer will consist of the footer.

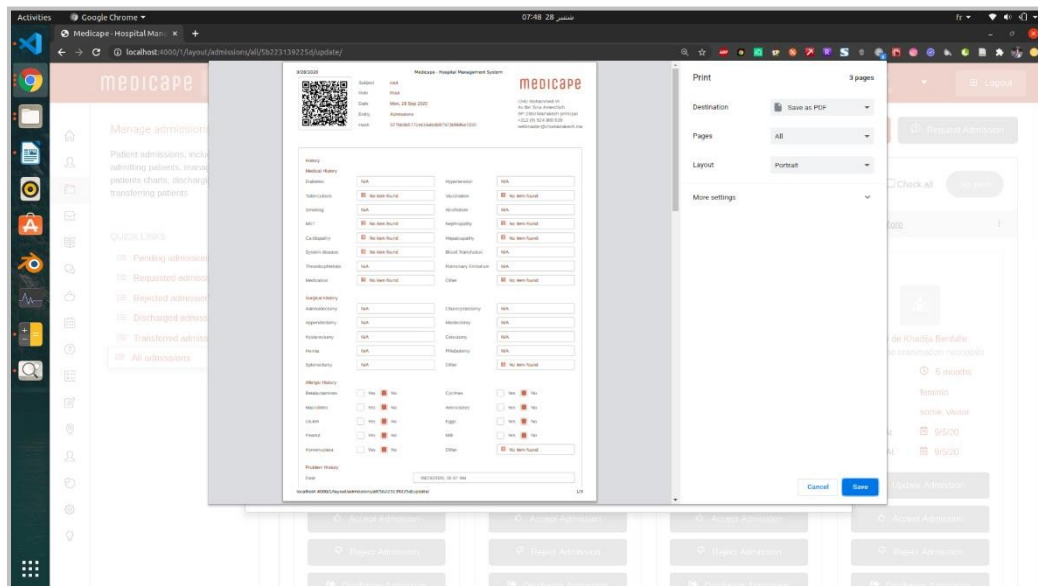


Figure 130: printing result example

J. Listing component

This component is a pivotal component in the front-end runtime abstraction tree; all entities and scheme records are displayed through this component.

It accommodates a wide array of use cases; it allows for consistent listing appearance but flexible to list all entities and scheme records.

Table XXXIX: supported listing component operations

Operation	Description
Filtering	Keeping only relevant result to a filters query
Ordering	Sorts result against a key-value in a specific order direction
Pagination	Navigates listing records through the pagination system
Checking	Checks a selection of records to carry a bulk action
Search	Runs a full-text search or a search by id
View switch	Switch display type
Range checking	Checks or uncheck a range of records

Supported filters are:

- Numeric range filters.
- Date range filters.
- Duration range filters.
- Exact match filter.
- Search match filter.

Figure 131: listing filters component

Sortable keys are title, subtitle, text, date, duration, numeric and Boolean types; Two sort directions are supported: ascending and descending; One can access directly to filters by right-clicking on sort keys.

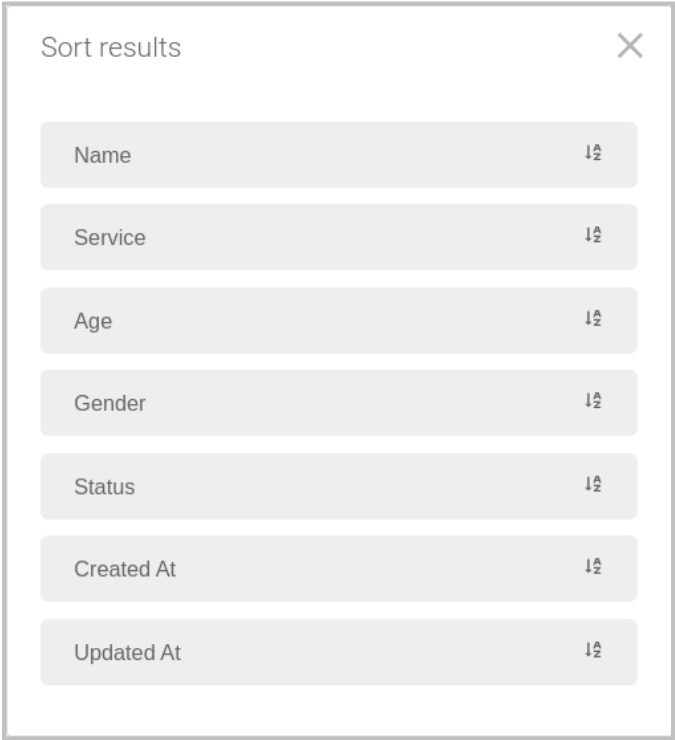


Figure 132: listing sorting component

Views switch can accommodate an endless amount of views; if the available views fit exactly the three conventional views pattern (3 views respectively, grid, row, and list), it will be shown in a button switch appearance; otherwise, it will be displayed on a drop-down selection list.

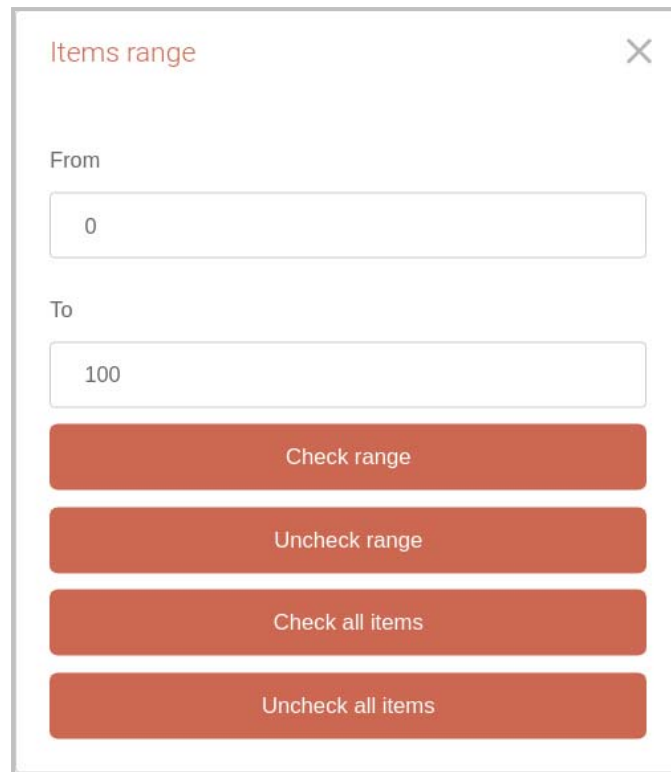


Figure 133: listing range checking component

K. File and storage management

metatype dependency is an extremely powerful dependency in terms of file upload and file management; It allows for automatic file transcoding, metadata extraction, file validation, and type coercion; Once a file is uploaded, its metadata will be automatically extracted, accurately two metadata types are extracted:

- **Generic blob metadata:** such as file size, extension, mime type, and upload date.
- **File type-specific metadata:** metadata extracted by the corresponding type transcoder, e.g., for an image type metadata would be like width, height, depth, channels ... etc.

File validation can leverage metadata that are both generic and specific; tolerance degree can be defined accordingly.

In case a file that was validated but only through a large degree of tolerance, it will be coerced asynchronously, e.g., an image that is larger than the specified dimension will be downsized to match specified dimensions asynchronously.

Files can be uploaded from 18 different providers; Providers include locale filesystem, camera, direct URL, and social networks such as Facebook, Twitter ... etc.

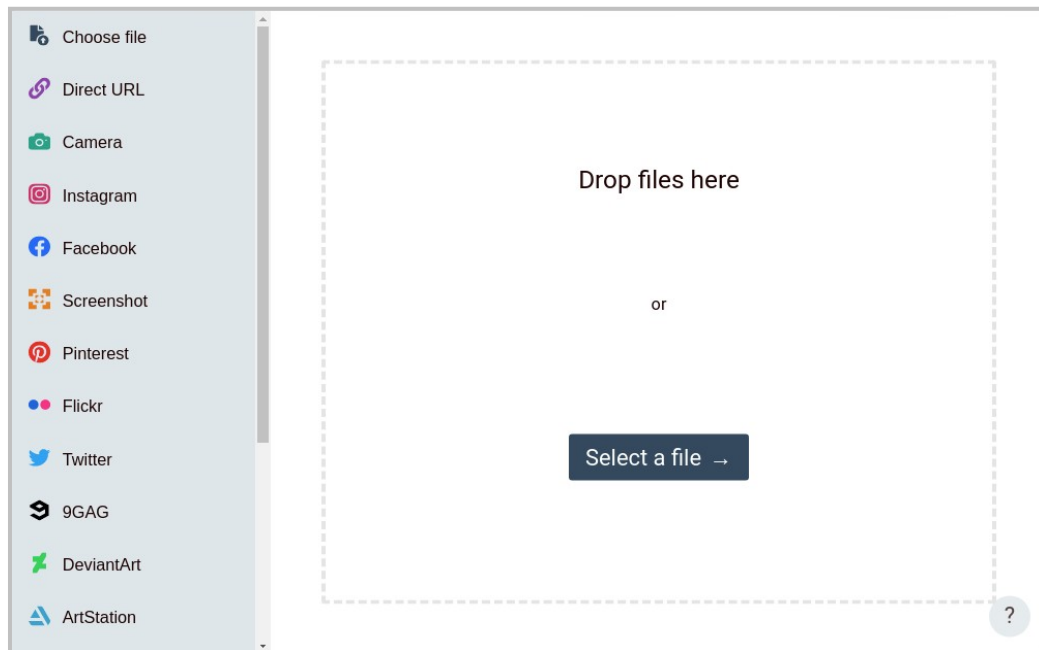


Figure 134: file upload component

Selected images can be transformed on the fly before being uploaded; Supported transformations are cropping, resizing, labeling, rotation, color transformations ... etc.



Figure 135: image on-the-fly transformation

Uploaded file path and metadata are serialized in a string format and digitally signed with a private key to prevent tampering; Metadata can be viewed on file metadata viewer.

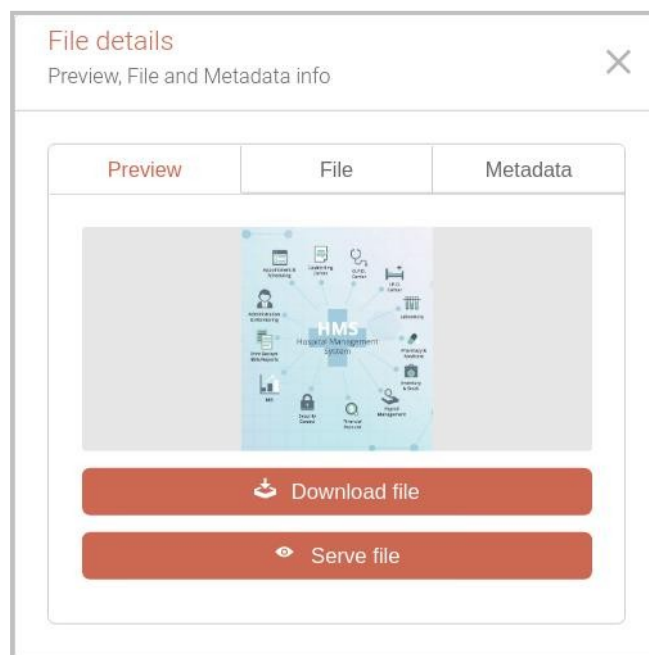


Figure 136: upload file preview

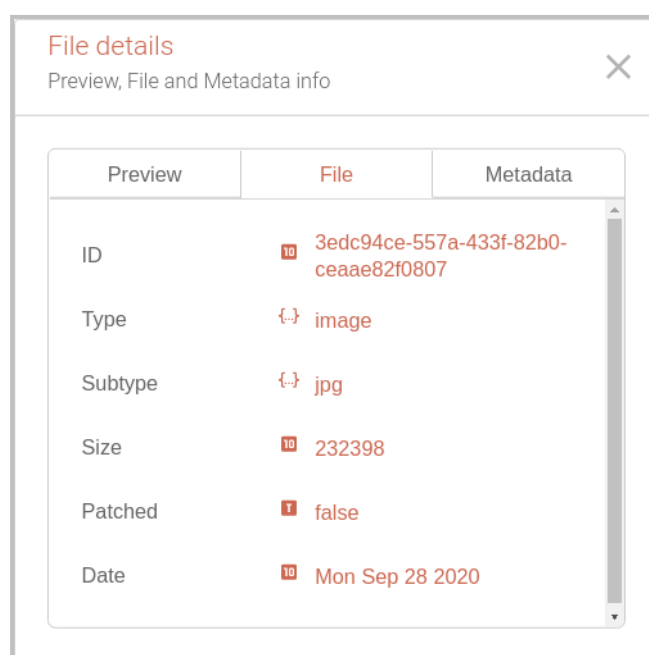


Figure 137: upload file details

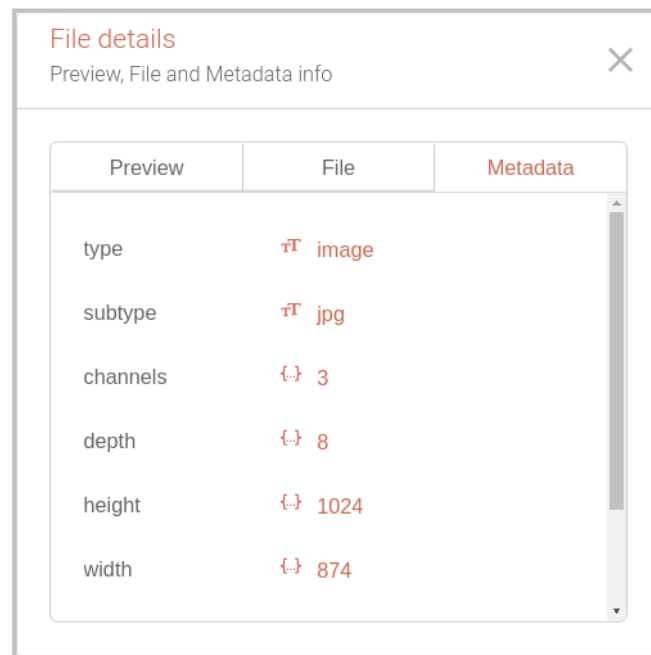


Figure 138: upload file metadata

To add file upload support to a segment, one can just enter a file or image as an input datatype, and the framework will take care of the rest.

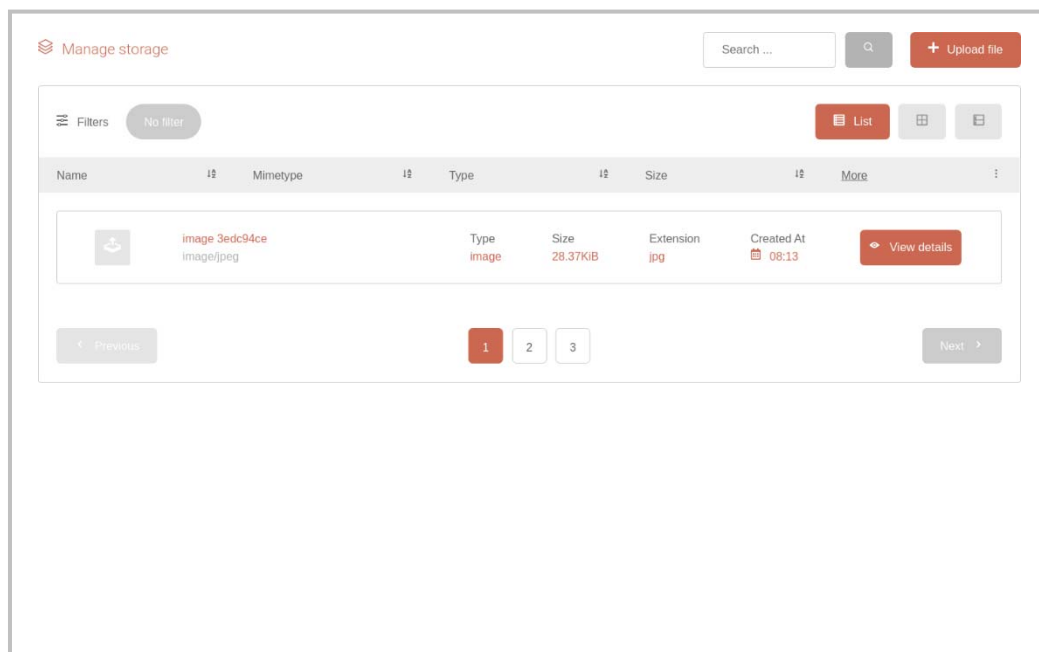


Figure 139: storage items listing

4. Identity, access management, and security

Access control (AC) is the selective restriction of access to resource while access management describes the process. The act of accessing may mean consuming, entering, or using. Permission to access a resource is called authorization [114].

A. Role-based access control

Role-based access control (RBAC) is a policy-neutral access-control mechanism defined around roles and privileges. The components of RBAC role-permissions and user-role make it simple to perform user assignments. RBAC can facilitate the administration of security in large organizations with hundreds of users and thousands of permissions [115,116].

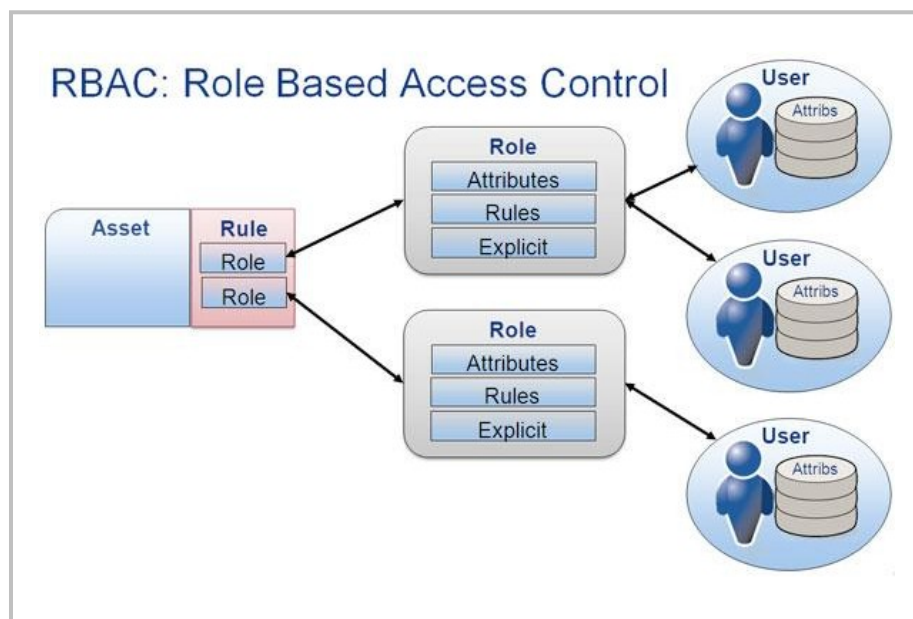


Figure 140: role-based access control

Each user is assigned a role on registration time; this role may later be updated via user management entity.

Each role defines a set of grants from the list of available grants, grants for content, settings, language, data, identity, and runtime management.

Grants are generated on runtime by merging predefined grants for the CMS with grants matrix for the runtime content.

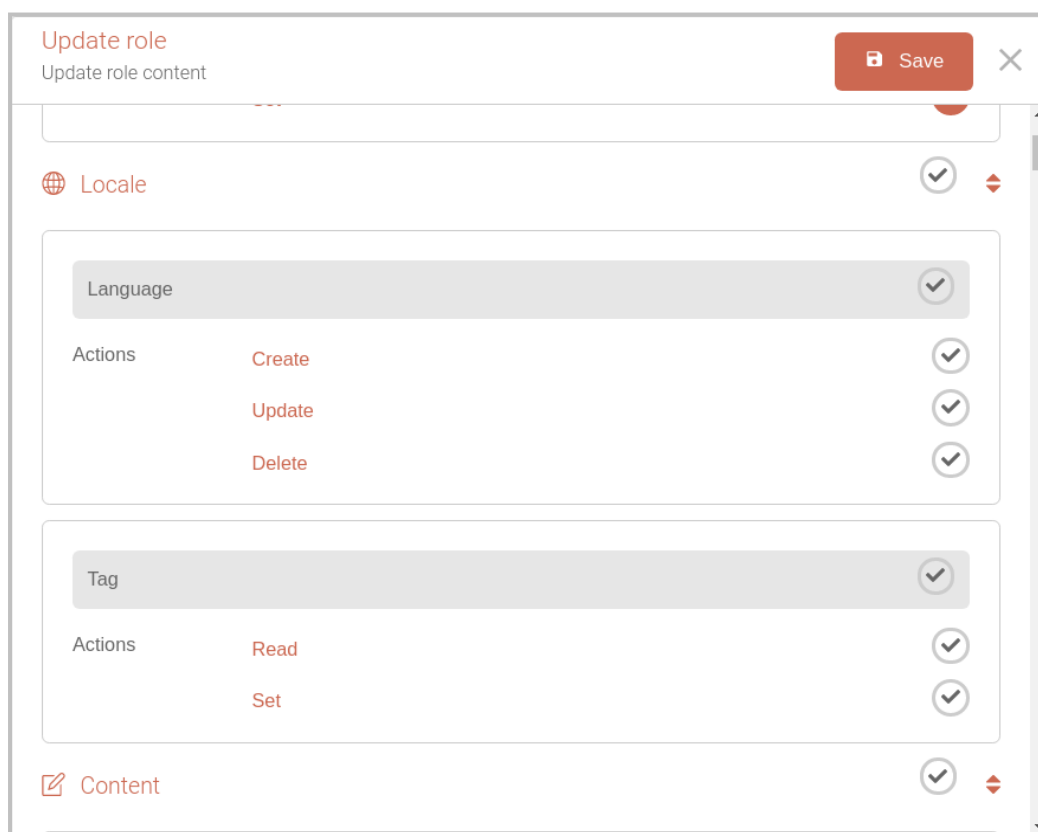


Figure 141: role grants matrix editor

The interface will automatically hide unwarranted areas, and an error will be thrown if a forbidden request was issued directly through an HTTP client.

Specific routes are publicly accessible such as language fetches; some others do require a login-state only, such as the user information update route, while the rest often require extra grant privileges.

The root role is added by default to every application instance with universal grant rights to carry any action on the system; a root user will be created on application boot as well.

RBAC is a versatile access control policy, but, as simplistic, it makes it easy to identify new groups and create hierarchies; it avoids the redundancy of ACL by assigning permissions to specific operations with organizational significance, rather than to low-level data objects [116].

Each role is identified by its key; keys must consist of alphanumeric or underscore characters only and must not be duplicated.

If a role is referenced elsewhere, its key gets locked and cannot be deleted or renamed until the lock is released.

Roles listing display grants count in addition to creation and last update dates, three conventional view variants are available (Grid, Row, and List) with the row variant as the default.

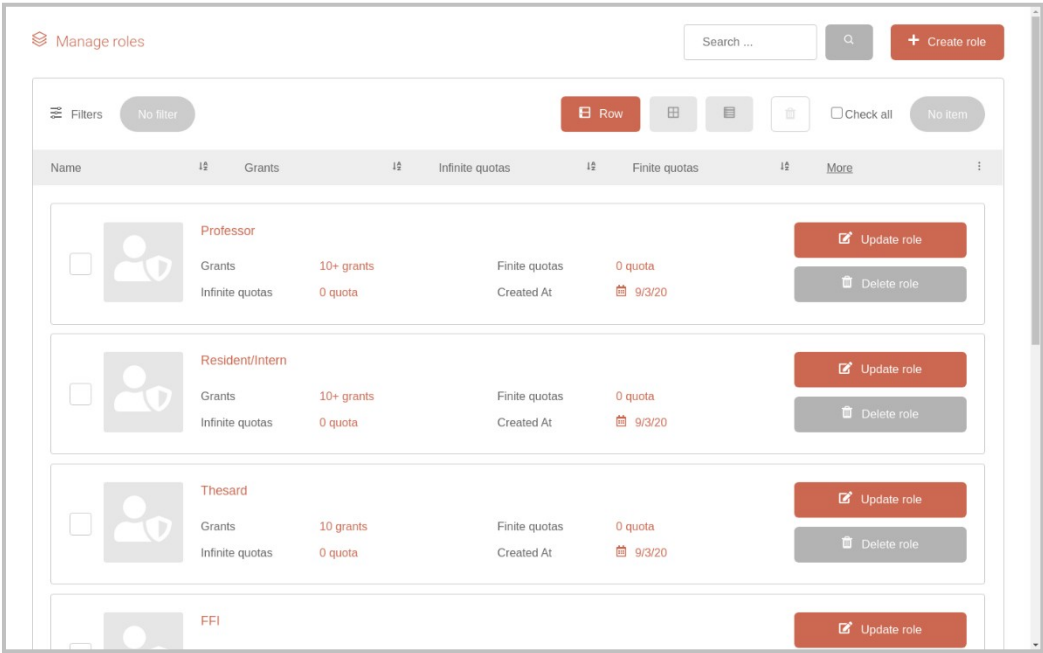


Figure 142: role entities listing

B. Attribute-based access control

Unlike role-based access control (RBAC), which employs pre-defined roles that carry a specific set of privileges associated with them and to which subjects are assigned, the key difference with ABAC is the concept of policies that express a complex Boolean rule set that can evaluate many different attributes [117,118].

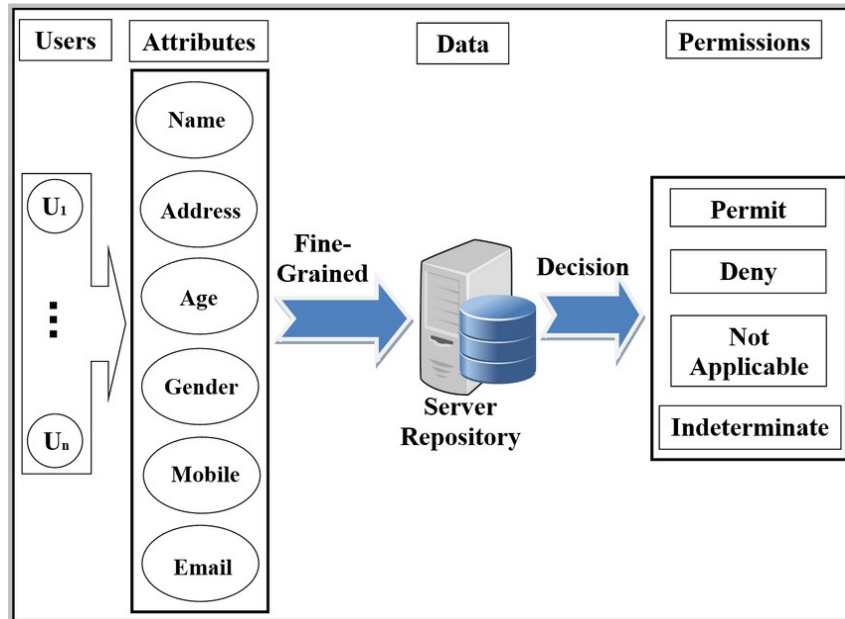


Figure 143: attribute-based access control

ABAC can be seen as [118]:

- Externalized authorization management.
- Dynamic authorization management.
- Policy-based access control.
- Fine-grained authorization.

Attribute-based access control can be asserted within expressions which are abundant throughout the entire application.

Every expression within the scheme runtime has complete access to the user's metadata and attributes and the session's metadata; Policies can be easily written in *metalambda* expressions, e.g., one can filter admissions by user's attributed department.

```

attributes: null
create: 1599164301362
▶ granted: (352) ["setting@account:set", "setting@detail:set", "locale@language:create", ...
▶ grants: (352) ["setting@account:set", "setting@detail:set", "locale@language:create", "...
id: 1
role: "role.root"
subject: "root"
update: 1599164301362

```

Figure 144: user metadata attributes

Our ABAC implementation is exceptionally flexible that CBAC, GBAC, and ERBAC policies can be implemented as well.

C. Hierarchical role-based access control

Hierarchical role-based access control H-RBAC or simply hierarchical access control is the next abstraction level of RBACs; roles are not just flat without no relation between each other but hierarchical, its hierarchy can be defined using recursive references or by comparing grant sets for superset/subset relations; Further restrictions can be applied based on hierarchy, e.g., a user can only invite users that are below him in the hierarchy level [119].

Our implementation is based on the set relations algorithm; a role is either a parent, a child or not related to another role; The root role is always a parent role.

For a role to be considered a child role, its grants set must be of a strict subset of the parent role.

Users cannot manage other users who's their roles are not a child of their role; likewise, they cannot add new roles unless they don't make for a child for them.

D. User management

User management describes the ability for administrators to manage user access to various IT resources. it is a core part of any directory service and is basic security essential for any organization [120].

Authentication, not to be confused with user management, is the act of proving an assertion, such as the identity of a computer system user. In contrast with identification, the act of indicating a person or thing's identity, authentication is the process of verifying that identity [120].

Table XL: supported account management operations

Action	Description
Register	A user can register to the application
Login	Sign in a new session
Logout	Sign out from an existing session
Update information	Update user own metadata
Change password	Change password given the old password
Recover password	Send a recovery code to email and use it to recover the password

The 'Signup' interface is a vertical form with a light gray border. At the top, the title 'Signup' is in orange. Below it is a paragraph: 'Signup to access members-only content, your membership might require additional activation procedure depending on the system'. The form contains several input fields: 'Username', 'New password', and 'Repeat password'. Below these is a 'Gender' section with a dropdown menu showing 'sexe' and an 'Edit' link. This is followed by a 'Birth Date' section with a date-time input showing '09/28/2020, 07:19 AM' and a calendar icon. Then is a 'Service' section with a dropdown menu showing 'services d'un hôpital' and an 'Edit' link. At the bottom, there is an orange button with a person icon and the text 'Register account', and a 'Go Back' link with a left arrow.

Figure 145: register new account interface

The 'Choose account' interface is a vertical form with a light gray border. The title 'Choose account' is in orange. Below it is a paragraph: 'Choose an account from the list below or logout, Click on signin to go back to home page'. There is a list of accounts, with one visible: 'root' with a user icon and 'Expires at 09:49'. To the right of the account name are icons for a key and a right arrow. At the bottom, there is an orange button with a key icon and the text 'Signin'.

Figure 146: login to an existing account

The 'Send recovery code' interface is a vertical form with a light gray border. The title 'Send recovery code' is in orange. Below it is a paragraph: 'An email/sms will be sent to your subject with instructions on how to recover your account'. There is a 'Username' input field. Below it is an orange button with a right arrow icon and the text 'Recover account'. At the bottom, there is a 'Go Back' link with a left arrow.

Figure 147: send recovery code interface

Recover account

The code you have received is valid for 24 hours, after that it will be no longer valid and you'll have to request a new one

Recover account

Send

Go Back

Figure 148: recover account interface

Old password

New password

Repeat password

Recover password

Send

Save

Code

Enter code

New password

Repeat password

Change details

Save

Username

Status

Role

root

Active

Root

Gender

sexe

Edit

Birth Date

09/28/2020, 07:22 AM

Service

services d'un hôpital

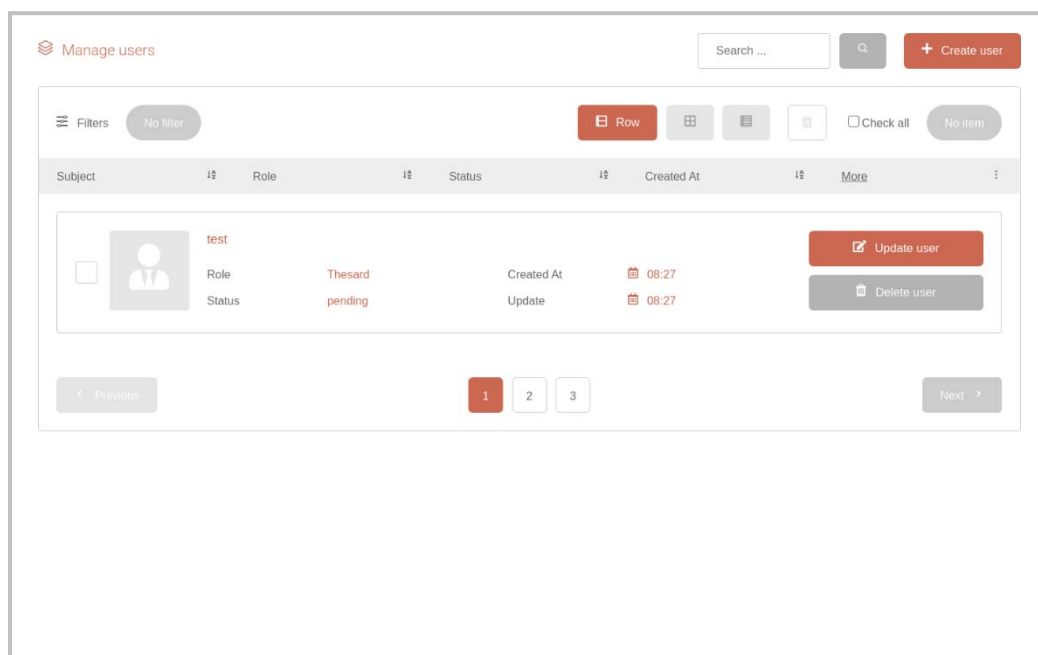
Edit

Figure 149: change account details interface

Table XLI: supported user management operations

Action	Description
Create	Create new users
Update	Update other users' metadata, excluding password
Activate	Activate other accounts
Disable	Disable other accounts
Delete	Delete other accounts
Reassign	Reassign users' roles

Each user is identified by a numeric auto-incremented id; users listing display subject, role, and status in addition to creation and last update dates, three conventional view variants are available (Grid, Row, and List) with the row variant as the default.

**Figure 150: user records listing**

E. Session management

A session is a temporary and interactive information interchange between a computer and user. A session is established at a certain point in time and then torn at some later point [121].

Session management refers to the process of securely handling multiple requests to a web-based application from a single user. Typically, a session is started when a user authenticates their

identity using a password or another authentication protocol. it involves sharing secrets with authenticated users, and as such, secure cryptographic network communications are essential to maintaining session management security [121].

Session management focuses on the entire session lifecycle from its creation destruction; Several vulnerabilities can arise if session management was flawed or poorly implemented.

Those countermeasures were taken:

- User's credentials are transmitted through a secure TLS connection with correct and valid certificates.
- Passwords are encrypted and protected from brute-force attacks against parallelization and salt key reuse using battle-tested and robust password encryption standards.
- Password or username login correctness leaks are contained.
- Tokens are transparent, digitally signed, and are short-lived tokens; JWT, JWK, JWA battle-tested standards were used.
- Tokens are refreshed on the 90th percentile of the token's age.
- Tokens are protected against tampering, against session fixation, and CSRF attacks.
- Passwords are never compromised or exposed to a request.
- Tokens are doubled checked, their metadata will be validated, and its digital signature is verified.
- Tokens are digitally signed with a strong private key.
- Tokens have double expiration windows: a max-age window, e.g., a password cannot be refreshed anymore past its max-age, and a timeout window, e.g., a token can be refreshed if the refresh request was issued within the timeout window or it will be expired otherwise.

These capabilities are supported:

- Multiple sessions are allowed.
- A maximum count of sessions cannot be created.
- Users can switch sessions at any time.
- Sessions are automatically refreshed as long as they don't pass the timeout window.
- Expired sessions will be automatically purged out.
- Request low-latency thanks to the transparent nature of JWT tokens.

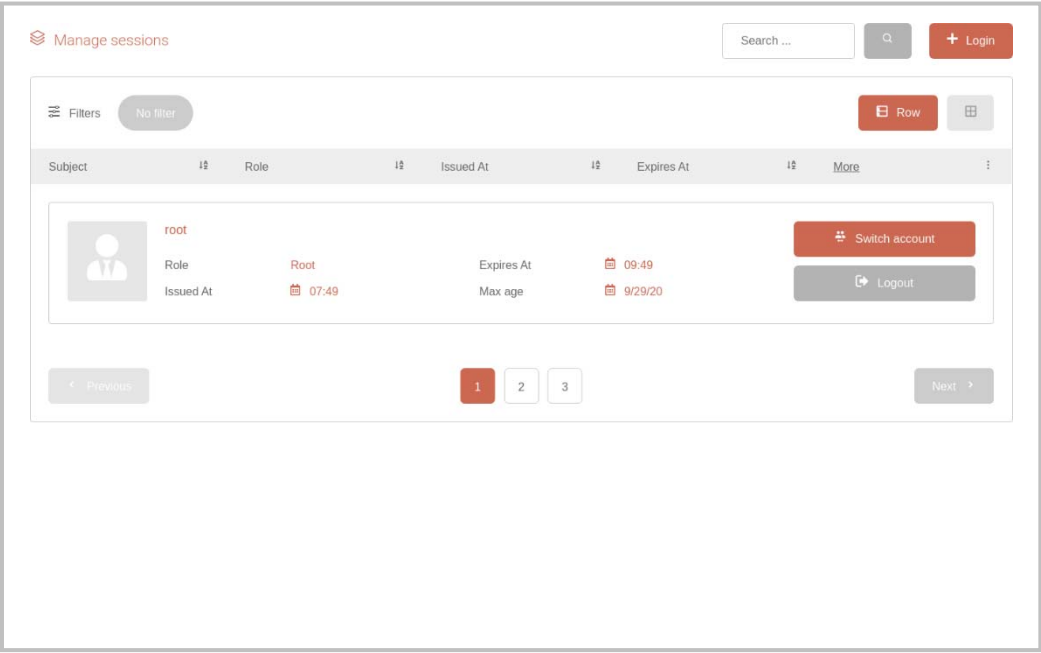


Figure 151: sessions listing

5. Ontology container

Entries with a large degree of variability can be captured via free-entry inputs, yet free entry inputs are considered to one of the worst data capture forms [122].

The ontology container comes in place to solve these issues offering a single source of truth for an auto-encodable hierarchical, semantic tree of concepts.

A. Ontology structure

In computer science, a Trie, also called a digital tree or prefix tree, is a kind of search tree—an ordered tree data structure used to store a dynamic set or associative array where the keys are usually strings. All the descendants of a node have a common prefix of the string associated with that node, and the root is associated with the empty string. Keys tend to be associated with leaves, though some inner nodes may correspond to keys of interest. Hence, keys are not necessarily associated with every node [81,123].

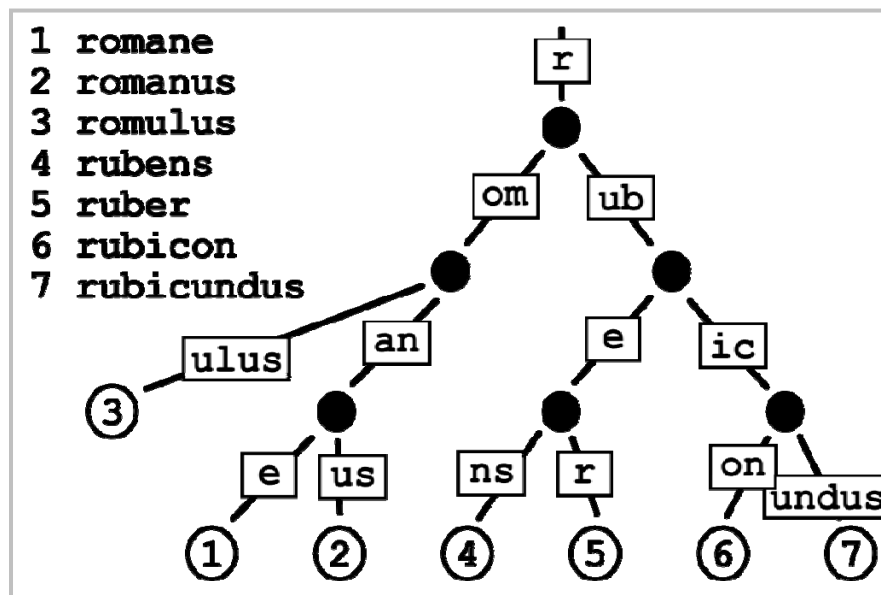


Figure 152: Trie data structure (digital prefix tree)

B. Ontology features

Keys features of the ontology container are:

- High-performance in-memory representation of the whole concepts tree.
- Realtime ontology lookup and traversal through the use of a graph data structure that allows both depth and breadth-first search.
- Concepts are automatically sorted alphabetically.
- Concepts are auto-encoded as a digital prefix tree (Trie), which allows both retrieval and traversal in $O(1)$ – constant time.
- Concept codes are semantic and hierarchic, thanks to the Trie data structure.
- Concept lookups are efficiently cached and memoized.

Table XLII: supported operations that can be carried on ontology

Operation	Description
Interpolate	Interpolate an ontology code to its ontology concept counterpart
Children	Enlist children for a given code
Parents	Enlist parents trail for a given code
Inspect	Returns deep children count and shallow children count for a given code
Ancestor	Return at the utmost the nth parent code related to a given code and a given root code, helpful in categorizing codes according to major chapters rather than distinct concepts
Search	Run a natural-language full-text search over a code child concepts
Strict prefix check	Check if a code lies within a given code sub-tree using the prefixed nature of the ontology container; a strict prefix means that a code lays strictly within the code sub-tree
Non-strict prefix check	Check if a code lies within a given code sub-tree using the prefixed nature of the ontology container; a non-strict prefix means that a code lies within the code sub-tree, the target code may be equal to the root code

C. Ontology overview

Taking data quality guarantees into consideration, we've decided to get rid of free-entry inputs and replace them with ontology concepts inputs whenever possible.

We have encoded a massive concept database with a total of 270,516+ concepts, it that covers attributes in EHR, HMS, and PMS system components.

Table XLIII: themes covered by the ontology

Theme	Description	Concepts
Anatomy	Complete human anatomy; major typologies supported: region; surface; surgical; function ... etc.	11,159
Diagnosis	Complete database of all human diagnostics provided by WHO CIM-11 2020 revision, French translation	170,819
Drug / Medication	Complete database of all drugs/treatments following the ATC (anatomical, therapeutic, chemical) hierarchy provided by WHOCC ATC	5,768
Finding	Complete ontology database of all medical findings	4,508
Function	Complete ontology database of all anatomic and biologic functions	13,494
Intervention	Complete ontology database of all medical, surgical, nursing, and paramedical interventions	18,131
Organism	Complete ontology database of all biologic organisms	18,632
Parameter	Complete ontology database of all procedure result parameters	648
Procedure	Complete ontology database of all medical procedures	5,163
Substance	Complete ontology database of major chemical substances	6,600
Symptom	Complete ontology database of all symptoms	848

Table XLIV: attributes covered by the ontology

Attribute	Description	Concepts
Units	Universal units of measure ontology	1,237
Devices	Ontology of medical devices	765
Facilities	Ontology of healthcare facilities	166
Demographics	Ontology of patient demographics	5,232
Consents	Ontology for informed consents	29
Assessments	Ontology of all medical assessments	735

Attributions to WHO CC and NLM for the IDC-11, ATC, SNOMED-CT and MeSH ontologies used in the diagnoses and drugs, anatomy, organisms, finding and function themes (UMLS license); and to NABM for the procedures lists (CC-0 license), Other ontology themes or parts were compiled from open source databases (MIT, CC-0, GPL ... etc).

6. Data warehouse system

Content captured by the universal content management system can be exploited by the universal data warehouse system; in our context, it serves as a clinical data warehouse.

A Clinical Data Repository (CDR) or Clinical Data Warehouse (CDW) is a real-time database that consolidates data from various clinical sources to present a unified view of system data. It is optimized to allow clinicians to retrieve data for a single patient rather than to identify a population of patients with common characteristics or to facilitate a specific clinical department's management. Typical data types that are often found within a CDR include clinical laboratory test results, patient demographics, pharmacy information, radiology reports and images, pathology reports, hospital admission, discharge and transfer dates, ontology codes, discharge summaries, and progress notes [45,124].

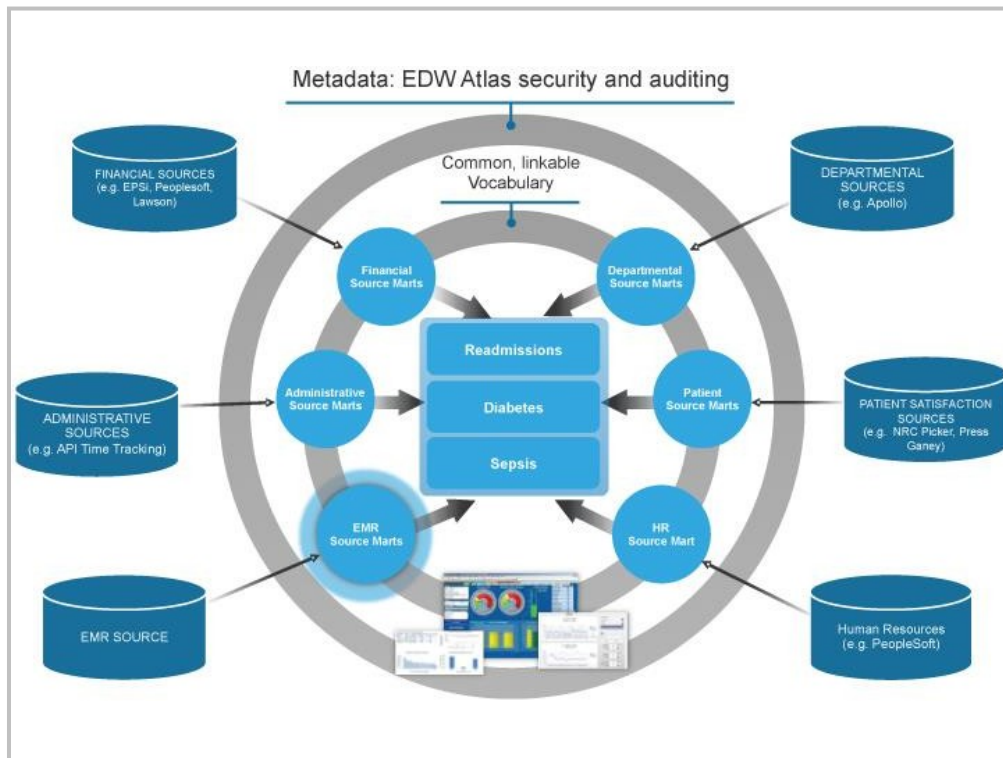


Figure 153: clinical data warehouse [124]

A. Expression container

metalambda container is qualified as an:

- Information retrieval query language.
- Data manipulation language.
- Procedural programming language.
- Declarative programming language.
- Single-threaded synchronous language capable of executing asynchronous calls in a synchronous manner.

External dependencies such as content, ontology ... etc. are ported in as interfaces:

- **Content:** you can pull scheme records and navigate relations within expressions.
- **Ontology:** you can interpolate and retrieve metadata about ontology codes.
- **Fetching:** you can download content from the internet, you can pull data out of Google Sheets ... etc.

Table XLV: supported list data transformations

Operation	Description
Transform	List entities can be remapped to different outcomes per item
Filter	Unwanted entities can be stripped out
Sort	Entities can be sorted according to an entity value increasingly or decreasingly
Limit	List length can be truncated down to a maximum count, or an offset can be skipped
Group	Entities can be grouped by similar key values, and grouped entities can be further aggregated

Supported arithmetic operations:

- Multiplication.
- Summation.
- Subdivision.
- Division.
- Modulus.
- Exponentiation.

Supported boolean operations:

- AND operation.
- OR operation.
- Boolean negation.

Supported comparison operations:

- Strict equality.
- Non-strict equality.
- Negated strict equality.
- Negated non-strict equality.
- Greater comparison.
- Lesser comparison.
- Greater or equal comparison.
- Lesser or equal comparison.

In addition to the built-in operations support, these functions are supported as well:

- **Statistics:** descriptive statistics operations are supported, such as the mean, mode, standard deviation ... etc.
- **Text:** manipulation of textual data, e.g., concatenation, splitting, join ... etc.
- **Date and duration:** interpolation of date or duration values.
- **Type:** type coercion, type checking, and fake data generation.

B. Query entity

Queries are the first layer of abstraction in the data management system; they interface directly with the content and the ontology.

Queries are ultimately *metalambda* expressions, whether supplied as expressions or over the visual editor.

Each query must return a list of key-value objects; expressions must be strongly typed down to the key-value datatype so the query runtime can infer keys datatypes.

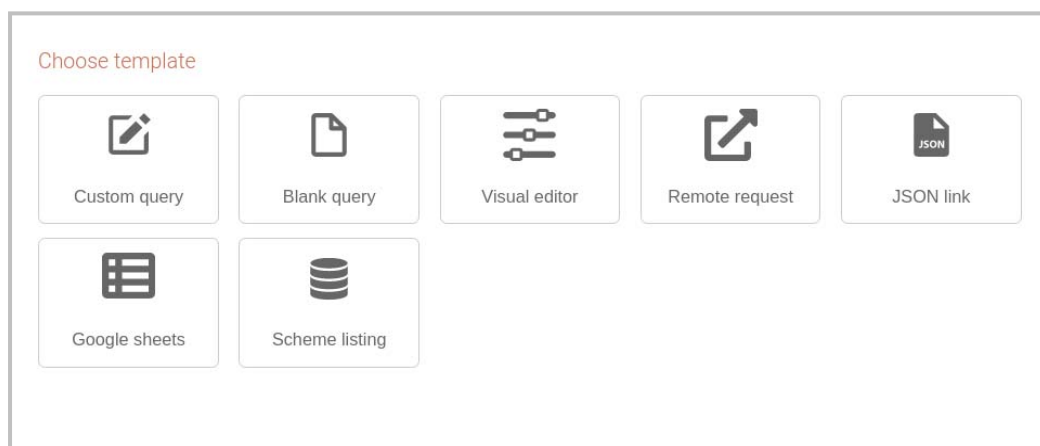


Figure 154: available predefined query templates



Figure 155: query expression editor

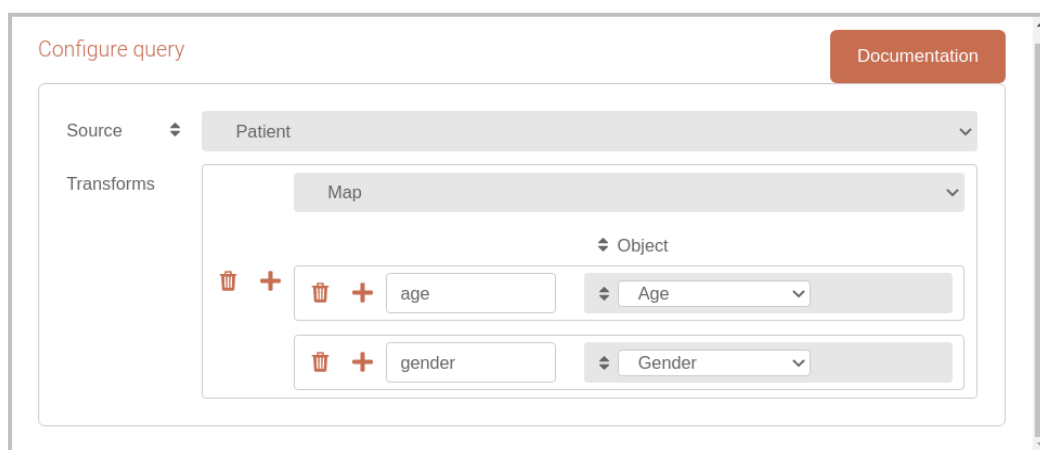


Figure 156: query visual editor

Expressions are a qualified data manipulation and information retrieval query language solution capable of executing the vast majority of modern data queries.

Queries can be verified in realtime; results are strongly typed and are visualized in a table component.

Data displayed in the table component can be navigated using pagination controls; It is possible to download results in CSV format (can be opened in any tabulation software effortlessly).

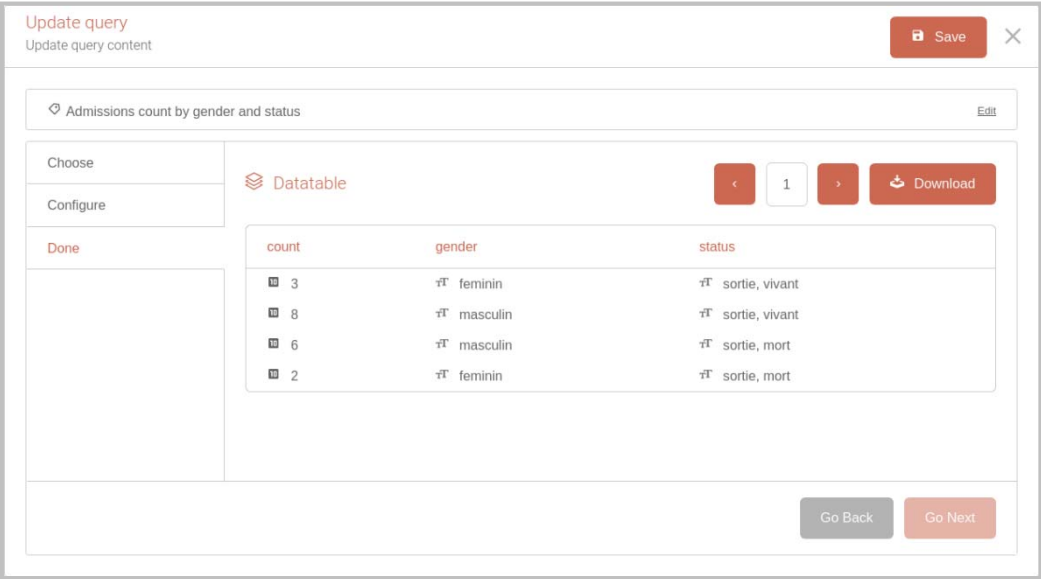


Figure 157: query results interface

Each query is identified by its key; keys must consist of alphanumeric or underscore characters only and must not be duplicated.

If a query is referenced elsewhere, its key gets locked and cannot be deleted or renamed until the lock is released.

Queries listing display query characters count in addition to creation and last update dates; three conventional view variants are available (Grid, Row, and List) with the list variant as the default.

C. Chart entity

Data visualization is the graphic representation of data. It involves producing images that communicate relationships among the represented data to viewers of the images [125].

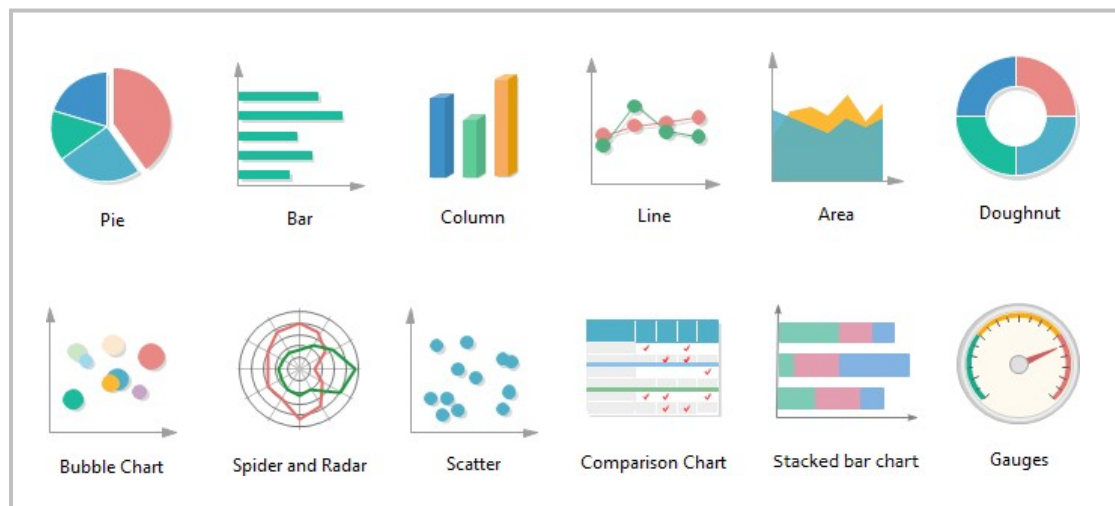


Figure 158: data visualization plots

Query results are returned as a strongly typed list of key-values; values are always scalars.

Table XLVI: supported visualization plot types

Plot	Description
Table	Display results in a tabular form, the default plot
Summary	Display a numeric vector through a descriptive statistics summarizer
Line	x-y line plot for numeric data; categorical and time-series axis are supported
Area	x-y area plot for numeric data; categorical and time-series axis are supported
Bar	x-y bar plot for numeric data; categorical and time-series axis are supported
Pie	Radial sectors or pie sector for probabilistic or prevalence vectors
Gauge	Gauge plots suitable for scalar vectors
Radar	Radar or web-like plot for categorical sets of numeric vectors
Scatter	x-y-z plots suitable for two-dimensional numeric vectors

Bubble	x-y-z plots with z-axis emphasis the scale of dots, suitable for three-dimensional numeric vectors
Heat-map	Suitable to display frequency vectors
Pyramid	Suitable for pyramid numeric vectors
Funnel	Suitable to numeric vectors of a funnel
Chord	Display interchange scales between components for a pivot NxN matrix

Each plot variant can be configured, its appearance customized; the combinations are endless; each configuration set requires a data shape, so it is mandatory to check configurations validity by toggling into the visualization tab.

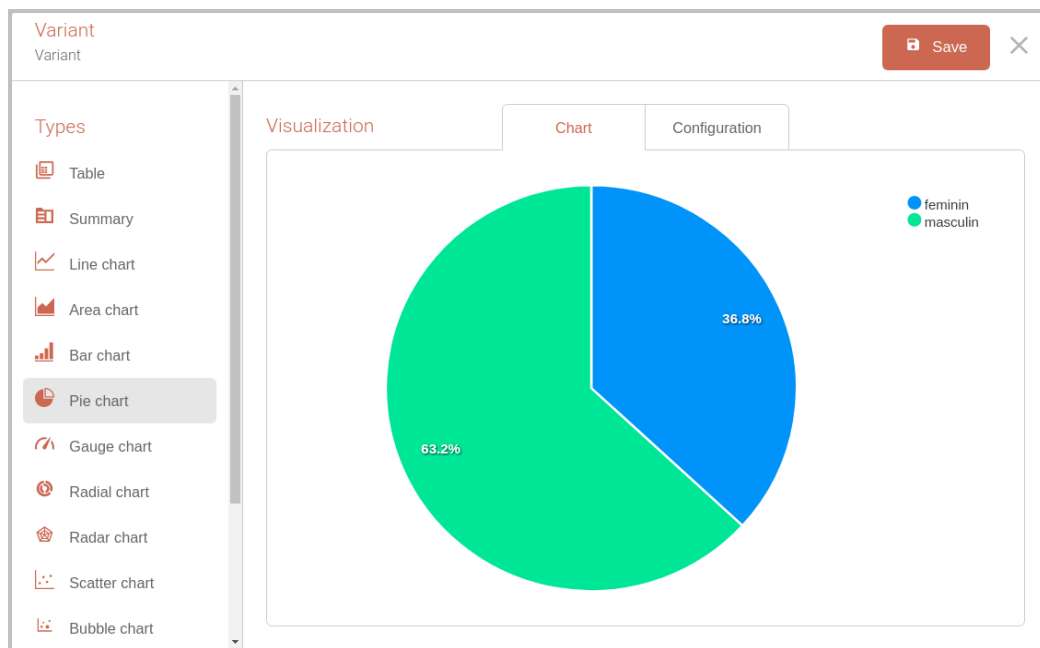


Figure 159: chart variant editor

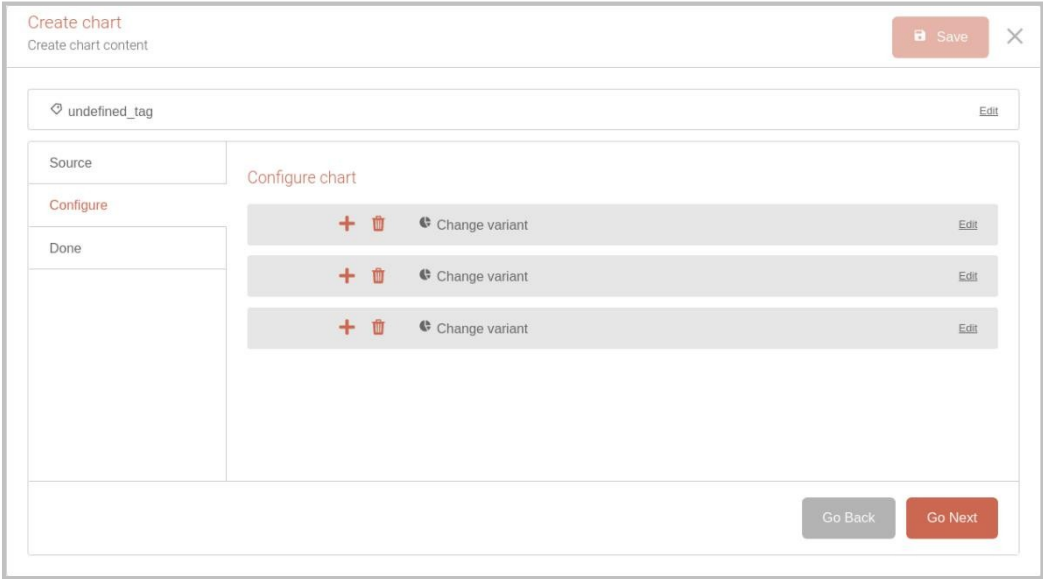


Figure 160: chart entity editor

Each chart is identified by its key; keys must consist of alphanumeric or underscore characters only and must not be duplicated.

If a chart is referenced elsewhere, its key gets locked and cannot be deleted or renamed until the lock is released.

Charts listing display query key, transformation status, and variants count in addition to creation and last update dates, three conventional view variants are available (Grid, Row, and List) with row variant as the default.

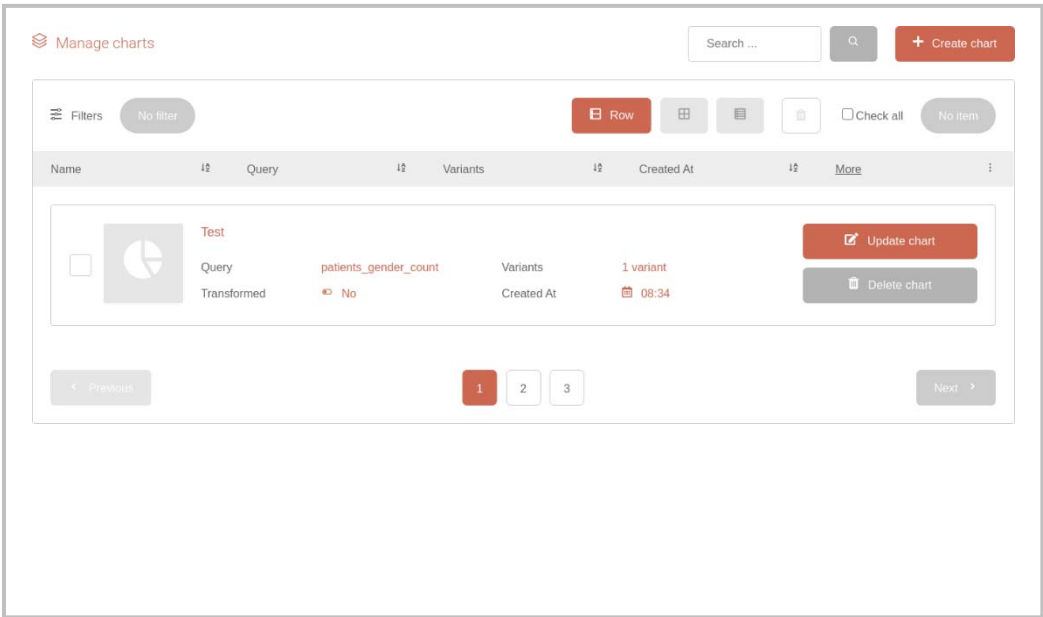


Figure 161: chart entities listing

D. Board entity

Boards can be helpful to group contextually similar charts into a single board; they are useful in:

- Aggregating similar data visualization.
- Maintain focus on specific charts.
- Limit access to individual boards.

Boards are edited through a re-sizable drag-and-drop grid component; columns and rows sized can be customized; an active chart plot variant can be selected; charts can be auto-packed as well.

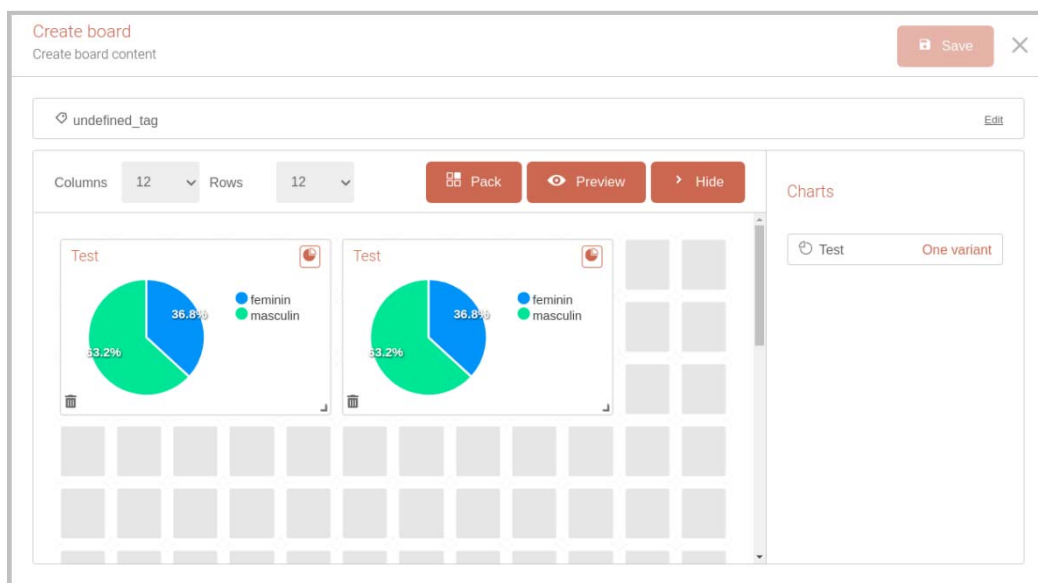


Figure 162: board entity editor

The same component is used to visualize boards in run-time but without the resizing and drag-and-drop features.

Each board is identified by its key; keys must consist of alphanumeric or underscore characters only and not be duplicated.

If a board is referenced elsewhere, its key gets locked and cannot be deleted or renamed until the lock is released.

Boards listing show rows, columns, and charts count in addition to creation and last update dates, three conventional view variants are available (Grid, Row, and List) with row variant as the default.

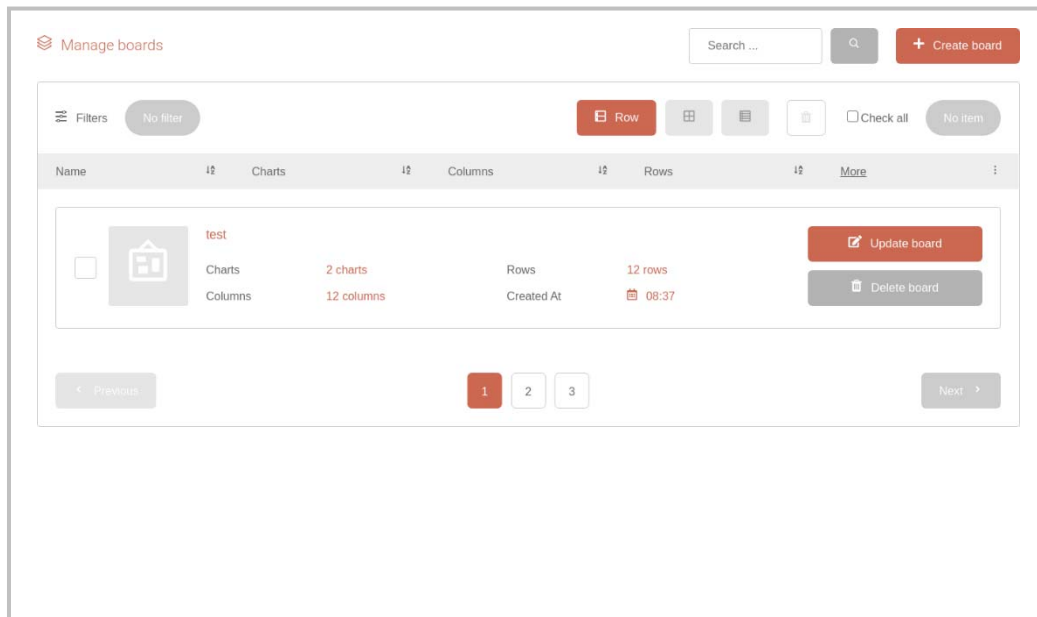


Figure 163: board entities listing

7. Localization and internationalization system

A. Tags and templates

Internationalization system organization follows the same separation between data definition and data entry, more precisely between container(system) and content.

There are two translations providers:

- **System provider:** centralize translation of the backend and the frontend; errors, messages, and labels are all automatically extracted from source code, linted, checked, validated, and compiled into a highly compressed JSON format; they are conventionally named as templates.
- **Content provider:** every property, entity, and attribute defined or entered in the content definition system is automatically extracted and tracked; they are conventionally named as tags.

The system provider supports an unlimited number of locales; support RTL and LTR languages; new languages can be added on the fly with zero downtime.

Tags are raw text values; their data structure is a key-value map data structure; each tag key has its locale-specific translation counterpart.

Templates make use of *metatext* dependency; they are dynamic expressions that can be computed on the runtime can they can access external variables.

Table XLVII: *metatext* built-in locale-specific features

Feature	Description
Inflections rules	Singularization, pluralization, or any inflection range
Date formatting	Locale-specific date formatting conventions
Duration formatting	Locale-specific duration formatting conventions
Numeric formatting	Flexible numeric values formatting
Unit formatting and conversion	Unit conversion and formatting for many scales
Text transformation	Title, sentence, pascal or camel case conversion and lowercase or uppercase conversion
Conjunction	Conjunction or dis-junction rules

The system provider already comes pre-translated in French and English locales.

B. Tag entity

Tags are the content internationalization provider; their data structure is a key-value map data structure; each tag key has its locale-specific translation counterpart.

Each tag key has a prefix and a suffix concatenated with a dot; keys are prefixed to prevent collisions between tags from different contexts; a key prefix has contextual meaning; meanwhile, a key suffix stands for the key name.

Each tag is associated with locale-specific translations, with the empty string as the default translation.

Update tag

Update tag content

Save

✕

root

Edit

en

Root

fr

Système

Figure 164: tag entity editor

The system automatically keeps track of all tags in use; unused tags can be purged easily.

Tags listing display key, prefix, suffix, and locales count in addition to creation and last update dates, three conventional view variants are available (Grid, Row, and List) with the row variant as the default.

Manage tags

Search ...

Clean unused

Filters

No filter

Row

Grid

List

Check all

No item

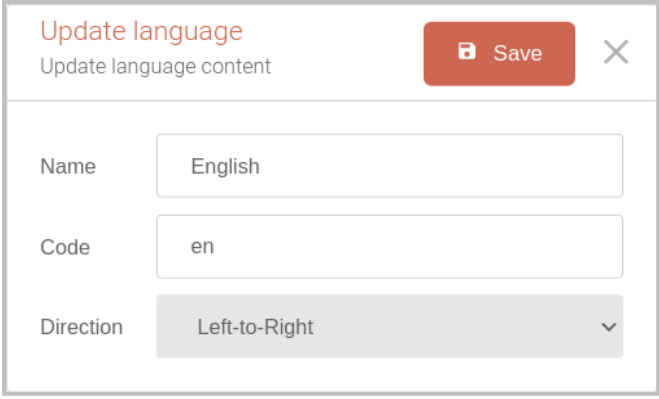
Name	Prefix	Locales	Suffix	More
☐ role.root	Prefix Locales	role 2 locales	Suffix Created At root 9/3/20	Update tag
☐ setting.title	Prefix Locales	setting 2 locales	Suffix Created At title 9/3/20	Update tag
☐ setting.name	Prefix Locales	setting 2 locales	Suffix Created At name 9/3/20	Update tag
☐ setting.paragra...	Prefix Locales	setting 2 locales	Suffix Created At paragraph 9/3/20	Update tag
☐ setting.header	Prefix Locales	setting 2 locales	Suffix Created At header 9/3/20	Update tag
☐ setting.footer	Prefix	setting	Suffix footer	Update tag

Figure 165: tag entities listing

C. Language entity

Languages are the cornerstone of the internationalization system; at least one language must exist or remain in the system.

Language metadata are language name, code, and direction; language code implements the IETF language tag standard, which is extensible to region, dialect, and private designations. It references ISO 639, ISO 3166, and ISO 15924.



The image shows a web-based form titled "Update language" with the subtitle "Update language content". The form has three input fields: "Name" containing "English", "Code" containing "en", and "Direction" with a dropdown menu currently showing "Left-to-Right". A red "Save" button is located in the top right corner of the form area.

Figure 166: language entity editor

Languages listing display name, code, locale, and direction in addition to creation and last update dates, three conventional view variants are available (Grid, Row, and List) with row variant as the default.

IV. Software testing

1. Unit testing

The software was built following a test-driven development approach. Mocha and Istanbul are two major development dependencies. They serve as a testing framework and as a test runner, respectively. Targeted unit tests thoroughly tested every piece of functionality in the system; testing approaches include fuzzy testing, static testing, fault injection ... etc.

Code quality was a significant concern during development. We chose Typescript language as the primary programming language due to its type-checking capabilities; Code was linted by TSLint from code smells and style inconsistencies.

External dependencies were selected by the code coverage metric; only software with code coverage above 95% and that is battle-tested are retained.

2. Integration testing

The system components were thoroughly tested via integration tests; testing techniques include mocking, stubbing, and snapshot testing.

3. Acceptance testing

A. Objectives

Our acceptance testing specifications are mainly based on this dissertation objectives:

- Ensure that all HMS and EHR components work properly
- Ensure that the EHR system captures all data relevant to the medical record
- Ensure that all EHR and HMS captured data are exploitable
- Ensure the ability to create real-time data queries (continuous clinical research pipelines)

B. Materials and methods

To assert acceptance testing objectives:

- We've submitted 19 real patient medical records hospitalized in the newborns department for the period between 01/06/2020 and 31/07/2020.
- Submitted data include patient demographics, history notes, examination notes, prescriptions sheets, requested procedures, interventions, diagnoses, and progress notes.
- We've exploited submitted data from the clinical data warehouse and conducted a medium scale retrospective epidemiological study; all records were included in the study.
- Execute and create data queries exploring basic counts, basic correlation, advanced counts, and advanced correlation for clinical, para-clinical, diagnosis, prognosis, and therapeutic parameters.

C. Results

- Patients gender count

12 out of 19 patients were male, whereas only 7 of them were female.

Table XLVIII: patients gender count

Gender	Count
Male	12
Female	7

```
bulk("scheme.patient")
  -> map((item) => {
    gender: item.value.item.gender ->
    interpolate(),
  })
  -> <[{
    gender: string,
    count: integer,
  }*>@group(["gender"], (item) => {
    gender: item[0].gender,
    count: item -> count(),
  })
```

Figure 167: patient gender count query

Query explanation: all records were pulled from “scheme.patient” datatable, each record was mapped for the gender value (originally an ontology code that gets interpolated), then all records were grouped by gender value, each group was counted.

- **Admissions count by gender and status**

6 male patients were deceased in contrast to only 2 deceased female patients; 8 male patients were discharged alive in contrast to 3 discharged female patients.

Table XLIX: admissions count by gender and status

Gender	Status	Count
Male	Alive	8
Female	Alive	3
Male	Deceased	6
Female	Deceased	2

```
bulk("scheme.admission")
  -> map((item) => [item, item ->
edge("patient:0:out")])
  -> map((item) => {
    gender: item[0].value.item.admission.gender
  -> interpolate(),
    status: item[0].value.item.admission.status
  -> interpolate(),
  })
  -> <[ {
    gender: string,
    status: string,
    count: integer,
  }*>@group(["gender", "status"], (item) => {
    gender: item[0].gender,
    status: item[0].status,
    count: item -> count(),
  })
```

Figure 168: admissions count by gender and status

Query explanation: all records were pulled from “scheme.admission ” datatable, each record was mapped for the gender value (originally an ontology code that gets interpolated) and status value, then all records were grouped by gender and status values, each group was counted.

- **Admissions count by reason**

6 male patients were deceased in contrast to only 2 deceased female patients; 8 male patients were discharged in contrast to 3 discharged female patients.

Table L: admissions count by reason

Reason	Count
Neonatal hyperbilirubinemia due to rhesus incompatibility hemolysis	3
Structural abnormality in the development of the heart or large vessels	2
Obstetric asphyxia	2
Massive pulmonary hemorrhage occurring during the perinatal period	1
Fetal or newborn sepsis	1
Congenital myocardial insufficiency	1
Severe obstetric asphyxia	1
Newborn urinary tract infection	1
Fetus or newborn with prolapse	1
Fetus or newborn with maternal infectious diseases	1
Primary atelectasis of the newborn	1
Kernicterus due to isoimmunization	1
Extreme prematurity of newborn, gestational age 23 weeks completed	1
Neonatal meningitis	1
Respiratory failure in newborns	1

```

bulk("scheme.admission")
  -> map((item) => {
    reason: item.value.item.admission.reason ->
    interpolate(),
  })
  -> group(["reason"], (item) => {
    reason: item[0].reason,
    count: item -> count(),
  })
  -> <[ {
    reason: string,
    count: integer,
  }*>@sort("desc", (item) =>
    <integer>item.count)

```

Figure 169: admissions count by reason query

Query explanation: all records were pulled from “scheme.admission” datatable, each record was mapped for the admission reason value (originally an ontology code that gets interpolated) then all records were grouped by reason value, each group was counted.

- **Admissions count by reason category**

The respiratory diseases category was the most frequent admission reasons category, followed by infections and hematologic affections, then malformations.

Table LI: admissions count by reason category

Category	Count
Respiratory disorders specific to the perinatal or neonatal period	6
Infections of the fetus or newborn	4
Bleeding or hematological disorders of the fetus or newborn	4
Structural–developmental abnormalities primarily affecting a bodily system	3
Fetuses or newborns affected by maternal factors or by complications of pregnancy, labor, or delivery	1
Newborn disorders related to the length of gestation or fetal growth	1

```
bulk("scheme.admission")
  -> map((item) => {
    reason: item.value.item.admission.reason ->
    ancestor("6", 3) -> interpolate(),
  })
  -> group(["reason"], (item) => {
    reason: item[0].reason,
    count: item -> count(),
  })
  -> <[{
    reason: string,
    count: integer,
  }*>@sort("desc", (item) =>
    <integer>item.count)
```

Figure 170: admissions count by reason category query

Query explanation: all records were pulled from “scheme.admission” datatable, each record was mapped for the admission reason category (max depth of 3), then all records were grouped by reason category, each group was counted, results were sorted by count decreasingly.

- **Apgar score mean by admission status**

A lower Apgar score was correlated with high mortality rates, yet standard deviation intervals make the correlation inconclusive.

Table LII: Apgar score mean by admission status

Status	Apgar score (mean)
Alive	8.63 $\pm$ 2.50
Deceased	6.37 $\pm$ 3.37

```
bulk("scheme.admission")
  -> map((item) => {
    status: item.value.item.admission.status ->
interpolate(),
    apgar:
item.value.item.history.delivery_history.apgar,
  })
  -> <[ {
    status: string,
    mean: float(digit:max=2),
    stdv: float(digit:max=2),
  }*>@group(["status"], (item) => {
    status: item[0].status,
    mean: item -> <[number*>@map((entry) =>
entry.apgar) -> mean(),
    stdv: item -> <[number*>@map((entry) =>
entry.apgar) -> stdv(),
  })
```

Figure 171: Apgar score mean by admission status query

Query explanation: all records were pulled from “scheme.admission” datatable, each record was mapped for admission status value (originally an ontology code that gets interpolated) and corresponding Apgar score value, then all records were grouped by status value, mean, and standard deviation aggregates were calculated for each group.

- **Farr score mean by admission status**

A lower Farr score was correlated with high mortality rates, yet standard deviation intervals make the correlation inconclusive.

Table LIII: Farr score mean by admission status

Status	Farr score (mean)
Alive	23.18 $\pm$ 4.53
Deceased	17.25 $\pm$ 6.62

```

bulk("scheme.admission")
  -> map((item) => {
    status: item.value.item.admission.status ->
    interpolate(),
    farr:
    item.value.item.examination.newborn_exam.farr_score
  })
  -> <[ {
    status: string,
    mean: float(digit:max=2),
    stdv: float(digit:max=2),
  }*>@group(["status"], (item) => {
    status: item[0].status,
    mean: item -> <[number*>@map((entry) =>
    entry.farr) -> mean(),
    stdv: item -> <[number*>@map((entry) =>
    entry.farr) -> stdv(),
  })

```

Figure 172: Farr score mean by admission status query

Query explanation: all records were pulled from “scheme.admission” datatable, each record was mapped for admission status value (originally an ontology code that gets interpolated) and corresponding Farr score value, then all records were grouped by status value, mean, and standard deviation aggregates were calculated for each group.

- **Unretained admission diagnosis category count**

Hematologic and infections are the most invoked but not retained diagnosis categories

Table LIV: unretained admission diagnosis category count

Category	Count
Bleeding or hematological disorders of the fetus or newborn	13
Infections of the fetus or newborn	8
Transient endocrine or metabolic disorders specific to the fetus or newborn	8
Structural–developmental abnormalities primarily affecting a bodily system	8
Neurological disorders specific to the perinatal or neonatal period	4
Respiratory disorders specific to the perinatal or neonatal period	3

```

bulk("scheme.admission")
  -> map((item) => {
    diagnosis: item.value.item.diagnosis,
  })
  -> reduce((prev, item) => prev ->
merge(item.diagnosis), [])
  -> map((item) => {
    disease: item.disease -> ancestor("6", 3) -
> interpolate(),
    retained: item.retained,
  })
  -> filter((item) => !item.retained)
  -> group(["disease"], (item) => {
    diagnosis: item[0].disease,
    count: item -> count(),
  })
  -> <[ {
    diagnosis: string,
    count: integer,
  }*]>@sort("desc", (item) =>
<integer>item.count)

```

Figure 173: unretained admission diagnosis category count query

Query explanation: all records were pulled from “scheme.admission” datatable, diagnosis list is concatenated for all records, then each record was mapped for disease category (max depth of 3) along with its retention status, then records that were retained got stripped out, then all records were grouped by disease value, each group was counted results were sorted by count decreasingly.

- **Maternal age mean by status**

It appears that newborns from older mothers are associated with high mortality rates, but standard deviation intervals are inconclusive.

Table LV: maternal age mean by status

Status	Maternal age (mean)
Alive	25 years 6 months $\pm$ 8 years 10 months
Deceased	30 years 1 month $\pm$ 6 years 1 month

```
bulk("scheme.admission")
-> map((item) => {
  status: item.value.item.admission.status ->
  interpolate(),
  age:
  item.value.item.history.gynecologic_history.age,
})
-> <[{
  status: string,
  mean: duration,
  stdv: duration,
}]*>@group(["status"], (item) => {
  status: item[0].status,
  mean: item -> <[number*]>@map((entry) =>
  entry.age) -> <integer>@mean(),
  stdv: item -> <[number*]>@map((entry) =>
  entry.age) -> <integer>@stdv(),
})
```

Figure 174: maternal age mean by status query

Query explanation: all records were pulled from “scheme.admission” datatable, each record was mapped for admission status value (originally an ontology code that gets interpolated) and corresponding maternal age value, then all records were grouped by status value, mean, and standard deviation aggregates were calculated for each group.

- **Procedure category count**

Biochemistry procedures are the most request procedures followed by hematology, then microbiology procedures.

Table LVI: procedure category count

Category	Count
Biochemistry	9
Hematology	8
Microbiology	6

```
bulk("scheme.admission")
  -> map((item) => {
    procedure: item.value.item.procedure
    -> map((procedure) =>
single("scheme.procedure", <integer>@procedure)),
  })
  -> reduce((prev, item) => prev ->
merge(item.procedure), [])
  -> map((item) => {
    procedure: item.value.item.procedure.type -
> interpolate(),
  })
  -> group(["procedure"], (item) => {
    procedure: item[0].procedure,
    count: item -> count(),
  })
  -> <[ {
    procedure: string,
    count: integer,
  }*>@sort("desc", (item) =>
<integer>item.count)
```

Figure 175: procedure category count query

Query explanation: all records were pulled from “scheme.admission” datatable, procedures list are concatenated for all records, relations with procedures datatable were traversed, then each record was mapped for procedure category (interpolated), then all records were grouped by procedure category value, each group was counted, results were sorted by count decreasingly.

- **Procedure requests count**

Urea-creatinine was the most request procedure, followed by blood hemogram and blood ionogram.

Table LVII: procedure requests count

Category	Count
Urea and creatinine (blood)	9
Blood count including platelets	8
Ionogram (blood)	8
C reactive protein (CRP) (blood)	7
Bicarbonates or total CO2	6
Calcium (blood)	6
Transaminase glut. oxalic. (TGO, ASAT, AST) (another biological medium)	5
Glucose (blood sugar) (blood)	5
Alanine aminotransferase (ALAT, TGP)	4
Glucose (Glucorachia) (CSF)	3
Microbiological examination of the puncture fluid (CSF, joint, peritoneum, etc.)	3
Total protein (proteinorachia) (CSF)	3
Qualitative microbiological blood culture examination	2
CRP (C reactive protein) (assay) (other fluids)	2
Bilirubin assay	1
Urine microbiome examination (CBUE)	1

```

bulk("scheme.admission")
  -> map((item) => {
    procedure: item.value.item.procedure
    -> map((procedure) =>
single("scheme.procedure", <integer>@procedure)),
  })
  -> reduce((prev, item) => prev ->
merge(item.procedure, [])
  -> reduce((prev, item) => prev ->
merge(item.value.item.procedure.procedures, []))
  -> map((item) => {
    procedure: item,
  })
  -> group(["procedure"], (item) => {
    procedure: item[0].procedure ->
interpolate(),
    count: item -> count(),
  })
  -> <[ {
    procedure: string,
    count: integer,
  }*>@sort("desc", (item) =>
<integer>item.count)

```

Figure 176: procedure requests count query

Query explanation: all records were pulled from “scheme.admission” datatable, procedures list are concatenated for all records, relations with procedures datatable were traversed, then each record was mapped for procedure list (interpolated), then all procedure lists were concatenated altogether, then all records were grouped by procedure value, each group was counted, results were sorted by count decreasingly.

- **Mean of admission anthropometrics by status**

Low head perimeter, low birth height, and low birth weight are associated with high mortality.

Table LVIII: mean of admission anthropometrics by status

Status	Height (mean)	Weight (mean)	Head Perimeter (mean)
Alive	48.18	3.18	34.81
Deceased	41.25	2.30	30.75

```
bulk("scheme.admission")
-> map((item) => {
  status: item.value.item.admission.status ->
  interpolate(),
  weight:
  item.value.item.examination.newborn_exam.weight_kg,
  height:
  item.value.item.examination.newborn_exam.height_cm,
  head_perimeter:
  item.value.item.examination.newborn_exam.head_perimeter,
})
-> <[{
  status: string,
  weight: float(digit:max=2),
  height: float(digit:max=2),
  head_perimeter: float(digit:max=2),
}*]>@group(["status"], (item) => {
  status: item[0].status,
  weight: item -> <[number*]>@map((entry) =>
  entry.weight) -> mean(),
  height: item -> <[number*]>@map((entry) =>
  entry.height) -> mean(),
  head_perimeter: item ->
  <[number*]>@map((entry) => entry.head_perimeter) ->
  mean(),
})
```

Figure 177: mean of admission anthropometrics by status query

Query explanation: all records were pulled from “scheme.admission” datatable, each record was mapped for admission status value (originally an ontology code that gets interpolated) along with anthropometrics; then all records were grouped by status value, mean aggregates were calculated for each group.

- **Correlation between the instructor occupation and admission status**

No correlation between childbirth instructor occupation and newborn mortality was found.

Table LIX: correlation between the instructor occupation and admission status

Instructor	Status	Count
Gynecologist	Alive	7
Midwife	Alive	4
Gynecologist	Deceased	5
Midwife	Deceased	3

```
bulk("scheme.admission")
  -> map((item) => {
    status: item.value.item.admission.status ->
    interpolate(),
    instructor:
    item.value.item.history.delivery_history.instructor
    -> interpolate(),
  })
  -> <[{
    status: string,
    instructor: string,
    count: integer,
  }*>@group(["status", "instructor"], (item) =>
  {
    status: item[0].status,
    instructor: item[0].instructor,
    count: item -> count(),
  })
```

Figure 178: correlation between the instructor occupation and admission status query

Query explanation: all records were pulled from “scheme.admission” datatable, each record was mapped for admission status value (originally an ontology code that gets interpolated) along with instructor occupation value, then all records were grouped by status and instructor values, each group was counted.

- **Correlation between delivery place and admission status**

Home childbirth seems to be correlated directly with newborn mortality; it appears that childbirths of deceased newborns take place either at a medium to high facility level or at home but never on a level 1 facility.

Table LX: correlation between delivery place and admission status

Delivery place	Status	Count
Level 1	Alive	1
Level 2	Alive	5
Level 3	Alive	5
Home	Deceased	1
Level 2	Deceased	5
Level 3	Deceased	2

```

bulk("scheme.admission")
  -> map((item) => {
    status: item.value.item.admission.status ->
    interpolate(),
    delivery_place:
    item.value.item.history.delivery_history.delivery_p
    lace -> interpolate(),
  })
  -> <[{
    status: string,
    delivery_place: string,
    count: integer,
  }*>@group(["status", "delivery_place"], (item)
=> {
  status: item[0].status,
  delivery_place: item[0].delivery_place,
  count: item -> count(),
})

```

Figure 179: correlation between delivery place and admission status query

Query explanation: all records were pulled from “scheme.admission” datatable, each record was mapped for admission status value (originally an ontology code that gets interpolated) along with delivery place value, then all records were grouped by status and delivery place values, each group was counted.

- **Correlation between delivery route and admission status**

C section delivery is associated with high mortality rates compared to vaginal delivery.

Table LXI: correlation between delivery route and admission status

Delivery route	Status	Count
Vaginal	Alive	8
C section	Alive	3
Vaginal	Deceased	5
C section	Deceased	3

```

bulk("scheme.admission")
  -> map((item) => {
    status: item.value.item.admission.status ->
interpolate(),
    delivery_route:
item.value.item.history.delivery_history.delivery_r
oute -> interpolate(),
  })
  -> <[ {
    status: string,
    delivery_route: string,
    count: integer,
  }*]>@group(["status", "delivery_route"], (item)
=> {
  status: item[0].status,
  delivery_route: item[0].delivery_route,
  count: item -> count(),
})

```

Figure 180: correlation between delivery route and admission status query

Query explanation: all records were pulled from “scheme.admission” datatable, each record was mapped for admission status value (originally an ontology code that gets interpolated) along with delivery route value, then all records were grouped by status and delivery route values, each group was counted.

- **Correlation between delivery durations with admission status**

Low labor durations and high rupture of membranes durations are correlated with high mortality rates.

Table LXII: correlation between delivery durations with admission status

Status	Labor duration (mean)	Rupture of membranes duration (mean)
Alive	5 hours 43 minutes	3 hours 38 minutes
Deceased	4 hours 37 minutes	3 hours 52 minutes

```

bulk("scheme.admission")
  -> map((item) => {
    status: item.value.item.admission.status ->
    interpolate(),
    labour_duration:
    item.value.item.history.delivery_history.labour_dur
    ation,
    rupture_of_membranes:
    item.value.item.history.delivery_history.rupture_of
    _membranes,
  })
  -> <[ {
    status: string,
    labour_duration: duration,
    rupture_of_membranes: duration,
  }*>@group(["status"], (item) => {
    status: item[0].status,
    labour_duration: item ->
    <[number*>@map((entry) => entry.labour_duration) -
    > <integer>@mean(),
    rupture_of_membranes: item ->
    <[number*>@map((entry) =>
    entry.rupture_of_membranes) -> <integer>@mean(),
  })

```

Figure 181: correlation between delivery durations with admission status query

Query explanation: all records were pulled from “scheme.admission” datatable, each record was mapped for admission status value (originally an ontology code that gets interpolated) along with labor duration and rupture of membranes duration, then all records were grouped by status value, mean aggregates were calculated for the two parameters in each group.

- **Correlation between pregnancy follow-up status and admission status**

No correlation between pregnancy follow-up status and mortality rates was found.

Table LXIII: correlation between pregnancy follow-up status and admission status

Followed-up pregnancy	Status	Count
Yes	Alive	6
No	Alive	5
Yes	Deceased	3
No	Deceased	5

```

bulk("scheme.admission")
  -> map((item) => {
    status: item.value.item.admission.status ->
    interpolate(),
    followed_pregnancy:
    item.value.item.history.obstetric_history.followed_
    pregnancy,
  })
  -> <[ {
    status: string,
    followed_pregnancy: string,
    count: integer,
  }*>@group(["status", "followed_pregnancy"],
(item) => {
  status: item[0].status,
  followed_pregnancy:
  item[0].followed_pregnancy,
  count: item -> count(),
})

```

Figure 182: correlation between pregnancy follow-up status and admission status query

Query explanation: all records were pulled from “scheme.admission” datatable, each record was mapped for admission status value (originally an ontology code that gets interpolated) along with follow-up value, then all records were grouped by status and follow-up values, each group was counted.

- **Correlation between consanguinity status and admission status**

Parental consanguinity is correlated with high mortality rates.

Table LXIV: correlation between consanguinity status and admission status

Consanguinity	Status	Count
Yes	Alive	1
No	Alive	10
Yes	Deceased	3
No	Deceased	5

```

bulk("scheme.admission")
  -> map((item) => {
    status: item.value.item.admission.status ->
    interpolate(),
    consanguinity:
    item.value.item.history.obstetric_history.consanguini-
    nity,
  })
  -> <[ {
    status: string,
    consanguinity: string,
    count: integer,
  }*]>@group(["status", "consanguinity"], (item)
=> {
  status: item[0].status,
  consanguinity: item[0].consanguinity,
  count: item -> count(),
})

```

Figure 183: correlation between consanguinity status and admission status query

Query explanation: all records were pulled from “scheme.admission” datatable, each record was mapped for admission status value (originally an ontology code that gets interpolated) along with consanguinity value, then all records were grouped by status and consanguinity values, each group was counted.

- **Correlation between consanguinity status and admission diagnosis**

Consanguinity is correlated with a higher incidence of respiratory diseases and malformations.

Table LXV: correlation between consanguinity status and admission diagnosis

Category	Consanguinity	Non-consanguinity
Infections of the fetus or newborn	0	4 (0.26)
Respiratory disorders specific to the perinatal or neonatal period	3 (0.75)	3 (0.2)
Bleeding or hematological disorders of the fetus or newborn	0	4 (0.26)
Structural–developmental abnormalities primarily affecting a bodily system	1 (0.25)	2 (0.13)
Fetuses or newborns affected by maternal factors or by complications of pregnancy, labor, or delivery	0	1 (0.06)
Newborn disorders related to the length of gestation or fetal growth	0	1 (0.06)

```

bulk("scheme.admission")
  -> map((item) => {
    diagnosis: item.value.item.admission.reason
  -> ancestor("6", 3) -> interpolate(),
    consanguinity:
item.value.item.history.obstetric_history.consanguini
nity,
  })
  -> <[ {
    diagnosis: string,
    consanguinity: integer,
    non_consanguinity: integer,
  }*>@group(["diagnosis"], (item) => {
    diagnosis: item[0].diagnosis,
    consanguinity: item -> filter((group) =>
group.consanguinity) -> count(),
    non_consanguinity: item -> filter((group)
=> !group.consanguinity) -> count(),
  })

```

Figure 184: correlation between consanguinity status and admission diagnosis query

Query explanation: all records were pulled from “scheme.admission” datatable, each record was mapped for admission diagnosis value (originally an ontology code that gets interpolated) along with consanguinity value, then all records were grouped by diagnosis and consanguinity values, each group was counted.

- **Parity and gestation mean by admission status**

High parity and gestation counts were correlated with high mortality rates.

Table LXVI: parity and gestation mean by admission status

Status	Parity (mean)	Gestations (mean)
Alive	2.09	2.09
Deceased	3.75	3.75

```
bulk("scheme.admission")
  -> map((item) => {
    status: item.value.item.admission.status ->
    interpolate(),
    gestations:
    item.value.item.history.obstetric_history.gestation
  },
  parities:
  item.value.item.history.obstetric_history.parity,
  })
  -> <[ {
    status: string,
    gestations: float(digit:max=2),
    parities: float(digit:max=2),
  }*>@group(["status"], (item) => {
    status: item[0].status,
    gestations: item -> <[number*>@map((entry)
=> entry.gestations) -> mean(),
    parities: item -> <[number*>@map((entry)
=> entry.parities) -> mean(),
  })
```

Figure 185: parity and gestation mean by admission status query

Query explanation: all records were pulled from “scheme.admission” datatable, each record was mapped for admission status value (originally an ontology code that gets interpolated) along with parity and gestation value, then all records were grouped by status, mean aggregates were calculated for each group.

Discussion

I. Project impact

Data quality is a critical determinant of any research quality: proper implementation of data exploration and data querying frameworks help to retrieve quality reports; data governance help protect, secure data and makes it transparent, portable, and accessible; these attributes are typical constraints of the research process.

Health informatics offers significant improvements in healthcare; it helps deliver safer, accurate, and consistent healthcare.

Impact of introducing health informatics on healthcare [15,34,35,36,37,40,41,126,127]:

- **Reduce typography and validity errors:** thanks to the data quality assertions.
- **Offer rich data capture interface:** type visual inference enables the consistency between datatypes and input appearances, *metatype* ecosystem support all major programming data structures.
- **Improve the readability of medical records.**
- **Centralize the writing experience:** throughout the use of consistent type inference, integrated systems, and user experience considerations.
- **Democratize access to data and charts:** throughout the use of security guidelines such as access control, user management, and identity management.
- **Rich integrations:** integration of formulas, scores, and assessment calculators in real-time.
- **More insights on patient's data:** offer more insights on patient's data such as plotting and active deep learning algorithms.

Health informatics help digitize the classic paper trails, establish centralized data solutions, reduce errors, automate operational processes, and automating research processes.

Impact of introducing health informatics on research [34,103,126,127,128]:

- **Digitize paper documents:** which makes data more transparent, available, and portable.
- **Assert data quality guarantees:** by the creation of a data quality framework and assessment measures.
- **Enable data governance:** thanks to the access control, user management, and session management technologies.
- **Shorten research lifecycle:** thanks to data automation.
- **Establish a continuous research pipeline:** data accessibility, exploration and querying pipelines, real-time queries, and immediate feedback are all factors that help establish a continuous research pipeline and, at the aftermost, a completely autonomous research pipeline.
- **Makes further automation techniques possible:** other automation frameworks can be

introduced, such as natural language, processing, machine learning, and expert systems.

Impact of automation on research [16,49,129,130]:

- **Short lifecycle iterations:** automating data querying, data exploration, and by asserting data quality guarantees.
- **Immediate feedback:** thanks to the real-time query execution, results are returned immediately, and thus feedback can be taken in real-time.
- **Reduce financial spending:** data is more accessible, centralized, and transparent.
- **Help overcome human resources shortage:** human activity is automated, and thus human intervention is no longer required or minimal.
- **Increase research reuse thanks to standardization:** reports and results can be reused by the creation of reuse pipelines.
- **Uncover conclusions out of existing research results:** results can be augmented and reused.
- **Uncover unseen conclusions or complex correlation models:** more data attributes are available, and thus, more correlations can be created; advanced data exploration, analysis, and querying algorithms can be used to draw tricky conclusions.

II. Data quality

Data quality refers to the state of qualitative or quantitative pieces of information [131,132].

Table LXVII: data quality dimensions [48]

Dimension	Description
Accessibility	The extent to which data is available or easily and quickly retrievable
Appropriate amount of data	The extent to which the volume of data is appropriate for the task at hand
Believability	The extent to which data is regarded as true and credible
Completeness	The extent to which data is not missing and is of sufficient breadth and depth for the task at hand
Concise representation	The extent to which data is compactly represented and to which data is represented in the same format
Ease of manipulation	The extent to which data is easy to manipulate and apply to different tasks
Free-of-error	The extent to which data is correct and reliable

Interpret-ability	The extent to which data is in appropriate languages, symbols, and units and the definitions are clear
Objectivity	The extent to which data is unbiased, unprejudiced, and impartial
Relevancy	The extent to which data is applicable and helpful for the task at hand
Reputation	The extent to which data is highly regarded in terms of its source or content
Security	The extent to which access to data is restricted appropriately to maintain its security
Timeliness	The extent to which the data is sufficiently up-to-date for the task at hand
Understand-ability	The extent to which data is easily comprehended
Value-added	The extent to which data is beneficial and provides advantages from its use

Data in our implementation is easily accessible through query requests; access to data was augmented with helper functions through a rich domain-specific language.

Our implementation can accommodate huge amounts of data. We've designed it to be scalable and shard-able; Data shapes are hugely compact thanks to the JSON serialization format and the concept-code separation provided by the ontology container.

Believability is granted by two factors: the strength of data quality guarantees asserted by our implementation, and by the credibility and reputation of the medical staff.

Data collected by our implementation is complete, exhaustive, and thorough; it covers all aspects of the EHR, HMS, and PMS; The ontology is exhaustive regarding the covered themes.

Data representation is concise thanks to the features provided by the universal content management systems.

Data can be easily manipulated thanks to *metalamba* rich features; all primary data transformation operations are supported; dozens of helper functions are available.

Data collected by our implementation are always syntactically valid and correct, but its lexical correctness, objectiveness can be enforced through peer-reviewing. Data collected by our implementation is relevant to all primary medical research studies.

Security is asserted by our development paradigm: security by design, security architecture, security in-depth, and security measures.

Data collected is always up-to-date; time points and duration points are calculated in realtime. Understandability is ensured thanks to the universal nature of the JSON data serialization standard.

III. Big data

Big data is a field that treats ways to analyze, extract or store data sets that are too large or complex to be dealt with by traditional data-processing application software [133,134,135]; challenges include capturing data, data storage, data analysis, search, sharing, transfer, visualization, querying, updating, information privacy, and data source [133,134,135].

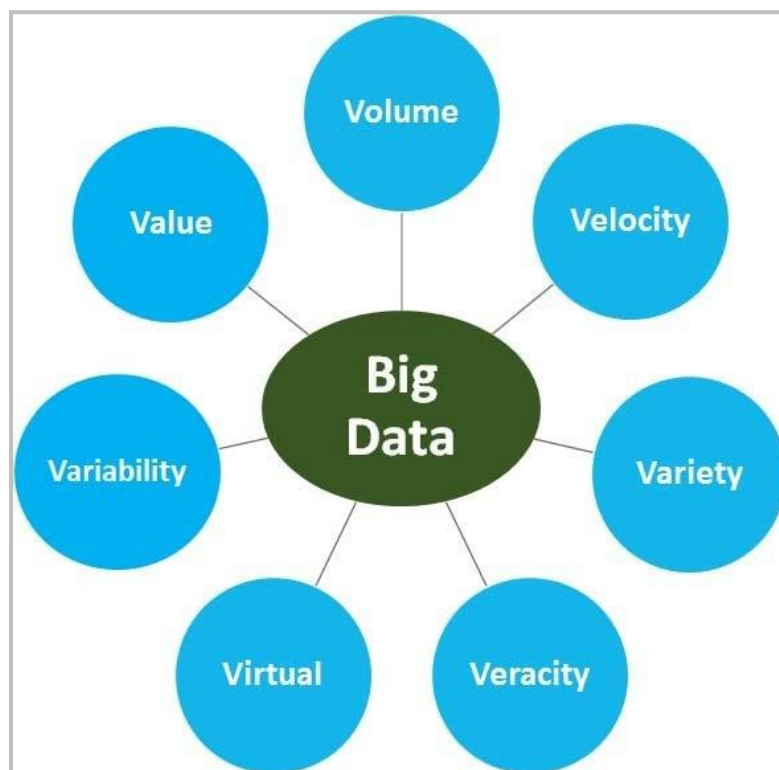


Figure 186: the seven "V"s of Big Data [136]

Big data can be described by the following characteristics [134,135]:

- **Volume:** the quantity of generated and stored data. The size of the data determines the value and potential insight and whether it can be considered big data or not.
- **Variety:** the type and nature of the data. This helps people who analyze it to use the resulting

insight effectively. Big data draws from text, images, audio, video; plus, it completes missing pieces through data fusion.

- **Velocity:** the speed at which the data is generated and processed to meet the demands and challenges that lie in the path of growth and development. Big data is often available in real-time. Compared to small data, big data are produced more continually. Two kinds of velocity related to big data are the frequency of generation and the frequency of handling, recording, and publishing.
- **Veracity:** the extended definition for big data refers to the data quality and the data value. The data quality of captured data can vary greatly, affecting accurate analysis.
- **Exhaustive:** whether the entire system is captured or recorded or not.
- **Fine-grained and uniquely lexical:** respectively, the proportion of each element's specific data per element collected and if the element and its characteristics are properly indexed or identified.
- **Relational:** if the data collected contains common fields that would enable a conjoining or meta-analysis of different data sets.
- **Extensional:** if new fields in each element of the data collected can be added or changed easily.
- **Scalability:** if the size of the data can expand rapidly.
- **Value:** the utility that can be extracted from the data.
- **Variability:** it refers to data whose value or other characteristics are shifting to the context they are being generated.

Our system asserts all Big Data attributes, and thus it is considered as a Big Data solution:

- **Volume:** massive data amounts can be ingested, datatables can be sharded, and data is compressed thanks to the use of JSON serialization standard and ontology concept encoding.
- **Variety:** all major datatypes are supported thanks to *metatypeecosystem*.
- **Velocity:** data capture duration is minimal thanks to the rich data entry interface, and thanks to the user experience consideration taken in mind; Data queries are executed in real-time and are always fresh.
- **Veracity:** thanks to the validity constraints and the rich, exhaustive, hierarchic, and restrictive nature of the ontology system.
- **Exhaustive:** all data related to the medical record are captured, the ontology database is exhaustive on the covered themes.
- **Fine-grained and uniquely lexical:** the ontology container capture data in a fine-grained manner thanks to its hierarchic and semantic nature.
- **Relational:** record relations are captured thanks to the scheme system; list and object data structures are supported as well.
- **Extensional:** thanks to the elastic scheme container, new data definitions can be added easily in real-time with zero downtime.
- **Value:** all data capture by the system is transparent, available, portable, and governed; and

is accessible to the query engine thanks to the universal data warehouse system.

- **Variability:** new scheme changes can be accommodated easily, in real-time, with zero downtime.

IV. Practical constraints

Three types of constraints complicate the computerization process [17]:

- Computerization requires complex modeling of medical data and knowledge;
- Logistical, technical and financial problems;
- Human problems which are often underestimated.

On the one hand, the initial models, modeled on the linear organization of "paper" records, quickly showed their shortcomings, hence the need for a different and efficient model making a balance between structured data and free text, where the difficulty lies practically. On the other hand, computerization requires a large budget, so we must convince decision-makers of the contributions of information technologies to invest enormous financial means in the research and logistics necessary for computerization [30].

In addition, since its inception, computing has raised security, practitioner acceptability and maintenance challenges. But the development of computer techniques makes their use safer, more attractive and easier.

The implementation of the electronic medical record is often faced with resistance from healthcare professionals. This is fueled by fears about the rigidity of computer systems, changes in habits and work organization. The time needed for adaptation and learning could be at the expense of that spent with patients [137,138].

Also fears of inaccessibility to the record in the event of a computer failure, the different management of the patient relationship where the computer "interferes", sometimes make practitioners reluctant [32].

V. French experience

Created by the French law of August 13, 2004 relating to health insurance, the personal health record is a service designed to help improve coordination, continuity and therefore the quality of care.

It is a single, computerized personal health record filed with an approved health data host. It is made available to any beneficiary of health insurance. As a result, he is the holder of this record and can therefore designate the host of his choice and control the process of opening, consulting and supplying his record [139].

It contains [139]:

- General medical data: history, allergies, vaccinations, consultation histories, summaries...;
- Care data: reports of additional examinations and diagnostic and therapeutic procedures;
- Prevention data: risk factors, preventive treatments, etc.;
- Medical imaging documents;
- An expression space for the holder.

The personal medical record has these advantages [139]:

- Allow the patient to transmit his health data to any caregiver (especially if the disease is chronic with several stakeholders) and at any time, even in an emergency, even if he is unconscious and even when he is in displacement;
- Avoid redundancy in the prescription of additional examinations (reduces the pain for the patient and the cost for society);
- Be a vehicle for health education (the patient, better informed, better taken care of);
- Encourage changes in medical behavior (learn to communicate with other caregivers, discuss and question their habits).

But the PHR cannot replace the professional medical record, and risks violating the confidentiality of the data and certain fundamental rights such as the right to hide data and the right not to know one's diagnosis.

There is also the risk of constituting data banks, the use of which cannot be predicted in the future by the pharmaceutical industry, insurance, security, the state, etc. The priority given to the computer characteristics of the file risks accumulating purely clinical and biological data, without precisely taking into account the clinical dimension of the doctor / patient relationship [35].

All these limits were raised by the National Consultative Committee of Ethics for Life Sciences and Health after it was seized in 2008 by the Minister of Health about the development of information technologies in the medical field. This committee affirmed that the PHR in its current

conception cannot be adopted for every citizen, at the national level as it does not meet the established objectives, while its cost of implementation is very high. Corrective measures were then proposed in his report [139].

After years of experimentation, the EHR found itself confronted with difficulties linked to its environment: non-communicating information systems, insufficient development of information systems for the production of care in the hospital, compartmentalized care organization, dispersed and inconsistent industrial offer, fragmented governance in the form of multiple players whose areas of intervention included significant areas of adherence, etc.

This is why the French Minister of Health, at the end of various investigative works commissioned on the subject, announced the changes which were to accompany the relaunch of the EHR. The Agency for Shared Health Information Systems (ASIP Santé) and the Performance Support Agency for Health and Medico-Social Establishments (ANAP) were therefore created, which will work in close consultation to construct and implement the conditions favorable to the deployment of shared health information systems in line with a national framework, taking into account the steps taken in the course of its progress and responding to the needs expressed by the various actors in the health field [140].

VI. Research automation

Automation can be applied through many research processes; it begins by ensuring data quality and data governance, data warehouses can be created to enforce those guarantees, data exploration, and results publishing can be automated as well [53].

Research aspects that can be automated [53]:

- **Data quality:** data quality assertions like its validity, veracity, variety ... etc.
- **Data governance:** access and data management frameworks.
- **Headline suggestion:** research subject suggesting and literature research.
- **Data exploration:** data exploration before data analysis.
- **Data querying:** data analysis and query execution.
- **Reports writing:** the writing of ready to publish research reports.
- **Citations management:** automated annotations, citations quality assessment ... etc.
- **Research quality:** research quality assessment, writing style assessment, critical reviewing of researches.
- **Peer reviewing:** evaluation of research by one or more people with similar competencies as the producers of the work.

1. Headline suggestion

Research subject suggesting is a task that always involves literature research and knowledge base research. The purpose of a literature review is to [141]:

- Place each work in the context of its contribution to understanding the research problem being studied.
- Describe the relationship of each work to the others under consideration.
- Identify new ways to interpret prior research.
- Reveal any gaps that exist in the literature.
- Resolve conflicts among seemingly contradictory previous studies.
- Identify areas of prior scholarship to prevent duplication of effort.
- Point the way in fulfilling a need for additional research.
- Locate your research within the context of existing literature.

This process can be automated by developing universal knowledge graphs and by employing natural-language generation algorithms.

The construction of knowledge graphs requires knowledge data-sets; data-sets can be pulled by using data mining techniques such as API integration, data scraping, data augmentation ... etc.

Pulled data-sets are prone to data quality inconsistencies. Therefore, they must be cleansed, normalized, and must have their quality assessed.

Natural language processing (NLP) is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, particularly how to program computers to process and analyze large amounts of natural language data [88,142].

Natural-language understanding/interpretation (NLU/NLI) is a subtopic of natural-language processing in artificial intelligence that deals with machine reading comprehension [142].

Transformers are a new natural language deep learning model (AI) capable of representing language models [143], they are designed to handle sequential data, such as natural language, for tasks such as translation and text summarization [143].

BERT transformer was a breakthrough in natural language understanding; it is a technique for natural language processing (NLP) pre-training developed by Google. BERT was created and published in 2018 by Jacob Devlin and his colleagues from Google. Google is leveraging BERT to better understand user searches [144].

To make an automated research headline generator, the suggested artificial intelligence

model would be a transformer trained on a knowledge base (as an input) to make a knowledge graph, to output research headlines through a generative–adversarial neural network (GAN).

2. Data exploration

Data exploration is the initial data analysis to understand the characteristics of the data including data size, completeness of the data, the correctness of the data, possible relationships amongst data elements, or files/tables in the data [145].

Data exploration is typically conducted using a combination of automated and manual activities. Automated activities can include data profiling or data visualization, or tabular reports to give the analyst an initial view of the data and an understanding of key characteristics [145].

Data exploration techniques often involve descriptive statistics and unsupervised learning techniques [145].

In descriptive statistics, summary statistics are used to summarize a set of observations, to communicate the largest amount of information as simply as possible. Statisticians commonly try to describe the observations in [146]:

- A measure of location or central tendency, such as the arithmetic mean.
- A measure of statistical dispersion like the standard mean absolute deviation.
- A measure of the shape of the distribution like skewness or kurtosis.
- If more than one variable is measured, statistical dependence is measured, such as a correlation coefficient.

Our implementation includes a summary plot type and a summary helper expression function that automatically compute the sever–number summary for a vector of numbers.

Unsupervised learning is a type of machine learning that looks for previously undetected patterns in a data set with no pre–existing labels and minimal human supervision. In contrast to supervised learning that usually uses human–labeled data, unsupervised learning, also known as self–organization, allows for modeling probability densities over inputs.

Two of the main methods used in unsupervised learning are principal component and cluster analysis. Cluster analysis is used in unsupervised learning to group or segment datasets with shared attributes to extrapolate algorithmic relationships [147].

Some of the most common algorithms used in unsupervised learning include: (1) Clustering, (2) Anomaly detection, (3) Neural Networks, and (4) Approaches for learning latent variable models. Each approach uses several methods, as follows [147]:

- Clustering:
 - o hierarchical clustering.
 - o k-means.
 - o Mixture models.
 - o DBSCAN.
 - o OPTICS algorithm.
- Anomaly detection:
 - o Local Outlier Factor.
- Neural Networks:
 - o Autoencoders.
 - o Deep Belief Nets.
 - o Hebbian Learning.
 - o Generative adversarial networks.
 - o Self-organizing map.
- Approaches for learning latent variable models such as:
 - o Expectation-maximization algorithm (EM).
 - o Method of moments:
 - Blind signal separation techniques.
 - Principal component analysis.
 - Independent component analysis.
 - Non-negative matrix factorization.
 - Singular value decomposition.

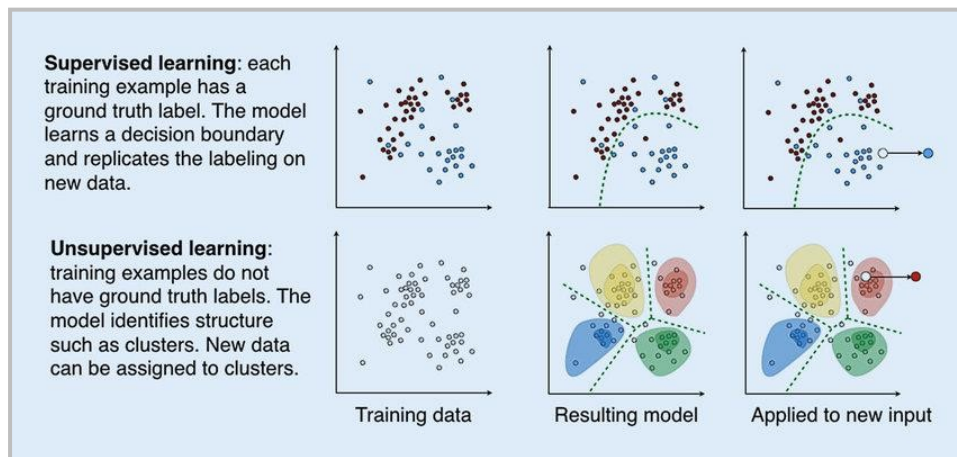


Figure 187: supervised learning vs. unsupervised learning [147]

3. Data querying

Query languages or data query languages (DQLs) are computer languages used to make queries in databases and information systems [148,149].

Querying is the process of writing, designing, and executing data fetching and manipulation of instructions written on domain-specific languages [148].

Writing data queries can be tedious; it requires knowledge of the data query language, data structure, and data scheme.

Some data query languages can have a steep learning curve, data structures can be ambiguous, and schemes can be undocumented.

Table LXVIII: query writing constraints

Constraints	Description
Syntactical correctness	Language grammar must be strictly followed
Lexical correctness	Language instructions must be written correctly
Types correctness	Datatype for inputs, outputs, function calls must be respected
Time complexity	The duration on which query may need to be computed must be within reasonable limits
Space complexity	The memory space the query may use must be within reasonable limits

A Natural Language Interface (NLI) facilitates users to pose queries to retrieve information from a database without using any artificial language such as the Structured Query Language (SQL). Several applications in various domains, including healthcare, require elaborating structured data having information on the text [149].

Two primary frameworks exist for natural language to database query conversion [149]:

- **Rule-based and syntax analysis:** employs traditional natural language processing techniques such as tokenization, stemming, lemmatization, POS tagging, and parsing to construct abstract syntax trees that will be converted to corresponding query syntax.
- **Machine learning:** employs deep learning techniques such as recurrent neural networks, transforms, generative adversarial networks, convolution neural networks, autoencoders ... etc.

Data can be directly streamlined into artificial intelligence agents:

- **Gradient tree boosting:** it consists of the construction of decision trees using gradient descent algorithms [150]; e.g. they can be used to construct decision trees when the patients

present certain conditions.

- **Regression neural networks:** it consists of a set of machine learning methods that allow us to predict a continuous outcome variable (y) based on the value of one or multiple predictor variables (x). Briefly, the regression model's goal is to build a mathematical equation that defines y as a function of the x variables [151]; e.g., can predict a patient expected parameter value given specific parameters or conditions.
- **Classification neural networks:** it is a supervised learning concept that categorizes data sets into classes [152]; e.g., can classify patients according to specific parameters, traits, or conditions.
- **Generative neural networks:** it is an unsupervised learning task in machine learning that involves automatically discovering and learning the regularities or patterns in input data in such a way that the model can be used to generate or output new examples that plausibly could have been drawn from the original dataset [153]; e.g., can be used to suggest new treatments for new/existing diseases.
- **Convolutional neural networks:** are a category of Neural Networks that have proven very effective in image recognition and classification [154]; e.g., can be used to classify X-ray or CT-Scan images for diseases.
- **Recurrent neural networks/Transformer:** derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable-length sequences of inputs [143,155]; e.g. can be used to generate reports for anatomy pathology, X-ray, CT-scan images.

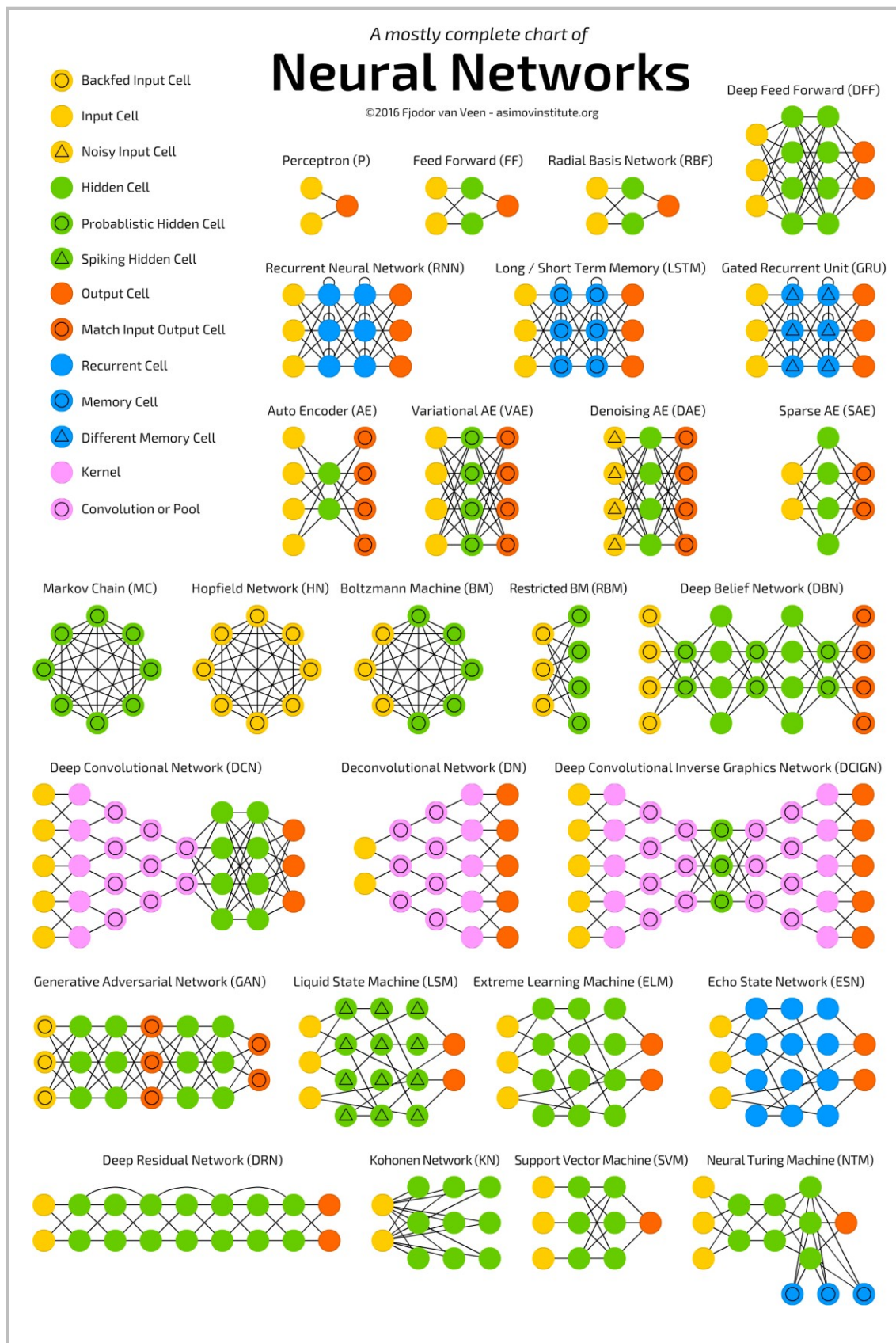


Figure 188: types of artificial neural networks

4. Reports writing

The production process, controlled by a production editor or publisher, then takes an article through copy editing, typesetting, inclusion in a specific issue of a journal, and then printing and online publication. Academic copy editing seeks to ensure that an article conforms to the journal's house style, that all of the referencing and labeling are correct, and that the text is consistent and legible; often, this work involves substantive editing and negotiating with the authors [156,157].

The author will review and correct proofs at one or more stages in the production process. The full automation of the proof correction cycles has only become possible with the onset of online collaborative writing platforms, such as Authorea, Google Docs, and various others, where a remote service oversees the copy-editing interactions of multiple authors and exposes them as explicit, actionable historical events [156].

Several technologies exist that can automate the process report writing, Document formatting standards like LaTeX or markdown that enable creating standardized rich documents using plain text instructions.

Markdown is a lightweight markup language with plain-text-formatting syntax, created in 2004 by John Gruber with Aaron Swartz. Markdown is often used for formatting readme files, for writing messages in online discussion forums, and to create rich text using a plain text editor [158].

LaTeX is a software system for document preparation. When writing, the writer uses plain text instead of the formatted text found in "What You See Is What You Get" word processors like Microsoft Word, LibreOffice Writer, and Apple Pages. The writer uses markup tagging conventions to define the general structure of a document (such as article, book, and letter), to stylize text throughout a document (such as bold and italics), and to add citations and cross-references [159].

Markdown documents allow the automated extraction of headlines, tables, figures; tables and figures can be auto-incremented automatically according to any numbering format; table of contents can be automatically generated with links to the corresponding paragraph.

LaTeX can display any mathematical formula, in addition to its being the publishing standard for many journals.

Natural language processing algorithms can be used to write paragraphs, summarize headlines, abstracts, detect plagiarism, speech tone, voice, or compute metrics such as reading time, reading difficulty, word count, or auto-complete sentence, spell check, suggest synonyms, simplify phrases ... etc.

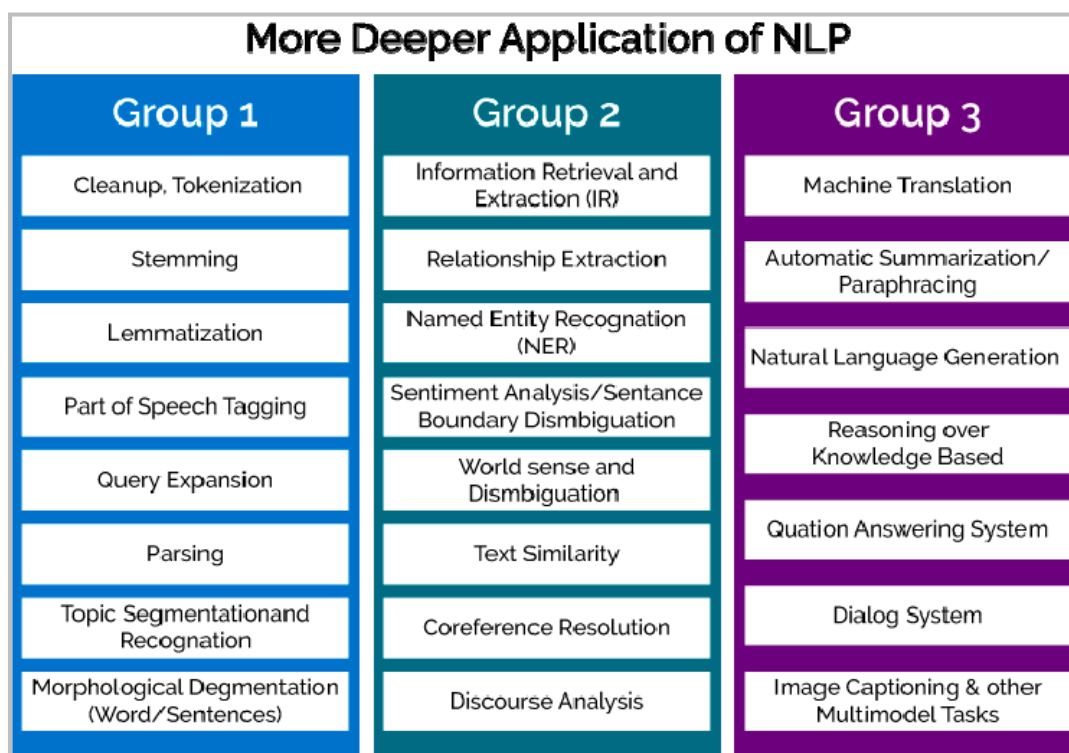


Figure 189: natural language processing applications [88,142,155]

5. Citations management

Enumerative bibliographies are based on a unifying principle such as creator, subject, date, topic, or other characteristics. An entry in an enumerative bibliography provides the core elements of a text resource, including a title, the creator(s), publication date, and place of publication [160].

Reference management software, citation management software, or bibliographic management software is software for scholars and authors to use for recording and utilizing bibliographic citations (references) and managing project references either as a company or an individual. Once a citation has been recorded, it can be used repeatedly to generate bibliographies, such as lists of references in scholarly books, articles, and essays. The development of reference management packages has been driven by the rapid expansion of scientific literature [161].

These software packages typically consist of a database in which full bibliographic references can be entered, plus a system for generating selective lists of articles in the different formats required by publishers and scholarly journals. Modern reference management packages can usually be integrated with word processors so that a reference list in the appropriate format is produced automatically as an article is written, reducing the risk that a cited source is not included in the reference list. They will also have a facility for importing the details of publications from bibliographic databases [161].

6. Research quality and peer-reviewing

Quality research most commonly denotes the scientific process, including all aspects of study design; in particular, it relates to the judgment regarding the match between the methods and questions, selection of subjects, measurement of outcomes, and protection against systematic bias, nonsystematic bias, and inferential error. Principles and standards for quality research designs are commonly found in texts, reports, essays, and guides to research design and methodology, and so on [162,163].

Besides, quality assessment plays a vital role in the research community. It enlightens crucial decisions on the funding of projects, teams, and whole institutions, on how research is conducted, on recruitment and promotion, on what is published or disseminated, and on what researchers and others choose to read. It makes trust in the work of the research community. Quality is, of course, not a straightforward concept. The Oxford English Dictionary (OED) defines it as the nature or standard of something as measured against other things of a similar kind, and especially the degree of excellence it possesses [162].

Standards for assessing the quality of research [162,163]:

- Pose a significant, important question that can be investigated empirically, and that contributes to the knowledge base.
- A well-defined research topic and a clear hypothesis.
- Test questions that are linked to relevant theory.
- Apply methods that best address the research questions of interest.
- Base research on transparent chains of inferential reasoning supported and justified by complete coverage of the relevant literature.
- Provide the necessary information to reproduce or replicate the study.
- Ensure the study design, methods, and procedures are sufficiently transparent and ensure an independent, balanced, and objective approach to the research.
- Provide a sufficient description of the sample, the intervention, and any comparison groups.
- Use appropriate and reliable conceptualization and measurement of variables.
- Evaluate alternative explanations for any findings.
- High-quality data were fit for their intended use and reliable, valid, relevant, and accurate.
- Findings of the study written in a way which brings clarity to important issues.
- Tables and graphics which are clear, accurate, and understandable with appropriate labeling of data values, cut points, and thresholds.
- Include both statistical significance results and effect sizes when possible.
- The conclusions and recommendations both logical and consistent with the findings.
- Assess the possible impact of systematic bias.
- Submit research to a peer-review process.
- Adhere to quality standards for reporting (i.e., clear, cogent, complete).
- Is respectful to people with other perspectives.

- Provides adequate references.
- Attempts to honestly present all perspectives.

Natural-language processing algorithms can be used to compute metrics such as voice, tone, formality, conciseness; NLP algorithms can detect plagiarism, classify deception, spot fraud, detected incited sentences, and detect stealing.

Citations can be checked for their style, their publisher's impact factor that can be crossed over some predefined threshold ... etc.

Overlooked features for assessing the quality of research [162]:

- Research questions are designed to reach a particular conclusion.
- Alternative perspectives or contrary findings are ignored or suppressed.
- Data and analysis methods are biased.
- Conclusions are based on faulty logic.
- Limitations of analysis are ignored, and the implications of results are exaggerated.
- Critical data and analysis details are unavailable for review by others.
- Researchers are unqualified and unfamiliar with specialized issues.
- Citations are primarily from special interest groups or popular media, rather than from peer-reviewed professional and academic organizations.

VII. Deployment

1. Requirements

The application is a Typescript (Javascript) application that runs on Node.JS server runtime; the application was bundled using Webpack and packaged on a Docker container lightweight image.

There is only one software requirement, Docker. Docker is a platform set as a service (PaaS) product that uses OS-level virtualization to deliver software packages called containers. Containers are isolated from one another and bundle their software, libraries, and configuration files; they can communicate through well-defined channels. All containers are run by a single operating system kernel and therefore use fewer resources than virtual machines.

Minimum hardware requirements are:

- **Processor:** Sandy bridge core i3, Phenom II x3, AMD a9, or equivalent.
- **Memory:** 4 GB DDR3, 800 MHz.
- **Disk:** 100 GB SSD or HDD.
- **Network:** 1MBPS internet speed.

If used, Docker container technology will be the only software required dependency and will

take care of all sub-dependencies automatically.

If the application is to be installed directly opting out of Docker technology, then NodeJS, NPM, ImageMagick dependencies must be met; NPM will install the remaining sub dependencies.

2. Deployment at scale

The application can be made available for all hospital departments with few minor changes:

- It already comes with an exhaustive ontology database.
- Its EHR scheme adapts to the admission department.
- It supports all hospital staff personas.
- More personas can be added easily.
- It can be easily integrated with radio imaging devices, a DICOM previewer can be easily added.
- It can be easily integrated with laboratory devices to publish results automatically.
- Its container-based architecture was built with scalability in mind.
- It supports container isolation and database sharding.
- It comes with a continuous integration pipeline.

To be deployed at a massive scale:

- Access control rules must be fine-tuned at the attribute-level to enforce security.
- Firewalls must be enforced, and kernels must be secured.
- Kubernetes integration must be written.
- Hardware/infrastructure must be made available.
- Autoscaling and scalability topology must be designed and implemented.
- Resilience and SLA profiles must be designed and implemented.
- Audit trails must be added.
- Application monitoring software must be installed.

Conclusion

Meta-research is the study of research through the use of research methods. It aims to reduce waste and increase research quality in all fields. Data quality, governance, and querying are pivotal elements in data and research lifecycles; thus, automating them helps establish continuous research pipelines.

Our objective was to introduce health informatics solutions combined with data science solutions to establish a continuous research pipeline and assert data quality attributes and data governance.

The software development lifecycle dictated our methodology; we've employed the Agile methodology and Scrum framework in our workflow; standardization, structuring, software quality, security, legislative, and user experience considerations were our major concerns.

We used “meta” suite to create a multi-purpose content management system and a universal data management system, and we've encoded content schemes to instruct visual components.

The resulting system consists of 8 major containers: hospital management system, practice management system, electronic medical record, multi-purpose content management system, universal data management system, identity, and access management system, internationalization system in addition to the ontology container.

We've developed an ontology container to augment our entry system capabilities, which represent concepts in a tree-like structure; each concept is encoded with a prefix string following a Trie data structure, which enables static operations in $O(1)$ time; We've added to the database 270,516 concepts for anatomy, diagnoses, findings, interventions, procedures, medications, organisms, substances in addition to dozen other attributes.

The software was tested by three levels of testing: unit, integration, and acceptance testing; Our acceptance testing objectives were to submit, investigate, and query data using real medical records. To assert these objectives, we've submitted 19 real patient medical records for a two months' period; Submitted data include patient demographics, history notes, examination notes, prescriptions sheets, requested procedures, interventions, diagnoses, and progress notes; We've created and executed data queries exploring clinical, para-clinical, diagnoses, prognosis, and therapeutic parameters.

Introducing health informatics impacts clinical research directly and improves healthcare by digitizing paper trails, asserting quality attributes, asserting data governance, enabling data querying and smart healthcare, and opening the door for other automation techniques. Many research aspects can be automated thanks to artificial intelligence, natural language processing algorithms, and expert systems.

Appendix

I. Appendix 1: Scrum framework

A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value [63].

Scrum is:

- Lightweight.
- Simple to understand.
- Difficult to master.

Scrum is a process framework that has been used to manage work on complex products since the early 1990s. Scrum is not a process, technique, or definitive method. Rather, it is a framework within which you can employ various processes and techniques. Scrum makes clear the relative efficacy of your product management and work techniques so that you can continuously improve the product, the team, and the working environment.

The Scrum framework consists of Scrum Teams and their associated roles, events, artifacts, and rules. Each component within the framework serves a specific purpose and is essential to Scrum's success and usage.

The rules of Scrum bind together the roles, events, and artifacts, governing the relationships and interaction between them. The rules of Scrum are described throughout the body of this document.

Specific tactics for using the Scrum framework vary and are described elsewhere.

1. Uses of Scrum

Scrum was initially developed for managing and developing products. Starting in the early 1990s, Scrum has been used extensively, worldwide, to:

- Research and identify viable markets, technologies, and product capabilities;
- Develop products and enhancements;
- Release products and enhancements, as frequently as many times per day;
- Develop and sustain Cloud (online, secure, on-demand) and other operational environments for product use; and,
- Sustain and renew products.

Scrum has been used to develop software, hardware, embedded software, networks of interacting function, autonomous vehicles, schools, government, marketing, managing the

operation of organizations and almost everything we use in our daily lives, as individuals and societies.

As technology, market, and environmental complexities and their interactions have rapidly increased, Scrum's utility in dealing with complexity is proven daily.

Scrum proved especially effective in iterative and incremental knowledge transfer. Scrum is now widely used for products, services, and the management of the parent organization.

The essence of Scrum is a small team of people. The individual team is highly flexible and adaptive. These strengths continue operating in single, several, many, and networks of teams that develop, release, operate and sustain the work and work products of thousands of people. They collaborate and interoperate through sophisticated development architectures and target release environments.

When the words "develop" and "development" are used in the Scrum Guide, they refer to complex work, such as those types identified above.

2. Scrum Theory

Scrum is founded on empirical process control theory, or empiricism. Empiricism asserts that knowledge comes from experience and making decisions based on what is known. Scrum employs an iterative, incremental approach to optimize predictability and control risk. Three pillars uphold every implementation of empirical process control: transparency, inspection, and adaptation.

A. Transparency

Significant aspects of the process must be visible to those responsible for the outcome. Transparency requires those aspects be defined by a common standard so observers share a common understanding of what is being seen.

For example:

- A common language referring to the process must be shared by all participants; and,
- Those performing the work and those inspecting the resulting increment must share a common definition of "Done".

B. Inspection

Scrum users must frequently inspect Scrum artifacts and progress toward a Sprint Goal to detect undesirable variances. Their inspection should not be so frequent that inspection gets in the way of the work. Inspections are most beneficial when diligently performed by skilled inspectors at the point of work.

C. Adaptation

If an inspector determines that one or more aspects of a process deviate outside acceptable limits, and that the resulting product will be unacceptable, the process or the material being processed must be adjusted. An adjustment must be made as soon as possible to minimize further deviation.

Scrum prescribes four formal events for inspection and adaptation, as described in the Scrum Events section of this document:

- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

3. Scrum Values

When the values of commitment, courage, focus, openness and respect are embodied and lived by the Scrum Team, the Scrum pillars of transparency, inspection, and adaptation come to life and build trust for everyone. The Scrum Team members learn and explore those values as they work with the Scrum events, roles and artifacts.

Successful use of Scrum depends on people becoming more proficient in living these five values. People personally commit to achieving the goals of the Scrum Team. The Scrum Team members have courage to do the right thing and work on tough problems. Everyone focuses on the work of the Sprint and the goals of the Scrum Team. The Scrum Team and its stakeholders agree to be open about all the work and the challenges with performing the work. Scrum Team members respect each other to be capable, independent people.

4. The Scrum Team

The Scrum Team consists of a Product Owner, the Development Team, and a Scrum Master. Scrum Teams are self-organizing and cross-functional. Self-organizing teams choose how best to accomplish their work, rather than being directed by others outside the team. Cross-functional teams have all competencies needed to accomplish the work without depending on others not part of the team. The team model in Scrum is designed to optimize flexibility, creativity, and productivity. The Scrum Team has proven itself to be increasingly effective for all the earlier stated uses, and any complex work.

Scrum Teams deliver products iteratively and incrementally, maximizing opportunities for feedback. Incremental deliveries of "Done" product ensure a potentially useful version of working product is always available.

A. The Product Owner

The Product Owner is responsible for maximizing the value of the product resulting from work of the Development Team. How this is done may vary widely across organizations, Scrum Teams, and individuals.

The Product Owner is the sole person responsible for managing the Product Backlog. Product Backlog management includes:

- Clearly expressing Product Backlog items;
- Ordering the items in the Product Backlog to best achieve goals and missions;
- Optimizing the value of the work the Development Team performs;
- Ensuring that the Product Backlog is visible, transparent, and clear to all, and shows what the Scrum Team will work on next; and,
- Ensuring the Development Team understands items in the Product Backlog to the level needed.

The Product Owner may do the above work, or have the Development Team do it. However, the Product Owner remains accountable.

The Product Owner is one person, not a committee. The Product Owner may represent the desires of a committee in the Product Backlog, but those wanting to change a Product Backlog item's priority must address the Product Owner.

For the Product Owner to succeed, the entire organization must respect his or her decisions. The Product Owner's decisions are visible in the content and ordering of the Product Backlog. No one can force the Development Team to work from a different set of requirements.

B. The Development Team

The Development Team consists of professionals who do the work of delivering a potentially releasable Increment of "Done" product at the end of each Sprint. A "Done" increment is required at the Sprint Review. Only members of the Development Team create the Increment.

Development Teams are structured and empowered by the organization to organize and manage their own work. The resulting synergy optimizes the Development Team's overall efficiency and effectiveness.

Development Teams have the following characteristics:

- They are self-organizing. No one (not even the Scrum Master) tells the Development Team how to turn Product Backlog into Increments of potentially releasable functionality;
- Development Teams are cross-functional, with all the skills as a team necessary to create a

product Increment;

- Scrum recognizes no titles for Development Team members, regardless of the work being performed by the person;
- Scrum recognizes no sub-teams in the Development Team, regardless of domains that need to be addressed like testing, architecture, operations, or business analysis; and,
- Individual Development Team members may have specialized skills and areas of focus, but accountability belongs to the Development Team as a whole.

C. Development Team Size

Optimal Development Team size is small enough to remain nimble and large enough to complete significant work within a Sprint. Fewer than three Development Team members decrease interaction and results in smaller productivity gains. Smaller Development Teams may encounter skill constraints during the Sprint, causing the Development Team to be unable to deliver a potentially releasable Increment. Having more than nine members requires too much coordination. Large Development Teams generate too much complexity for an empirical process to be useful. The Product Owner and Scrum Master roles are not included in this count unless they are also executing the work of the Sprint Backlog.

D. The Scrum Master

The Scrum Master is responsible for promoting and supporting Scrum as defined in the Scrum Guide. Scrum Masters do this by helping everyone understand Scrum theory, practices, rules, and values.

The Scrum Master is a servant-leader for the Scrum Team. The Scrum Master helps those outside the Scrum Team understand which of their interactions with the Scrum Team are helpful and which aren't. The Scrum Master helps everyone change these interactions to maximize the value created by the Scrum Team.

- **Scrum Master Service to the Product Owner**

The Scrum Master serves the Product Owner in several ways, including:

- Ensuring that goals, scope, and product domain are understood by everyone on the Scrum Team as well as possible;
- Finding techniques for effective Product Backlog management;
- Helping the Scrum Team understand the need for clear and concise Product Backlog items;
- Understanding product planning in an empirical environment;
- Ensuring the Product Owner knows how to arrange the Product Backlog to maximize value;
- Understanding and practicing agility; and,
- Facilitating Scrum events as requested or needed.

- **Scrum Master Service to the Development Team**

The Scrum Master serves the Development Team in several ways, including:

- Coaching the Development Team in self-organization and cross-functionality;
- Helping the Development Team to create high-value products;
- Removing impediments to the Development Team's progress;
- Facilitating Scrum events as requested or needed; and,
- Coaching the Development Team in organizational environments in which Scrum is not yet fully adopted and understood.

- **Scrum Master Service to the Organization**

The Scrum Master serves the organization in several ways, including:

- Leading and coaching the organization in its Scrum adoption;
- Planning Scrum implementations within the organization;
- Helping employees and stakeholders understand and enact Scrum and empirical product development;
- Causing change that increases the productivity of the Scrum Team; and,
- Working with other Scrum Masters to increase the effectiveness of the application of Scrum in the organization.

5. Scrum Events

Prescribed events are used in Scrum to create regularity and to minimize the need for meetings not defined in Scrum. All events are time-boxed events, such that every event has a maximum duration. Once a Sprint begins, its duration is fixed and cannot be shortened or lengthened. The remaining events may end whenever the purpose of the event is achieved, ensuring an appropriate amount of time is spent without allowing waste in the process.

Other than the Sprint itself, which is a container for all other events, each event in Scrum is a formal opportunity to inspect and adapt something. These events are specifically designed to enable critical transparency and inspection. Failure to include any of these events results in reduced transparency and is a lost opportunity to inspect and adapt.

A. The Sprint

The heart of Scrum is a Sprint, a time-box of one month or less during which a "Done", useable, and potentially releasable product Increment is created. Sprints have consistent durations throughout a development effort. A new Sprint starts immediately after the conclusion of the previous Sprint.

Sprints contain and consist of the Sprint Planning, Daily Scrums, the development work, the Sprint Review, and the Sprint Retrospective.

During the Sprint:

- No changes are made that would endanger the Sprint Goal;
- Quality goals do not decrease; and,
- Scope may be clarified and re-negotiated between the Product Owner and Development Team as more is learned.

Each Sprint may be considered a project with no more than a one-month horizon. Like projects, Sprints are used to accomplish something. Each Sprint has a goal of what is to be built, a design and flexible plan that will guide building it, the work, and the resultant product increment.

Sprints are limited to one calendar month. When a Sprint's horizon is too long the definition of what is being built may change, complexity may rise, and risk may increase. Sprints enable predictability by ensuring inspection and adaptation of progress toward a Sprint Goal at least every calendar month. Sprints also limit risk to one calendar month of cost.

- **Cancelling a Sprint**

A Sprint can be cancelled before the Sprint time-box is over. Only the Product Owner has the authority to cancel the Sprint, although he or she may do so under influence from the stakeholders, the Development Team, or the Scrum Master.

A Sprint would be cancelled if the Sprint Goal becomes obsolete. This might occur if the company changes direction or if market or technology conditions change. In general, a Sprint should be cancelled if it no longer makes sense given the circumstances. But, due to the short duration of Sprints, cancellation rarely makes sense.

When a Sprint is cancelled, any completed and "Done" Product Backlog items are reviewed. If part of the work is potentially releasable, the Product Owner typically accepts it. All incomplete Product Backlog Items are re-estimated and put back on the Product Backlog. The work done on them depreciates quickly and must be frequently re-estimated.

Sprint cancellations consume resources, since everyone regroups in another Sprint Planning to start another Sprint. Sprint cancellations are often traumatic to the Scrum Team, and are very uncommon.

B. Sprint Planning

The work to be performed in the Sprint is planned at the Sprint Planning. This plan is created by the collaborative work of the entire Scrum Team.

Sprint Planning is time-boxed to a maximum of eight hours for a one-month Sprint. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendants understand its purpose. The Scrum Master teaches the Scrum Team to keep it within the time-box.

Sprint Planning answers the following:

- What can be delivered in the Increment resulting from the upcoming Sprint?
- How will the work needed to deliver the Increment be achieved?

- **Topic One: What can be done this Sprint?**

The Development Team works to forecast the functionality that will be developed during the Sprint. The Product Owner discusses the objective that the Sprint should achieve and the Product Backlog items that, if completed in the Sprint, would achieve the Sprint Goal. The entire Scrum Team collaborates on understanding the work of the Sprint.

The input to this meeting is the Product Backlog, the latest product Increment, projected capacity of the Development Team during the Sprint, and past performance of the Development Team. The number of items selected from the Product Backlog for the Sprint is solely up to the Development Team. Only the Development Team can assess what it can accomplish over the upcoming Sprint.

During Sprint Planning the Scrum Team also crafts a Sprint Goal. The Sprint Goal is an objective that will be met within the Sprint through the implementation of the Product Backlog, and it provides guidance to the Development Team on why it is building the Increment.

- **Topic Two: how will the chosen work get done?**

Having set the Sprint Goal and selected the Product Backlog items for the Sprint, the Development Team decides how it will build this functionality into a "Done" product Increment during the Sprint. The Product Backlog items selected for this Sprint plus the plan for delivering them is called the Sprint Backlog.

The Development Team usually starts by designing the system and the work needed to convert the Product Backlog into a working product Increment. Work may be of varying size, or estimated effort. However, enough work is planned during Sprint Planning for the Development Team to forecast what it believes it can do in the upcoming Sprint. Work planned for the first days of the Sprint by the Development Team is decomposed by the end of this meeting, often to units of

one day or less. The Development Team self-organizes to undertake the work in the Sprint Backlog, both during Sprint Planning and as needed throughout the Sprint.

The Product Owner can help to clarify the selected Product Backlog items and make trade-offs. If the Development Team determines it has too much or too little work, it may renegotiate the selected Product Backlog items with the Product Owner. The Development Team may also invite other people to attend to provide technical or domain advice.

By the end of the Sprint Planning, the Development Team should be able to explain to the Product Owner and Scrum Master how it intends to work as a self-organizing team to accomplish the Sprint Goal and create the anticipated Increment.

C. Sprint Goal

The Sprint Goal is an objective set for the Sprint that can be met through the implementation of Product Backlog. It provides guidance to the Development Team on why it is building the Increment. It is created during the Sprint Planning meeting. The Sprint Goal gives the Development Team some flexibility regarding the functionality implemented within the Sprint. The selected Product Backlog items deliver one coherent function, which can be the Sprint Goal. The Sprint Goal can be any other coherence that causes the Development Team to work together rather than on separate initiatives.

As the Development Team works, it keeps the Sprint Goal in mind. In order to satisfy the Sprint Goal, it implements functionality and technology. If the work turns out to be different than the Development Team expected, they collaborate with the Product Owner to negotiate the scope of Sprint Backlog within the Sprint.

D. Daily Scrum

The Daily Scrum is a 15-minute time-boxed event for the Development Team. The Daily Scrum is held every day of the Sprint. At it, the Development Team plans work for the next 24 hours. This optimizes team collaboration and performance by inspecting the work since the last Daily Scrum and forecasting upcoming Sprint work. The Daily Scrum is held at the same time and place each day to reduce complexity.

The Development Team uses the Daily Scrum to inspect progress toward the Sprint Goal and to inspect how progress is trending toward completing the work in the Sprint Backlog. The Daily Scrum optimizes the probability that the Development Team will meet the Sprint Goal. Every day, the Development Team should understand how it intends to work together as a self-organizing team to accomplish the Sprint Goal and create the anticipated Increment by the end of the Sprint.

The structure of the meeting is set by the Development Team and can be conducted in different ways if it focuses on progress toward the Sprint Goal. Some Development Teams will use

questions, some will be more discussion based. Here is an example of what might be used:

- What did I do yesterday that helped the Development Team meet the Sprint Goal?
- What will I do today to help the Development Team meet the Sprint Goal?
- Do I see any impediment that prevents me or the Development Team from meeting the Sprint Goal?

The Development Team or team members often meet immediately after the Daily Scrum for detailed discussions, or to adapt, or replan, the rest of the Sprint's work.

The Scrum Master ensures that the Development Team has the meeting, but the Development Team is responsible for conducting the Daily Scrum. The Scrum Master teaches the Development Team to keep the Daily Scrum within the 15-minute time-box.

The Daily Scrum is an internal meeting for the Development Team. If others are present, the Scrum Master ensures that they do not disrupt the meeting.

Daily Scrums improve communications, eliminate other meetings, identify impediments to development for removal, highlight and promote quick decision-making, and improve the Development Team's level of knowledge. This is a key inspect and adapt meeting.

E. Sprint Review

A Sprint Review is held at the end of the Sprint to inspect the Increment and adapt the Product Backlog if needed. During the Sprint Review, the Scrum Team and stakeholders collaborate about what was done in the Sprint. Based on that and any changes to the Product Backlog during the Sprint, attendees collaborate on the next things that could be done to optimize value. This is an informal meeting, not a status meeting, and the presentation of the Increment is intended to elicit feedback and foster collaboration.

This is at most a four-hour meeting for one-month Sprints. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendees understand its purpose. The Scrum Master teaches everyone involved to keep it within the time-box.

The Sprint Review includes the following elements:

- Attendees include the Scrum Team and key stakeholders invited by the Product Owner;
- The Product Owner explains what Product Backlog items have been "Done" and what has not been "Done";
- The Development Team discusses what went well during the Sprint, what problems it ran into, and how those problems were solved;
- The Development Team demonstrates the work that it has "Done" and answers questions

about the Increment;

- The Product Owner discusses the Product Backlog as it stands. He or she projects likely target and delivery dates based on progress to date (if needed);
- The entire group collaborates on what to do next, so that the Sprint Review provides valuable input to subsequent Sprint Planning;
- Review of how the marketplace or potential use of the product might have changed what is the most valuable thing to do next; and,
- Review of the timeline, budget, potential capabilities, and marketplace for the next anticipated releases of functionality or capability of the product.

The result of the Sprint Review is a revised Product Backlog that defines the probable Product Backlog items for the next Sprint. The Product Backlog may also be adjusted overall to meet new opportunities.

F. Sprint Retrospective

The Sprint Retrospective is an opportunity for the Scrum Team to inspect itself and create a plan for improvements to be enacted during the next Sprint.

The Sprint Retrospective occurs after the Sprint Review and prior to the next Sprint Planning. This is at most a three-hour meeting for one-month Sprints. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendants understand its purpose.

The Scrum Master ensures that the meeting is positive and productive. The Scrum Master teaches all to keep it within the time-box. The Scrum Master participates as a peer team member in the meeting from the accountability over the Scrum process.

The purpose of the Sprint Retrospective is to:

- Inspect how the last Sprint went with regards to people, relationships, process, and tools;
- Identify and order the major items that went well and potential improvements; and,
- Create a plan for implementing improvements to the way the Scrum Team does its work.

The Scrum Master encourages the Scrum Team to improve, within the Scrum process framework, its development process and practices to make it more effective and enjoyable for the next Sprint. During each Sprint Retrospective, the Scrum Team plans ways to increase product quality by improving work processes or adapting the definition of "Done", if appropriate and not in conflict with product or organizational standards.

By the end of the Sprint Retrospective, the Scrum Team should have identified improvements that it will implement in the next Sprint. Implementing these improvements in the next Sprint is the

adaptation to the inspection of the Scrum Team itself. Although improvements may be implemented at any time, the Sprint Retrospective provides a formal opportunity to focus on inspection and adaptation.

6. Scrum Artifacts

Scrum's artifacts represent work or value to provide transparency and opportunities for inspection and adaptation. Artifacts defined by Scrum are specifically designed to maximize transparency of key information so that everybody has the same understanding of the artifact.

A. Product Backlog

The Product Backlog is an ordered list of everything that is known to be needed in the product. It is the single source of requirements for any changes to be made to the product. The Product Owner is responsible for the Product Backlog, including its content, availability, and ordering.

A Product Backlog is never complete. The earliest development of it lays out the initially known and best-understood requirements. The Product Backlog evolves as the product and the environment in which it will be used evolves. The Product Backlog is dynamic; it constantly changes to identify what the product needs to be appropriate, competitive, and useful. If a product exists, its Product Backlog also exists.

The Product Backlog lists all features, functions, requirements, enhancements, and fixes that constitute the changes to be made to the product in future releases. Product Backlog items have the attributes of a description, order, estimate, and value. Product Backlog items often include test descriptions that will prove its completeness when "Done".

As a product is used and gains value, and the marketplace provides feedback, the Product Backlog becomes a larger and more exhaustive list. Requirements never stop changing, so a Product Backlog is a living artifact. Changes in business requirements, market conditions, or technology may cause changes in the Product Backlog.

Multiple Scrum Teams often work together on the same product. One Product Backlog is used to describe the upcoming work on the product. A Product Backlog attribute that groups items may then be employed.

Product Backlog refinement is the act of adding detail, estimates, and order to items in the Product Backlog. This is an ongoing process in which the Product Owner and the Development Team collaborate on the details of Product Backlog items. During Product Backlog refinement, items are reviewed and revised. The Scrum Team decides how and when refinement is done. Refinement usually consumes no more than 10% of the capacity of the Development Team. However, Product

Backlog items can be updated at any time by the Product Owner or at the Product Owner's discretion.

Higher ordered Product Backlog items are usually clearer and more detailed than lower ordered ones. More precise estimates are made based on the greater clarity and increased detail; the lower the order, the less detail. Product Backlog items that will occupy the Development Team for the upcoming Sprint are refined so that any one item can reasonably be "Done" within the Sprint time-box. Product Backlog items that can be "Done" by the Development Team within one Sprint are deemed "Ready" for selection in a Sprint Planning. Product Backlog items usually acquire this degree of transparency through the above described refining activities.

The Development Team is responsible for all estimates. The Product Owner may influence the Development Team by helping it understand and select trade-offs, but the people who will perform the work make the final estimate.

- **Monitoring Progress Toward Goals**

At any point in time, the total work remaining to reach a goal can be summed. The Product Owner tracks this total work remaining at least every Sprint Review. The Product Owner compares this amount with work remaining at previous Sprint Reviews to assess progress toward completing projected work by the desired time for the goal. This information is made transparent to all stakeholders.

Various projective practices upon trending have been used to forecast progress, like burn-downs, burn-ups, or cumulative flows. These have proven useful. However, these do not replace the importance of empiricism. In complex environments, what will happen is unknown. Only what has already happened may be used for forward-looking decision-making.

B. Sprint Backlog

The Sprint Backlog is the set of Product Backlog items selected for the Sprint, plus a plan for delivering the product Increment and realizing the Sprint Goal. The Sprint Backlog is a forecast by the Development Team about what functionality will be in the next Increment and the work needed to deliver that functionality into a "Done" Increment.

The Sprint Backlog makes visible all the work that the Development Team identifies as necessary to meet the Sprint Goal. To ensure continuous improvement, it includes at least one high priority process improvement identified in the previous Retrospective meeting.

The Sprint Backlog is a plan with enough detail that changes in progress can be understood in the Daily Scrum. The Development Team modifies the Sprint Backlog throughout the Sprint, and the Sprint Backlog emerges during the Sprint. This emergence occurs as the Development Team works through the plan and learns more about the work needed to achieve the Sprint Goal.

As new work is required, the Development Team adds it to the Sprint Backlog. As work is performed or completed, the estimated remaining work is updated. When elements of the plan are deemed unnecessary, they are removed. Only the Development Team can change its Sprint Backlog during a Sprint. The Sprint Backlog is a highly visible, real-time picture of the work that the Development Team plans to accomplish during the Sprint, and it belongs solely to the Development Team.

- **Monitoring Sprint Progress**

At any point in time in a Sprint, the total work remaining in the Sprint Backlog can be summed. The Development Team tracks this total work remaining at least for every Daily Scrum to project the likelihood of achieving the Sprint Goal. By tracking the remaining work throughout the Sprint, the Development Team can manage its progress.

C. Increment

The Increment is the sum of all the Product Backlog items completed during a Sprint and the value of the increments of all previous Sprints. At the end of a Sprint, the new Increment must be "Done," which means it must be in useable condition and meet the Scrum Team's definition of "Done". An increment is a body of inspectable, done work that supports empiricism at the end of the Sprint. The increment is a step toward a vision or goal. The increment must be in useable condition regardless of whether the Product Owner decides to release it.

7. Artifact Transparency

Scrum relies on transparency. Decisions to optimize value and control risk are made based on the perceived state of the artifacts. To the extent that transparency is complete, these decisions have a sound basis. To the extent that the artifacts are incompletely transparent, these decisions can be flawed, value may diminish and risk may increase.

The Scrum Master must work with the Product Owner, Development Team, and other involved parties to understand if the artifacts are completely transparent. There are practices for coping with incomplete transparency; the Scrum Master must help everyone apply the most appropriate practices in the absence of complete transparency. A Scrum Master can detect incomplete transparency by inspecting the artifacts, sensing patterns, listening closely to what is being said, and detecting differences between expected and real results.

The Scrum Master's job is to work with the Scrum Team and the organization to increase the transparency of the artifacts. This work usually involves learning, convincing, and change. Transparency doesn't occur overnight, but is a path.

D. Definition of "Done"

When a Product Backlog item or an Increment is described as "Done", everyone must understand what "Done" means. Although this may vary significantly per Scrum Team, members

must have a shared understanding of what it means for work to be complete, to ensure transparency. This is the definition of "Done" for the Scrum Team and is used to assess when work is complete on the product Increment.

The same definition guides the Development Team in knowing how many Product Backlog items it can select during a Sprint Planning. The purpose of each Sprint is to deliver Increments of potentially releasable functionality that adhere to the Scrum Team's current definition of "Done".

Development Teams deliver an Increment of product functionality every Sprint. This Increment is useable, so a Product Owner may choose to immediately release it. If the definition of "Done" for an increment *is* part of the conventions, standards or guidelines of the development organization, all Scrum Teams must follow it as a minimum.

If "Done" for an increment is *not* a convention of the development organization, the Development Team of the Scrum Team must define a definition of "Done" appropriate for the product. If there are multiple Scrum Teams working on the system or product release, the Development Teams on all the Scrum Teams must mutually define the definition of "Done".

Each Increment is additive to all prior Increments and thoroughly tested, ensuring that all Increments work together.

As Scrum Teams mature, it is expected that their definitions of "Done" will expand to include more stringent criteria for higher quality. New definitions, as used, may uncover work to be done in previously "Done" increments. Any one product or system should have a definition of "Done" that is a standard for any work done on it.

II. Appendix 2: software quality dimensions

1. Reliability

The root causes of poor reliability are found in a combination of non-compliance with good architectural and coding practices. This non-compliance can be detected by measuring the static quality attributes of an application which provides an estimate of the level of business risk and the likelihood of potential application failures and defects the application will experience when placed in operation [164,165].

Assessing reliability requires checks of at least the following software engineering best practices and technical attributes [164,166,167]:

- Application Architecture Practices.
- Coding Practices.
- Complexity of algorithms.

- The complexity of programming practices.
- Compliance with Object–Oriented and Structured Programming best practices (when applicable).
- Component or pattern reuse ratio.
- Dirty programming.
- Error & Exception handling (for all layers – GUI, Logic & Data).
- Multi–layer design compliance.
- Resource bounds management.
- Software avoids patterns that will lead to unexpected behaviors.
- The software manages data integrity and consistency.
- Transaction complexity level.

2. Efficiency

As with Reliability, the causes of performance inefficiency are often found in violations of good architectural and coding practices, which can be detected by measuring an application's static quality attributes which predict potential operational performance bottlenecks and future scalability problems, especially for applications requiring high execution speed for handling complex algorithms or huge volumes of data [164,165].

Assessing performance efficiency requires checking at least the following software engineering best practices and technical attributes [164,167]:

- Application Architecture Practices.
- Appropriate interactions with expensive and/or remote resources.
- Data access performance and data management.
- Memory, network, and disk space management.
- Coding Practices.
- Compliance with Object–Oriented and Structured Programming best practices (as appropriate).
- Compliance with SQL programming best practices.

3. Security

Most security vulnerabilities result from poor coding and architectural practices such as SQL injection or cross–site scripting [164,167].

Assessing security requires at least checking the following software engineering best practices and technical attributes [164,167]:

- Application Architecture Practices.
- Multi–layer design compliance.

- Security best practices (Input Validation, SQL Injection, Cross-Site Scripting, etc.).
- Programming Practices (code level).
- Error & Exception handling.
- Security best practices (system functions access, access control to programs).

4. Maintainability

Maintainability includes concepts of modularity, understandability, changeability, testability, reusability, and transferability from one development team to another [164,165,167].

Assessing maintainability requires checking the following software engineering best practices and technical attributes [164,167,167]:

- Application Architecture Practices.
- Architecture, Programs and Code documentation embedded in source code.
- Code readability.
- The complexity level of transactions.
- Complexity of algorithms.
- The complexity of programming practices.
- Compliance with Object-Oriented and Structured Programming best practices (when applicable).
- Component or pattern reuse ratio.
- Controlled level of dynamic coding.
- Coupling ratio.
- Dirty programming.
- Documentation.
- Hardware, OS, middleware, software components, and database independence.
- Multi-layer design compliance.
- Portability.
- Programming Practices (code level).
- Reduced duplicate code and functions.
- Source code file organization cleanliness.

III. Appendix 3: security considerations and measures

1. Security by design

Security by design, or alternately secure by design, means that the software has been designed from the ground up to be secure. In this case, security is considered as the main feature [168,169,170].

Some of the techniques in this approach include [168,169]:

- The least privilege principle, where each part of the system has only the permissions for its function. Even if an intruder accesses that component, they only have restricted access to the system.
- Automated theorem proving the correctness of software subsystems.
- Code reviews and unit testing methods to make modules safer where evidence of formality is not feasible.
- Defense-in-depth, where the design is such that more than one subsystem needs to be violated to compromise the system's integrity and the information it holds.
- Default secure settings, and design to "fail secure" rather than "fail insecure". Ideally, a secure system should require a deliberate, conscious, knowledgeable, and free decision on legitimate authorities to make it insecure.

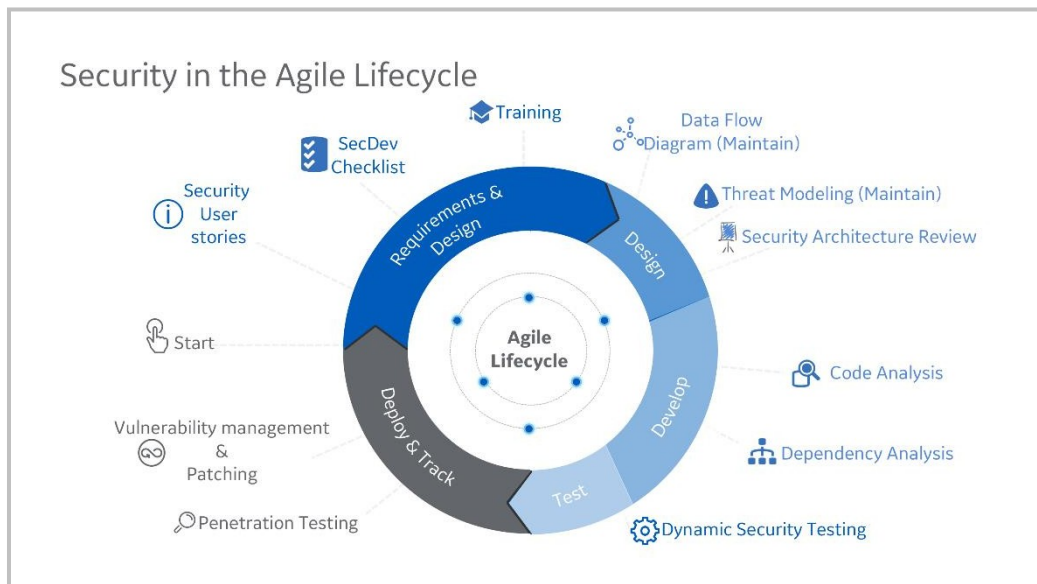


Figure 190: security by design applied to the Agile methodology [171]

2. Security architecture

Security architecture is the design specifying how security controls / countermeasures are positioned and how they contribute to the overall IT architecture. These controls help to protect the quality attributes of the system: confidentiality, integrity, availability, transparency and insurance services [170].

The key attributes of security architecture are [172]:

- Different component relationships and how they rely on each other.
- Determining controls based on risk management, good practice, finance, and regulatory matters.
- The standardization of controls.

Practicing security architecture offers the best basis to consistently resolve business, IT, and organizational security concerns.

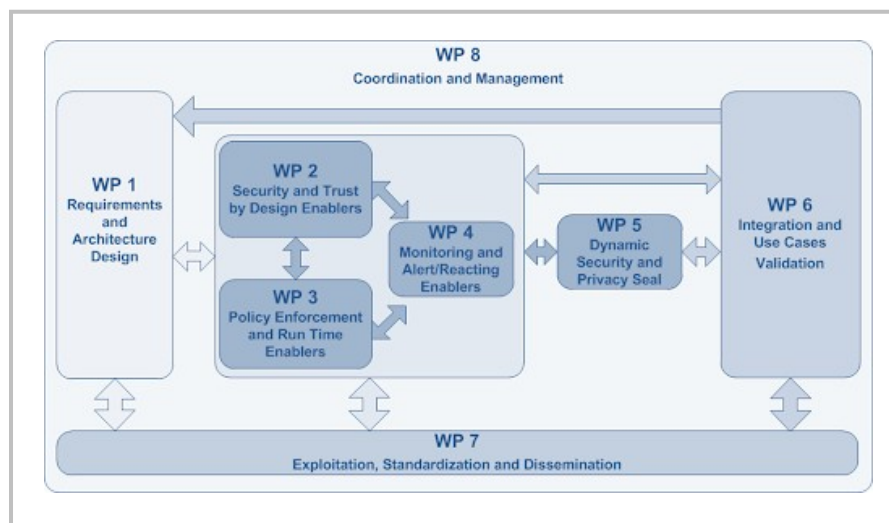


Figure 191: security architecture [172]

3. Security measures

A software "secure" state is the conceptual ideal achieved using three processes: threat prevention, detection, and response. These processes are based on various policies and system components [173,174]:

- Access control systems and cryptography can shield device files and data, respectively.
- Firewalls are the most popular network security protection mechanisms. They can (if properly configured) shield access to internal network resources and block such attacks by filtering packets. Firewalls may be hardware- or software-based. Firewalls are the most popular network security protection mechanisms. They can (if properly configured) shield access to internal network resources and block such attacks by filtering packets. Firewalls may be hardware or software-based.

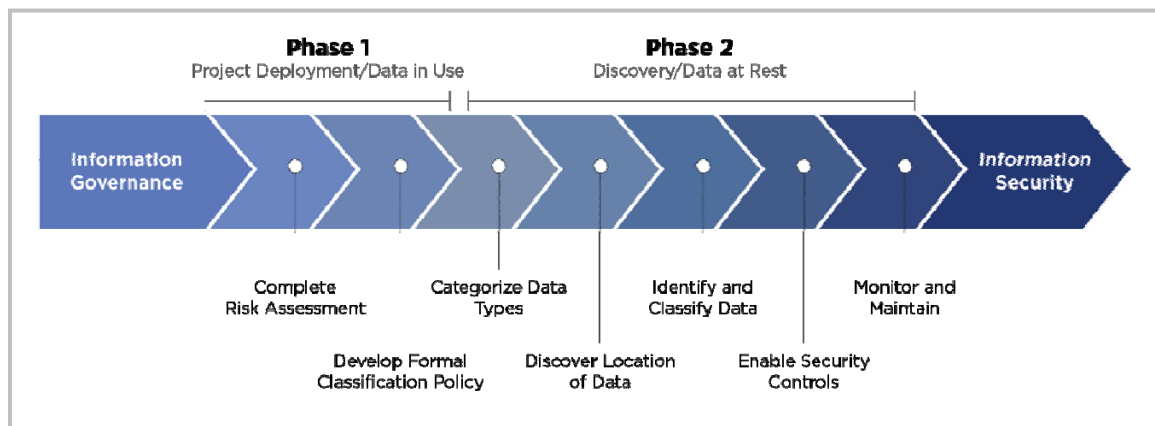


Figure 192: security measures [172]

4. Identity management

Identity management (ID management) is the organizational mechanism for identifying, authenticating, and authorizing individuals or groups of people to access applications, systems, or networks by integrating user rights and restrictions with existing identities. It also includes how and by whom descriptive details about the user can be accessed and updated. Identity management (ID management) is the organizational mechanism for identifying, authenticating, and authorizing individuals or groups of people to access applications, systems, or networks by integrating user rights and restrictions with existing identities. It also includes how and by whom descriptive details about the user can be accessed and updated [175,176].



Figure 193: identity management [175]

5. Access control

Access control (AC) is the selective restriction of access to a place or other resource while access management describes the process. Accessing may mean consuming, entering, or using. Permission to access a resource is called authorization [114].

Access control requires authentication, authorization, auditing. A narrower definition of access control will only cover access approval, whereby the device decides to approve or deny a request for access from an already authenticated subject depending on what the subject can access. Authentication and access control are also combined into a single operation to be accepted based on successful authentication or an anonymous access token Access control requires authentication, authorization, auditing. A narrower definition of access control will only cover access approval, whereby the device decides to approve or deny a request for access from an already authenticated subject depending on what the subject can access. Authentication and access control are also combined into a single operation to be accepted on the basis of successful authentication or an anonymous access token [114].

Determining what to access, how to access, and how grants are shaped, distributed and assigned is governed by access control policies; Many regulating policies exist, each with its own pros and cons and spectrum of use [114].

IV. Appendix 4: legislative considerations

The medical record was over time subject to multiple legal and ethical obligations and recommendations, whether during its preparation or its management.

This chapter illustrates the importance of regulations in the existence and management of the medical record, but also the importance of the medical record in the legal field, namely in the protection of the rights of patients, doctors and healthcare establishments.

1. Medical record

The medical record in hospitals, an informal tool kept for a long time by doctors, has become a formal obligation in many Western countries in application of legislative and regulatory texts.

In Morocco, the medical record is an ethical obligation cited in Articles 22, 24 and 60 of the Code of Physicians and in Article 44 of the Code of Dental Surgeons. It became legal with the promulgation of Law 65-00 on compulsory health insurance [177,178,179]. However, no law specifies the modalities of its holding, nor the sanctions applied in the event of derogation.

The internal regulations of hospitals cite this in Articles 137 and 140 and make the hospital the owner of the medical record and responsible for its preservation during hospitalization and after discharge. It organizes the methods of communication of information and assigns responsibility for safeguarding the confidential nature of record information to the hospital administration [180].

In France, from 1970, the law of December 31 and its implementing decree of 1974 made the medical record a legal obligation in public health establishments. This obligation was generalized in 1991 to all health establishments and even for liberal medicine in 1995. Thus any defect in its keeping or in its updating is sanctioned by law [18,181].

The patient's record was the subject of a finalization in June 2003 by the National Agency for Accreditation and Evaluation in Health in France (ANAES), which has published recommendations aimed at improving the quality of medical care. It considers the medical record as the main element among ten accreditation standards [32].

In the United States, the "Joint Commission on Accreditation of Healthcare Organization" (JCAHO) has particularly studied the criteria for evaluating the keeping of medical records (and their purposes), and has detailed and clarified them in the 2 reference books that are the "Comprehensive Accreditation Manual for Hospitals: The Official Handbook" and the "Hospital Accreditation Standards" [18].

A. Medical record content

- **Morocco**

The Ministry of Health in a perspective of improving care, and in collaboration with the public assistance of Paris hospitals, introduced in 1995, a hospitalization record organized in a single medium for medical, nursing and administrative information. This intervention, envisaged as a pilot project, did not materialize as the initiators of the project wanted. With the commitment of the Ministry of Health in hospital reform, a process has been initiated to revise the hospital information system in order to provide the public hospital with modern structures and management methods. The product of this revision is the normative framework of the Hospital Information and Management System (SIG-Ho) in general and of the patient record in particular as the hub of the entire hospital information system implemented by the Ministry of Health in 2001 [182].

In fact, this approach has tried to develop collection that supports hospital information which are 71 in number (forms and summary statements and reports, vouchers, invoices and certificates, tickets, bulletins and notices, dashboards, etc.), the main one being the hospital medical record:

- Patient identification card
 - o Administrative and social information
 - o Residence number
- Admission form: full hospitalization or day hospitalization
 - o History, reason for going to hospital
 - o Medical history, current treatments
 - o Data from the initial clinical examination, conclusions of the first examination and subsequent examination
 - o Results and reports of para-clinical examinations
 - o Pre-anesthetic consultation sheet
 - o Per and post-operative monitoring sheet
 - o Operative or delivery report
 - o Medical observation of the newborn
 - o Medical prescriptions
 - o Transfusion monitoring card
 - o Hospitalization report (with diagnosis, clinical summary, exit procedures and exit requirements)
- Nursing records
 - o Objective and protocols of care, planning of nursing care
 - o Acts and actions implemented (education and prevention), evaluation of care in relation to defined objectives
 - o Summary sheet of the nursing record

The Hospitals and Ambulatory Care Directorate carried out an evaluation of this system one year after its introduction which revealed that only 53% of hospitals implemented the record and 25% of these record systems were completed.

- **France**

In France, the contents of the medical record are more detailed, since the subject of numerous obligations and legislative clarifications and recommendations of health authorities represented by the National Agency for Accreditation and Health Evaluation (ANAES) whose functions are taken over by the High Authority for Health (HAS).

Administrative record

The hospital administration must take steps to ensure that the administrative record, drawn up during contact with the patient, is distinct from the medical file and must not contain any data of a medical nature. It feeds the patient's file with all the elements allowing to identify the patient, his administrative situation and his social cover.

It includes for each hospitalized patient, the name and address of the patient, his registration number, his affiliation fund, the date and time of his entry, the admission discipline, the notion of possible transfer, the date and time of exit. A return home aptitude form is required for patients treated in an alternative structure [183]. It also includes, if necessary, the various authorizations required by the regulations, in particular [19]:

- Authorizations to operate on a minor patient.
- Refusal to authorize an autopsy or organ removal from a deceased person.
- Discharges for discharge against medical advice.
- The findings of fugue.
- The identity of the trusted person and that of the person to be notified who will be consulted in the event that they are unable to express their wishes and receive the information necessary for this purpose. This designation is made in writing.

The administration must keep the register of patient arrivals and departures indefinitely, as well as an emergency register.

ANAES recommends that this identification be reliable and collected with the greatest possible precision from official administrative documents presented by the patient, such as the identity card, passport or residence permit. It also recommends respecting the rules for entering the identity of patients who are the subject of a procedure disseminated to the persons concerned and regular assessment [3].

Standardized, high-quality administrative identification avoids duplication resulting from new information gathering that could generate errors. It allows the production of labels that can be

used for requests for additional examinations, small bar codes, they serve as a means which helps in the archiving of files [3].

Patient record

Article No. R. 1112-2 of Decree No. 2003-462 of May 21, 2003 of the French Public Health Code specifies and classifies the minimum content of the patient's record [19]:

- Formalized information collected during outpatient consultations provided in the establishment, during reception in the emergency department or at the time of admission and during the stay in a health establishment, and in particular:
 - o The letter from the doctor who initiated the consultation or admission;
 - o The reasons for hospitalization;
 - o The search for antecedents and risk factors;
 - o The conclusions of the initial clinical evaluation;
 - o The type of care provided and the prescriptions made on entry;
 - o The nature of the care provided and the prescriptions established during the outpatient visit or the emergency room;
 - o Information relating to care during hospitalization: clinical status, care received, paraclinical examinations, in particular imaging;
 - o Information on the medical approach adopted under the conditions provided for in Article L. 1111-4;
 - o The anesthesia file;
 - o The operating or delivery report;
 - o The written consent of the patient for situations where this consent is required in this form by legal or regulatory means;
 - o Mention of the transfusion acts performed on the patient and, where applicable, a copy of the transfusion incident sheet mentioned in the second paragraph of article R. 1221-40;
 - o Items relating to the medical prescription, its execution and additional examinations;
 - o The nursing care file or, failing that, information relating to nursing care;
 - o Information relating to the care provided by other health professionals;
 - o Correspondence exchanged between health professionals.
- The formalized information established at the end of the stay. They include in particular:
 - o The hospitalization report and the letter written on the occasion of discharge;
 - o The discharge prescription and duplicate discharge orders;
 - o The exit procedures (home, other structures);
 - o The nurse liaison form.
- Information stating that they were collected from third parties not involved in the therapeutic care or concerning such third parties.

Only the information listed in 1 ° and 2 ° can be communicated. " [184]; This list is not exhaustive, as the file may also include a number of other documents:

- Copies of medical certificates;
- The autopsy reports;
- Documents allowing the traceability of the actions and information of the patient concerning numerous vigilances: haemovigilance, biovigilance, nosocomiovigilance, pharmacovigilance, etc [19].

There can be no distinction between the hospitalization and consultation records. The information resulting from the consultation is an integral part of the patient's record and cannot be separated from the hospitalization file. French law requires that each part of the record include the identification of the patient, and each writing must include the date and the identity of the professional who carried it out. Medical prescriptions must be time stamped and signed by prescribing physician and include the legible name of the physician [19].

ANAES recommends that the patient's record include the trace of the benefit–risk reflection, the diagnostic and therapeutic strategy adopted for the patient before each invasive procedure, and updated information on the evolution of his clinical condition and its intake. in charge. As well as, after his release, the conclusions of the stay and any follow-up procedures. In the event of the patient's death, it is recommended to note in the file the circumstances and causes of death. ANAES paid particular attention to the hospitalization report. And recommends the following structuring [19]:

- Patient identification;
- Identification of the dates of contact (date of consultation, dates of entry and exit from hospitalization);
- Place of contact;
- Entry mode;
- Reason for contact;
- Patient history;
- Lifestyle;
- History of the disease ;
- Physical examination ;
- Significant biological results;
- Results of additional examinations;
- Treatments performed, including transfusions;
- Evolution in the service and discussion;
- Mode of discharge (patient's destination), including the date and time of discharge, the means of transport and any accompaniment, as well as the list of items given to the patient;
- Exit processing;

- Action to be taken (monitoring to be instituted, reconconvocation, etc.);
- Conclusion in summary form.

Paramedical record

It is an essential component of the patient record of which it is an integral part. It includes the nursing care record or, failing that, information relating to nursing care and information relating to care provided by other health professionals possibly organized in "sub-records" [19].

French legislative texts have assigned the nurse responsibility for the constitution and management of this case [16]. These texts specify that the nursing care file includes all the elements relating to nursing care in particular, the monitoring parameters (temperature, pulsations, blood pressure, respiratory rate, volume of diuresis, weight, measurements, pupillary reflexes, skin defense reflexes, observations of manifestations of the state of consciousness, evaluation of pain) and the trace of all the acts of care in application of medical prescriptions (injections, infusions, treatment of pain, etc.). Caregivers and auxiliaries must record their observations and actions in this record. Each paramedical professional can create a sub-sheet specific to his profession which will be an integral part of the patient's record [185].

ANAES has published clinical practice guidelines for the paramedical care record. She encourages the nurse after each intervention to note the date, nature and results of the nursing care provided as well as his observations on the evolution of the patient's condition and any information collected that may be useful for his care. overall. It requires that all medical prescriptions be noted, dated and signed by prescribing physicians in the nursing record. It establishes in writing a summary of the nursing care of the patient upon discharge [19].

B. Data privacy protection

09-08 Regulation which consists of five chapters:

- The first chapter describes the creation of a national commission based on the security of personal data; it highlights its area of interest, its extent of intervention, and how it interacts with entities dealing with personal data, primarily through a data governance officer charged with managing data access to personal data, as well as how these entities will be responsabilized towards the protection of personal data [73].
- The second chapter outlines national commission competences and authorization procedure [73].
- The third chapter is about certain specific data treatment categories, mainly statistical and historical data treatment, and about ensuring data anonymity if the result is going to be publicly published or at least justify the reason behind the divulgation of identifying personal data [73].
- The fourth chapter elaborates on the rights of persons of interest regarding their data, mainly the right to access, rectify, and oppose [73].

- The fifth chapter is about the transfer of personal data to foreign countries, which cannot be done without the authorization of the national committee [73].

Related regulations are mainly decrees which promulgate the application of 09–08 law.

The 24th article of the Moroccan constitution elaborates on the right to protect private life [72].

The Health Insurance Portability and Accountability Act of 1996 (HIPAA or the Kennedy-Kassebaum Act) was enacted by the 104th United States Congress and signed by President Bill Clinton in 1996. It was created primarily to modernize the flow of healthcare information, stipulate how Personally Identifiable Information maintained by the healthcare and healthcare insurance industries should be protected from fraud and theft, and address limitations on healthcare insurance coverage [74,75].

The 27th article of the Moroccan constitution elaborates the right to information access, which cannot be limited only by the law, to ensure the protection of everything that concerns national defense, the internal and external security of the state, as well as the personal life of the individual [72].

C. Access and communication

The right to information is a universal principle recognized by all regulations around the world. In addition, the communication of the elements of the medical file is a main element ensuring the continuity of care, main role of the patient file. However, access to and communication of medical information are subject to various legal and ethical rules all aimed at protecting professional secrecy. This covers everything that has come to the knowledge of the doctor in the exercise of his profession, that is to say not only what has been entrusted to him, but also what he has seen, heard or understood. . It is binding on anyone with access to medical information whether or not they are directly involved in care [19].

The Moroccan code of ethics makes medical secrecy an obligation in article 4. It also specifies for these people the circumstances which may lead them to reveal medical information in articles 31 and 56. The penal code provides for penalties of imprisonment from one month to six months and a fine of 200 to 1,000 Dirhams for any person revealing these secrets except in the case where the law obliges or authorizes him to act as a whistleblower [186]. In addition to the patient and the attending physician, access to the patient's record is guaranteed for all health professionals, in the health establishment, who intervene in the care, since the patient's agreement is presumed upon admission to the hospital. establishment. It is also allowed to the legal representative of the patient and to the beneficiaries after his death, as well as for the controlling physicians as indicated in article 28 of law 65–00 on the code of basic medical coverage and article 33 of the decree. 2–05–733 taken for its application, the same for the representatives of the judicial authorities in the event of such

proceedings [179,187].

In France, the legislator understood very well the importance of the establishment of a balance between the right to information and the continuity of care on the one hand and the respect of professional secrecy and the protection of privacy. of the patient on the other hand. Thus it clearly specified the beneficiaries of the right of access to the patient's record and the terms of this access. These people are [19,188]:

- The patient himself who has direct access to his record except:
 - o The minor child who only has an indirect right of access;
 - o In the context of hospitalization at the request of a third party or compulsory hospitalization;
 - o The adult subject to a guardianship measure cannot obtain communication of his record;
- The beneficiaries of the deceased patient have limited access. The healthcare professional, before authorizing it, should therefore check:
 - o The patient's lack of opposition expressed during his lifetime;
 - o The identity of the applicant and his status as beneficiary through the production of an official document;
 - o The motivation for access, which can only result from one of the following three reasons:
 - Know the causes of death;
 - To assert his rights ;
 - Defend the memory of the deceased.
- Parents of a minor child in the absence of his opposition. Parents of adult children are considered as third parties and they cannot access medical information concerning their child except with their consent.
- Third parties: some doctors and jurisdictions well defined by law:
 - o Doctors who participate in patient care: The hospital doctor's access to the hospitalized patient's record is an important element for the continuity and coordination of care, and the limitation of examinations and medical acts.
 - o The doctors designated by the patient to read the record
 - The intermediary doctor who helps the patient to understand the elements appearing in his record;
 - The conciliator of the commission for relations with users and the quality of care;
 - The occupational physician, his access is only possible with the written consent of the employee.
- Doctors authorized by law to access the record:
 - o The doctor responsible for the medical information department;
 - o Social security medical advisers;

- o Medical inspectors of health;
 - o Expert doctors from ANAES;
- The insurance company doctor cannot, under any circumstances, access medical information concerning an insured;
- Legal access to the patient's record:
 - o The medical expert before the courts who must strictly respect the framework of his mission;
 - o Entering the patient's record in the event of a criminal investigation: it is during an investigation or an investigation that the file

D. Archiving

If the keeping, the content and the communication of the patient's record represent three pillars determining the quality of care, an archiving done in good conditions is another. It meets three essential objectives [19]:

- The traceability of medical acts and prescribed care, to ensure continuity of patient care and easy communication between the various actors who work with the patient;
- A medico-legal interest by keeping documents which constitute proof that can be used in the event of legal action being brought, particularly in the area of medical liability;
- A medico-economic interest since the archived file authorizes research and studies to improve medical practices and health management. This is only possible if accessibility to archived documents is ensured.

The internal regulations of hospitals make the hospital, in Morocco, the owner of the medical record and responsible for its preservation during hospitalization and after discharge. As a result, the medical record must always remain within the health establishment where it was designed and the doctor and the patient as well as the various persons specified above only have a right of access to it. However, Moroccan legislation remains poor in terms of medical archiving [177].

On the contrary, in France, the archiving of the medical record is subject to exhaustive regulations. It also makes the hospital the owner of the medical record and responsible for its preservation. These records before being archived must undergo a sorting by the practitioner who designed them in order to keep only the documents necessary and useful for the subsequent follow-up of the patient and the documents that the regulations require to keep [19]. Computer media currently have the same legal value as paper. They have the advantage in archiving in terms of fidelity, durability, economy of archiving space and speed of access. The retention periods for medical records and archives are set according to criteria of medical specialty, pathology and the nature of the documents. For example, pediatrics, neurology, stomatology and chronic disease records are kept until 70 years. Other documents such as entry and exit records, hereditary disease records and inquiries are kept indefinitely [19].

At the end of this legal archiving period, the elimination of a record must respect two rules:

- First, the doctor responsible for archiving and the director of the establishment must give their consent for the destruction of a record;
- Secondly, any elimination must give rise to the establishment of a report of destruction which represents a discharge of responsibility for the director of the hospital. The patient's file being the property of the hospital cannot under any circumstances be entrusted to doctors in the event of cessation of activity or to the patient's beneficiaries after his death. In the event of the abolition of a public or private health establishment participating in the execution of the public hospital service, its archives must be transferred, either to the departmental archives, or to the archives of the establishment which contains the powers of the abolished establishment [19].

V. Appendix 5: user experience considerations

Several user experience principles were taken into consideration [77,78]:

- **Clarity and simplicity:** in half of a second, users evaluate the design of an application. Visually focus attention on the main button versus a bunch of buttons on the page.
- **Scannability:** making a page scannable will appeal to the audience. Most will scan the content for something that strikes them, and then they switch to reading when they want to find out more.
- **Consistency:** providing a straightforward, consistent design is simpler for users. They can then know what to expect when reusing colors, behaviors, and aesthetics, which reduces the need for them to figure out the interface. When users are familiar with some of the design aspects, it makes the process clearer and easier to use.
- **Common elements:** creativity with standardized patterns can make your interface hard to work with and not promote usability. Although one may think non-traditional is appealing, it may make it harder for users to navigate, and thus it falls into a problem area. Creativity and usability need to have a balance.
- **Audience research:** when you use styles and designs that your audience is already comfortable with, they can be eased into your site. You can then differentiate yourself with your ideas on their needs.
- **Hierarchy:** when putting the most critical elements on the interface, highlight them so that users focus on them. There are many ways to highlight things, but the most effective is making it larger than anything else on the screen.
- **Less is more:** reducing the operational and cognitive costs of the users. In placing value on this, the design's usability and consistency improve.
- **Confirmation:** accidents happen all the time. One typical digital example: a person may unintentionally place an order. The design should help correct this, though, because you do not want to give the user a poor experience. This makes confirmation another one of the essential UX design principles.

- **Control:** user control focuses on greater flexibility of use and better control of where a user is within a design or product, enhancing user experience. Furthermore, it allows users to backpedal and recover from errors.
- **Visual grammar:** visual grammar has its roots in graphic design, but it sits at the pivot of all visual communication, which plays a significant role in user experience. Visual grammar consists of everything that makes up the visual elements of design: icons, illustrations, patterns, and more.

Abstract

Abstract

Meta-research is the study of research through the use of research methods. It aims to reduce waste and increase research quality in all fields. Data quality, governance, and querying are pivotal elements in data and research lifecycles; thus, automating them helps establish continuous research pipelines.

Our objective was to introduce health informatics solutions combined with data science solutions to establish a continuous research pipeline and assert data quality attributes and data governance.

The software development lifecycle dictated our methodology; we've employed the Agile methodology and Scrum framework in our workflow; standardization, structuring, software quality, security, legislative, and user experience considerations were our major concerns.

We used "meta" suite to create a multi-purpose content management system and a universal data management system, and we've encoded content schemes to instruct visual components. The resulting system consists of 8 major containers: hospital management system, practice management system, electronic medical record, multi-purpose content management system, universal data management system, identity, and access management system, internationalization system in addition to the ontology container.

We've developed an ontology container to augment our entry system capabilities, which represent concepts in a tree-like structure; each concept is encoded with a prefix string following a Trie data structure, which enables static operations in $O(1)$ time; We've added to the database 270,516 concepts for anatomy, diagnoses, findings, interventions, procedures, medications, organisms, substances in addition to dozen other attributes.

The software was tested by three levels of testing: unit, integration, and acceptance testing; Our acceptance testing objectives were to submit, investigate, and query data using real medical records. To assert these objectives, we've submitted 19 real patient medical records for a two months' period; Submitted data include patient demographics, history notes, examination notes, prescriptions sheets, requested procedures, interventions, diagnoses, and progress notes; We've created and executed data queries exploring clinical, para-clinical, diagnoses, prognosis, and therapeutic parameters.

Introducing health informatics impacts clinical research directly and improves healthcare by digitizing paper trails, asserting quality attributes, asserting data governance, enabling data querying and smart healthcare, and opening the door for other automation techniques. Many research aspects can be automated thanks to artificial intelligence, natural language processing algorithms, and expert systems.

Resumé

La méta-recherche est l'étude de la recherche par l'utilisation de méthodes scientifiques. Elle vise à réduire les dépenses et à améliorer la qualité de la recherche dans tous les domaines. La qualité des données, la gouvernance et l'interrogation sont des éléments essentiels dans les cycles de vie des données et de la recherche ; ainsi, leur automatisation permet d'établir des pipelines de recherche continues. Notre objectif était d'introduire des solutions d'informatique de santé combinées à des solutions de science des données afin d'établir un pipeline de recherche continu et d'affirmer les attributs de qualité des données et la gouvernance des données.

Le cycle de vie du développement logiciel a dicté notre méthodologie ; nous avons utilisé la méthodologie Agile et le cadre Scrum dans notre flux de travail ; la normalisation, la structuration, la qualité des logiciels, la sécurité, la législation et l'expérience utilisateur ont été nos principales préoccupations.

Nous avons utilisé la suite "meta" pour créer un système de gestion de contenu polyvalent et un système de gestion de données universel, et nous avons encodé des schémas de contenu pour instruire les composants visuels. Le système qui en résulte est composé de 8 conteneurs principaux : système de gestion hospitalière, système de gestion de pratique, dossier médical électronique, système de gestion de contenu polyvalent, système de gestion de données universel, système de gestion d'identité et d'accès, système d'internationalisation en plus du conteneur d'ontologie.

Nous avons développé un conteneur d'ontologie pour augmenter les capacités de notre système d'entrée, qui représente les concepts dans une structure arborescente ; chaque concept est encodé avec une chaîne de préfixe suivant une structure de données Trie, qui permet des opérations statiques en temps $O(1)$; Nous avons ajouté à la base de données 270 516 concepts pour l'anatomie, les diagnostics, les résultats, les interventions, les procédures, les médicaments, les organismes, les substances en plus de douzaines d'autres attributs.

Le logiciel a été testé par trois niveaux de tests : tests unitaires, d'intégration et d'acceptation ; Nos objectifs de tests d'acceptation étaient de soumettre, d'examiner et d'interroger des données en utilisant de vrais dossiers médicaux. Pour atteindre ces objectifs, nous avons soumis 19 dossiers médicaux de patients réels sur une période de deux mois. Les données soumises comprennent des données démographiques sur les patients, antécédents, examen clinique, fiches de prescription, des bilans demandés, des interventions, des diagnostics et les mises au points.

L'introduction de l'informatique de santé a un impact direct sur la recherche clinique et améliore les soins de santé en numérisant les traces écrites, en affirmant les attributs de qualité, en affirmant la gouvernance des données, en permettant l'interrogation des données et les soins de santé intelligents, et en ouvrant la porte à d'autres techniques d'automatisation. De nombreux aspects de la recherche peuvent être automatisés grâce à l'intelligence artificielle, aux algorithmes de traitement du langage naturel et aux systèmes experts.

ملخص

التحليل التلوي للبحوث هو دراسة البحث من خلال استخدام الأساليب العلمية. يهدف إلى تجميع الأدلة من دراسات البحث في جميع المجالات. تعد جودة البيانات والحوكمة والسجلات عنصراً أساسياً في دورات حياة كل من البيانات والبحوث؛ وبالتالي، فإن أهمتهم تساهم في إنشاء خطوط أنابيب بحث مستمرة

وبالتالي فإن الهدف من هذه الأطروحة هو إدماج حلول الحوسبة الصحية جنباً إلى جنب مع حلول علوم البيانات لإنشاء خط أنابيب بحثي مستمر وتأكيد سمات جودة البيانات وحوكمة البيانات.

لقد حددت دورة حياة تطوير البرمجيات من حيثها؛ استخدمنا منهجية Agile وإطار عمل Scrum في سير العمل لدينا؛ كان التوحيد القياسي والهيكل وجودة البرامج والأمن والتشبيعات ونجربة المستخدم من اهتماماتنا الرئيسية

استخدمنا مجموعة "meta" لإنشاء نظام إدارة محتوى متعدد المستويات ونظام عالمي لإدارة البيانات، وقمنا ببرمجة مخططات المحتوى لتوجيه المكونات الرئيسية.

يتكون النظام الناتج من 8 حاويات رئيسية: نظام إدارة المنشآت، ونظام إدارة الممارسة، والسجل الطبي الإلكتروني، ونظام إدارة المحتوى متعدد الأغراض، ونظام إدارة البيانات الكوني، ونظام إدارة الهوية والوصول ونظام التدويل بالإضافة إلى حاوية الأنطولوجيا.

لقد قمنا بتطوير حاوية الأنطولوجيا لتدعيم قدرات نظام الإدخال لدينا، عبر تمثيل المفاهيم في بنية بيانات شجرية؛ يتم ترميز كل مفهوم بسلسلة بادئة تتبع بنية بيانات Trie، والتي تسمح بعمليات ثابتة في وقت O(1)؛ لقد أضفنا 270.516 مفهوم إلى قاعدة البيانات تم التشريح والتشخيص والتناجج والإجراءات والإجراءات والأدوية والكائنات الحية والمواد بالإضافة إلى عشرات السمات الأخرى.

تم اختبار البرنامج من خلال ثلاثة مستويات من الاختبار: اختبارات الوحدة والتكامل والقبول؛ كانت أهداف اختبار القبول لدينا هي تدعيم ومراجعة واستجواب البيانات باستخدام سجلات طبية حاسوبية. لنحقيق هذه الأهداف، قمنا بإدخال 19 سجلاً طبياً نغزاً للمرضى على مدار شهرين. تتضمن البيانات المقدمة التركيبات السريرية للمرضى، والتاريخ، والفحص السريري، وسجلات الوصفات الطبية، والفحوصات المطلوبة، والإجراءات، والتشخيصات والتحديثات.

يؤثر إدخال المعلومات الصحية على الأبحاث السريرية بشكل مباشر ويحسن الرعاية الصحية من خلال رؤية المسارات الوراثية، وتأكيد سمات الجودة، وتأكيد حوكمة البيانات، وتمكين السجلات عن البيانات والرعاية الصحية الذكية، ونواح الباب للبيانات الأتمتة الأخرى. يمكن أتمتة العديد من جوانب البحث بفضل الذكاء الاصطناعي وخوارزميات معالجة اللغة الطبيعية والأنظمة الخبيرة.

Bibliography

1. **J. P. A. Ioannidis, D. Fanelli, D. D. Dunne, and S. N. Goodman**
“Meta-research: Evaluation and Improvement of Research Methods and Practices,” *PLOS Biology*, vol. 13, no. 10, 2015, doi: 10.1371/journal.pbio.1002264.
2. **E. Chung, N. Kwon, and J. Lee**
“Understanding scientific collaboration in the research life cycle: Bio- and nanoscientists’ motivations, information-sharing and communication practices, and barriers to collaboration,” *Journal of the Association for Information Science and Technology*, vol. 67, no. 8, pp. 1836–1848, 2015, doi: 10.1002/asi.23520.
3. **A. Goben and R. Raszewski**
“The data life cycle applied to our own data,” *Journal of the Medical Library Association : JMLA*, vol. 103, no. 1, pp. 40–44, 2015, doi: 10.3163/1536-5050.103.1.008.
4. **DDI**
Overview of the DDI Version 3.0 Conceptual Model. 2004.
5. **S. Higgins**
“The lifecycle of data management,” *Managing Research Data*, pp. 17–46, 2012, doi: 10.29085/9781856048910.003.
6. **C. R. Kothari**
Research Methodology: Methods and Techniques. New Age International, 2004.
7. **Pro-Academic-Writers.com**
“Limitations in Research: Writing about Your Study Limitations.” <https://pro-academic-writers.com/blog/limitations-in-research> (accessed Oct. 05, 2020).
8. **Z. Claybaugh**
“Research Guides: Organizing Academic Research Papers: Limitations of the Study.” <https://library.sacredheart.edu/c.php?g=29803&p=185934> (accessed Oct. 05, 2020).
9. **ProThesisWriter.com**
“Limitations in Research: What are Limitations in a Study (Examples).” <https://prothesiswriter.com/blog/limitations-in-research> (accessed Oct. 05, 2020).
10. **C. Orpen**
“The Causes and Consequences of Academic Procrastination: a research note,” *Westminster Studies in Education*, vol. 21, no. 1, pp. 73–75, Jan. 1998, doi: 10.1080/0140672980210107.

11. **A. Nejatizadeh, M. Sarnayzadeh, K. Kahnouji, R. Ghasemi, and N. Nakhodaei**
“Constraining Factors of Research among faculty members at Hormozgan University of Medical Sciences,” *Electron Physician*, vol. 8, no. 5, pp. 2405–2409, May 2016, doi: 10.19082/2405.
12. **J. P. A. Ioannidis**
“Meta-research: The art of getting it wrong,” *Research Synthesis Methods*, vol. 1, no. 3–4, pp. 169–184, 2010, doi: 10.1002/jrsm.19.
13. **T. E. Hardwicke et al.**
“Data availability, reusability, and analytic reproducibility: evaluating the impact of a mandatory open data policy at the journal Cognition,” *Royal Society Open Science*, vol. 5, no. 8, p. 180448, doi: 10.1098/rsos.180448.
14. **J. P. A. Ioannidis**
“Why Most Published Research Findings Are False,” *PLOS Medicine*, vol. 2, no. 8, p. e124, Aug. 2005, doi: 10.1371/journal.pmed.0020124.
15. **S. Falan and B. Han**
“Moving towards efficient, safe, and meaningful healthcare: issues for automation,” *International journal of electronic healthcare*, vol. 6, no. 1, pp. 76–93, 2011.
16. **T. B. Sheridan**
Humans and automation: System design and research issues. Santa Monica, CA, US: Human Factors and Ergonomics Society, 2002, pp. xii, 264.
17. **P. Degoulet and M. Fieschi**
Traitement de l’information médicale : Méthodes et applications hospitalières. 1991.
18. **F. Pradeau, M. Pradeau, E. Benz-Lemoine, and P. Ingrand**
“Le dossier du patient dans les établissements de santé. Tenue, contenu, archivage et communication – aspects médico-légaux – revue de littérature,” *Journal de Médecine Légale Droit Médical*, vol. 41, pp. 355–366, 1998.
19. **ANAES**
“Evaluation des pratiques professionnelles dans les établissements de santé ; Dossier du patient: amélioration de la qualité de la tenue et du contenu réglementations et recommandations,” Paris, 2003.

20. **C. Honnorat**
“Apprentissage de l’exercice médical : Le Dossier Médical. Université de Rennes 1 – Faculté de Médecine.” 2005, [Online]. Available: <http://www.med.univ-rennes1.fr/resped/s/mg/AEMDQ04.pdf>.
21. **H. Falcoff**
Dossier médical en médecine générale. La revue du praticien, médecine générale. 1997.
22. **G. Moutel**
Évolution du dossier médical, nouveaux enjeux de la relation médecins– soignants patients: approche historique, médicale, médico-légale et éthique, Faculté de médecine, Université Paris 5. Paris, 2004.
23. **SOCIETE SCIENTIFIQUE DE MEDECINE GENERALE**
Recommandations de bonne pratique: le dossier médical. SOCIETE SCIENTIFIQUE DE MEDECINE GENERALE, 2001.
24. **Collège des médecins du québec**
Guide d'exercice : la tenue des dossiers par le médecin en centre hospitalier de soins généraux et spécialisés. Collège des médecins du québec, 2005.
25. **O. Kandel, B. Gavid, C. Roy, and J. Maugard**
“Le dossier médical informatisé : Comment peut-il contribuer à l'évaluation des pratiques professionnelles,” vol. 20, 742 vols., La revue du praticien – médecine générale, pp. 967–969.
26. **Z. Souriti et al.**
Évaluation du dossier médical en neurologie, vol. 57, 1 vols. Revue d'Épidémiologie et de Santé Publique, 2009.
27. **M. Goumberk**
“Approche évolutive du dossier du patient,” Faculté de médecine et de pharmacie de Fès – UCA, Fez, 2009.
28. **F. Boulay, T. Chevallier, Y. Gendreike, V. Mailland, Y. Joliot, and R. Sambuc**
Méthode d'audit de la qualité du dossier médical : audit de 467 dossiers médicaux dans le cadre du contrôle de qualité PMSI, vol. 10, 1 vols. 1998.
29. **French Republic**
Conseil Supérieur des Systèmes d'Information de Santé. 1999.

30. **M. Astrid and V. Ginneken**
“The computerized patient record: balancing effort and benefit,” *International Journal of Medical Informatics*, pp. 97–119, 2002.
31. **K. Häyrinen, K. Saranto, and P. Nykänen**
“Definition, structure, content, use and impacts of electronic health records: A review of the research literature,” *International journal of medical informatics*, vol. 2008, no. 77, pp. 291–304, 2008.
32. **F. Bourdillon**
“Le dossier du patient. EMC – Médecine,” vol. 2, 4 vols., 2005, pp. 385–391.
33. **French Republic**
Avis n°104 : Le dossier médical personnel et l’informatisation des données de santé. Paris, 2007.
34. **NIH**
Health Informatics. National Institutes of Health.
35. **W. R. Hendee**
“To Err is Human: Building a Safer Health System,” *Journal of Vascular and Interventional Radiology*, vol. 12, no. 1, 2001, doi: 10.1016/s1051-0443(01)70072-3.
36. **K. Gai, M. Qiu, L.-C. Chen, and M. Liu**
“Electronic Health Record Error Prevention Approach Using Ontology in Big Data,” 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems, 2015, doi: 10.1109/hpcc-css-icess.2015.168.
37. **PrognosisCIS**
EHR Software: Electronic Medical Record Software(EMR) PrognosisCIS EHR. 2020.
38. **R. Amarasingham, L. Plantinga, M. Diener-West, D. J. Gaskin, and N. R. Powe**
“Clinical Information Technologies and Inpatient Outcomes,” *Archives of Internal Medicine*, vol. 169, no. 2, p. 108, 2009, doi: 10.1001/archinternmed.2008.520.
39. **L. Chen, M. A. Babar, and B. Nuseibeh**
“Characterizing Architecturally Significant Requirements,” *IEEE Software*, vol. 30, no. 2, pp. 38–45, 2013.

40. **MobileTech**
Mobile Tech Contributions to Healthcare & Patient Experiences. 2014.
41. **P. C. Tang, J. S. Ash, D. W. Bates, J. M. Overhage, and D. Z. Sands**
“Personal Health Records: Definitions, Benefits, and Strategies for Overcoming Barriers to Adoption,” *Journal of the American Medical Informatics Association*, vol. 13, no. 2, pp. 121–126, 2006, doi: 10.1197/jamia.m2025.
42. **Existek Blog**
Hospital Management System: Features, Modules, Functions, Advantages: Existek Blog. 2020.
43. **B. Nock**
What Is a Medical Practice Management System? 2017.
44. **E. S. Berner and T. J. L. Lande**
“Overview of Clinical Decision Support Systems,” *Health Informatics Clinical Decision Support Systems*, pp. 3–22, 2007, doi: 10.1007/978-0-387-38319-4_1.
45. **A. K. Hamoud, A. S. Hashim, and W. A. Awadh**
“Clinical Data Warehouse A Review,” *Iraqi Journal for Computers and Informatics*, vol. 44, no. 2, 2018, doi: 10.25195/2017/4424.
46. **AAMC**
“Basic Science,” *Tomorrow’s Doctors, Tomorrow’s Cures* | AAMC, 2016.
<https://www.aamc.org/what-we-do/mission-areas/medical-research/basic-science>.
47. **B. Röhrig, J.–B. Du Prel, D. Wachtlin, and M. Blettner**
“Types of study in medical research: part 3 of a series on evaluation of scientific publications,” *Deutsches Arzteblatt International*, vol. 106, no. 15, p. 262, 2009.
48. **L. Cai and Y. Zhu**
“The Challenges of Data Quality and Data Quality Assessment in the Big Data Era,” *Data Science Journal*, vol. 14, p. 2, 2015, doi: 10.5334/dsj-2015-002.
49. **G. Lăzăroiu, M. Kovacova, J. Kliestikova, P. Kubala, K. Valaskova, and V. V. Dengov**
“Data governance and automated individual decision-making in the digital privacy General Data Protection Regulation,” *Administratie si Management Public*, no. 31, pp. 132–142, 2018.

50. **L. Pouchard**
“Revisiting the Data Lifecycle with Big Data Curation,” *International Journal of Digital Curation*, vol. 10, no. 2, pp. 176–192, 2016, doi: 10.2218/ijdc.v10i2.342.
51. **C. Tenopir et al.**
“Data Sharing by Scientists: Practices and Perceptions,” *PLoS ONE*, vol. 6, no. 6, 2011, doi: 10.1371/journal.pone.0021101.
52. **A. Ball**
Review of data management lifecycle models. University of Bath, IDMRC, 2012.
53. **F. J. Hüser, M. K. Elbæk, and P. Martinez lavanchy**
DTU Research Data Life Cycle. Technical University of Denmark, 2016.
54. **J. Dooley**
Software Development and Professional Practice. 2011.
55. **YourDictionary**
“SOFTWARE | 8 Definitions of Software – YourDictionary.”
<https://www.yourdictionary.com/software> (accessed Oct. 28, 2020).
56. **Electrical Engineering**
“Modern Software Engineering Concepts and Practices,” *Advances in Computer and Electrical Engineering*, 2011, doi: 10.4018/978-1-60960-215-4.
57. **J. Fewell**
Agile vs. agile. 2017.
58. **Beck and Kent**
Manifesto for Agile Software Development. 2013.
59. **D. Batra**
“Adapting Agile Practices for Data Warehousing, Business Intelligence, and Analytics,” *Journal of Database Management*, vol. 28, no. 4, pp. 1–23, 2017, doi: 10.4018/jdm.2017100101.
60. **I. Fisher**
“Agile: Not a Methodology,” *PM-ProCommunity*, May 18, 2020. <https://www.pm-procommunity.com/post/agile-not-a-methodology> (accessed Oct. 29, 2020).

61. **InfoQ**
“What Exactly is the Agile Mindset?” <https://www.infoq.com/articles/what-agile-mindset/> (accessed Oct. 29, 2020).
62. **AgileConnection**
“Agile Is Not a Process, It’s a Mindset,” AgileConnection.
<https://www.agileconnection.com/article/agile-not-process-it-s-mindset> (accessed Oct. 29, 2020).
63. **ScrumGuides**
“Scrum Guide | Scrum Guides.” <https://www.scrumguides.org/scrum-guide.html> (accessed Oct. 28, 2020).
64. **T. Preston-Werner,**
Semantic Versioning 2.0.0-rc.2. 2013.
65. **C. Kaner**
“Exploratory Testing.” 2006, [Online]. Available: <http://www.kaner.com/pdfs/ETatQAI.pdf>.
66. **P. Bourque and R. Dick**
Guide to the Software Engineering Body of Knowledge. IEEE Computer Society, 2018.
67. **K. Wiegers**
Creating a Software Engineering Culture. 2013.
68. **R. Blinder**
Testing Object-Oriented Systems: Objects, Patterns, and Tools. 2000.
69. **B. Beizer**
Software Testing Techniques (Second ed.). New York: Van Nostrand Reinhold, 1990.
70. **R. Cimperman**
UAT Defined: A Guide to Practical User Acceptance Testing. UAT Defined: A Guide to Practical User Acceptance Testing, 2006.
71. **K. Dauch, A. Hovak, and R. Nestler**
“Information Assurance Using a Defense In-Depth Strategy,” 2009 Cybersecurity Applications & Technology Conference for Homeland Security, 2009, doi: 10.1109/catch.2009.25.

72. **Kingdom of Morocco**
Produser, Textes et lois.
73. **Kingdom of Morocco**
Loi 09-08.
74. **B. K. Atchinson and D. M. Fox**
“From The Field: The Politics Of The Health Insurance Portability And Accountability Act,”
Health Affairs, vol. 16, no. 3, pp. 146-150, 1997, doi: 10.1377/hlthaff.16.3.146.
75. **US Congress**
104th Congress, 1st Session, S.1028. 1995.
76. **All About UX**
“User experience definitions,” All about UX, 2012. <http://www.allaboutux.org/ux-definitions>.
77. **B. Pines**
“7 UX Principles for Creating a Great Website,” Webdesigner Depot, 2018.
<https://www.webdesignerdepot.com/2018/05/7-ux-principles-for-creating-a-great-website/>.
78. **Springboard**
“16 Important UX Design Principles for Newcomers,” Springboard, 2019.
<https://www.springboard.com/blog/ux-design-principles/>.
79. **R. Mark and N. Ford**
Fundamentals of software architecture: an engineering approach. 2019.
80. **S. Brown**
The C4 model for visualising software architecture. 2020.
81. **P. Black**
“Dictionary of Algorithms and Data Structures,” National Institute of Standards and
Technology, 2009. <https://xlinux.nist.gov/dads/HTML/trie.html>.
82. **Shaffer**
Data Structures & Algorithm Analysis in C++ (3rd ed.). Mineola, NY: Dover, 2011.
83. **E. Reilly**
Concise Encyclopedia of Computer Science. 2004.

84. **E. F. Codd**
“A Relational Model of Data for Large Shared Data Banks,” Commun. ACM, vol. 13, no. 6, pp. 377–387, Jun. 1970, doi: 10.1145/362384.362685.
85. **UPENN**
“Top 10 Web Application Security Vulnerabilities,” 2018.
http://www.upenn.edu/computing/security/swat/SWAT_Top_Ten_A6.php.
86. **P. Hall and O. Martyn**
Software Without Frontiers: A multi-platform, multi-cultural, multi-nation approach. 1996.
87. **T. Masaru**
Generalized LR Parsing. Springer Science & Business Media, 2012.
88. **C. Manning,**
Foundations of Statistical Natural Language Processing. MIT Press, 1999.
89. **Computer Desktop Encyclopedia**
“Solution stack,” Computer Desktop Encyclopedia, 2018.
<https://encyclopedia2.thefreedictionary.com/Solution+stack>.
90. **T. Vimal**
“Difference Between Frontend and Backend MVC,” Joomla!tuts, 2016.
<http://joomlatuts.net/joomla-2-5/87-how-backend-model-view-controller-mvc-works-in-joomla/98-difference-between-frontend-and-backend-mvc>.
91. **J. Barbier**
“It’s Here: Docker 1.0,” 2014. <https://blog.docker.com/2014/06/its-here-docker-1-0/>.
92. **NodeJS**
“Node.js Foundation – Node.js,” 2015. <https://nodejs.org/foundation/>.
93. **P. Bright**
“Microsoft TypeScript: the JavaScript we need, or a solution looking for a problem?,” 2015.
<https://arstechnica.com/information-technology/2012/10/microsoft-typescript-the-javascript-we-need-or-a-solution-looking-for-a-problem/>.
94. **ExpressJS**
“Express.js home page,” 2020. <https://expressjs.com/>.

95. **LessCSS**
"Official Less website," 2020. <http://lesscss.org/>.
96. **Facebook coporation**
"React – A JavaScript library for building user interfaces," 2018. <https://reactjs.org/>.
97. **ApexCharts**
"ApexCharts.js – Open Source JavaScript Charts for your website," 2020.
<https://apexcharts.com/>.
98. **MochaJS**
"Mocha – the fun, simple, flexible JavaScript test framework," 2020. <https://mochajs.org/>.
99. **IstanbulJS**
"Istanbul, a JavaScript test coverage tool," 2020. <https://istanbul.js.org/>.
100. **Nodemon**
"nodemon," 2020. <https://nodemon.io/>.
101. **Webpack**
"Webpack – a module bundler for javascript," 2020. <https://webpack.js.org/>.
102. **Palantir**
"TSLint – Palantir Open Source," 2020. <https://palantir.github.io/tslint/>.
103. **T. D. Gunter and N. P. Terry**
"The Emergence of National Electronic Health Record Architectures in the United States and Australia: Models, Costs, and Questions," J Med Internet Res, vol. 7, no. 1, p. e3, Mar. 2005, doi: 10.2196/jmir.7.1.e3.
104. **P. J. Tuite and K. Krawczewski**
"Parkinsonism: A Review-of-Systems Approach to Diagnosis," Semin Neurol, vol. 27, no. 02, pp. 113-122, Mar. 2007, doi: 10.1055/s-2007-971174.
105. **EMR Consultant**
"Hospital information systems definition given by US based," 2010.
<http://www.emrconsultant.com/education/hospital-information-systems>.
106. **AMA**
"A Clinician's Guide to Electronic Prescribing," American Medical Association, 2011.

107. **DHHS**
“How does e-prescribing work?,” U.S. Department of Health and Human Services, 2011.
108. **ACR**
“E-Prescribing,” American College of Rheumatology, 2011.
109. **Faden and Beauchamp**
A History and Theory of Informed Consent. 1986.
110. **New England Journal of Medicine**
“Informed Consent for Comparative Effectiveness Trials,” New England Journal of Medicine, vol. 370, no. 20, pp. 1958–1960, 2014, doi: 10.1056/NEJMc1403310.
111. **Beauchamp, Childress, and James**
Principles of Biomedical Ethics. New York: Oxford University Press, 1994.
112. **S. Shin**
“Introduction to json (javascript object notation),” Presentation www.javapassion.com, 2010.
113. **IETF**
“The JavaScript Object Notation (JSON) Data Interchange Format – RFC 7158,” Internet Engineering Task Force (IETF), 2013.
114. **IETF**
“Internet Security Glossary, Version 2 – RFC 4949,” Network Working Group, 2007.
<https://tools.ietf.org/html/rfc4949>.
115. **Ferraiolo and Kuhn**
“Role-Based Access Control.” 15th National Computer Security Conference: 554–563, 1992, [Online]. Available: <http://csrc.nist.gov/groups/SNS/rbac/documents/ferraiolo-kuhn-92.pdf>.
116. **R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman**
“Role-based access control models,” Computer, vol. 29, no. 2, pp. 38–47, 1996.
117. **NIST**
“Guide to Attribute Based Access Control (ABAC) Definition and Considerations.” NIST, ABAC, 2014, [Online]. Available: <https://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.SP.800-162.pdf>.

118. **NIST**
“A Comparison of Attribute Based Access Control (ABAC) Standards for Data ServiceApplications.” NIST, 2016.
119. **C.-C. Teng, F. N. Mensah, J. Ekstrom, R. Helps, and C. G. Jensen**
“Hierarchical Role-Based Access Control for Multi-User Collaborative CAD Environment,” Computer-Aided Design and Applications, vol. 13, no. 3, pp. 397-408, 2016, doi: 10.1080/16864360.2015.1114398.
120. **L. Vince**
“What is User Management? – Essential IT – JumpCloud,” JumpCloud, 2019.
<https://jumpcloud.com/blog/what-is-user-management>.
121. **I. Eggleston Gene (Cary, IL), Hansen, Mitch**
“System having virtual session manager used sessionless-oriented protocol to communicate with user device via wireless channel and session-oriented protocol to communicate with host server,” no. 5771353, Jun. 1998, [Online]. Available: <https://www.freepatentsonline.com/5771353.html>.
122. **E. Dalci**
“CWE-20: Improper Input Validation.” 2020, [Online]. Available: <https://cwe.mitre.org/data/definitions/20.html>.
123. **H. Samet**
The design and analysis of spatial data structures, vol. 85. Addison-Wesley Reading, MA, 1990.
124. **T. Campbell**
“Clinical Data Repository Versus a Data Warehouse — Which Do You Need?,” Health Catalyst Insights, 2014.
125. **PerceptualEdge**
“Stephen Few-Perceptual Edge-Selecting the Right Graph for Your Message-2004.” 2014, [Online]. Available: http://www.perceptualedge.com/articles/ie/the_right_graph.pdf.
126. **L. Goldberg et al.**
“Usability and Accessibility in Consumer Health Informatics: Current Trends and Future Challenges,” American Journal of Preventive Medicine, vol. 40, no. 5, Supplement 2, pp. S187-S197, May 2011, doi: 10.1016/j.amepre.2011.01.009.

127. **D. Ravì et al.**
“Deep Learning for Health Informatics,” *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 1, pp. 4–21, Jan. 2017, doi: 10.1109/JBHI.2016.2636665.
128. **J. Brender, C. Nøhr, and P. McNair**
“Research needs and priorities in health informatics,” *International Journal of Medical Informatics*, vol. 58–59, pp. 257–289, Sep. 2000, doi: 10.1016/S1386–5056(00)00092–7.
129. **K. Säfsten, M. Winroth, and J. Stahre**
“The content and process of automation strategies,” *International Journal of Production Economics*, vol. 110, no. 1, pp. 25–38, Oct. 2007, doi: 10.1016/j.ijpe.2007.02.027.
130. **M. H. Olson and H. C. Lucas**
“The impact of office automation on the organization: some implications for research and practice,” *Commun. ACM*, vol. 25, no. 11, pp. 838–847, Nov. 1982, doi: 10.1145/358690.358720.
131. **Redman and T. C**
“Data driven: profiting from your most important business asset,” *Choice Reviews Online*, vol. 46, no. 06, 2009, doi: 10.5860/choice.46–3345.
132. **K. P. Fadahunsi et al.**
“Protocol for a systematic review and qualitative synthesis of information quality frameworks in eHealth,” *BMJ Open*, vol. 9, no. 3, 2019, doi: 10.1136/bmjopen–2018–024722.
133. **M. Hilbert and P. Lopez**
“The World’s Technological Capacity to Store, Communicate, and Compute Information,” *Science*, vol. 332, no. 6025, pp. 60–65, 2011, doi: 10.1126/science.1200970.
134. **T. Breur**
“Statistical Power Analysis and the contemporary ‘crisis’ in social sciences,” *Journal of Marketing Analytics*, vol. 4, no. 2–3, pp. 61–65, 2016, doi: 10.1057/s41270–016–0001–3.
135. **D. Boyd and K. Crawford**
“Six Provocations for Big Data,” *SSRN Electronic Journal*, 2011, doi: 10.2139/ssrn.1926431.
136. **M. F. Uddin, N. Gupta, et al**
“Seven V’s of Big Data understanding Big Data to extract value,” in *Proceedings of the 2014 zone 1 conference of the American Society for Engineering Education*, 2014, pp. 1–5.

137. **J. Ovreteit, T. Scott, T. Rundall, S. Shortell, and M. Brommelsa**
Implementation of electronic medical records in hospitals: two case studies, vol. 2007, 84 vols. Health Policy, 2007.
138. **ANAES**
La tenue du dossier médical en médecine générale état des lieux et recommandations. Paris: ANAES, 1996.
139. **French Republic**
Inspection générale des finances, Inspection générale des affaires sociales, conseil général des technologies de l'information. Rapport interministériel sur le dossier médical personnel. Paris, 2007.
140. **French Republic**
"Agence des systèmes d'information partagés de santé (ASIP Santé). Dossier médical personnel," Feb. 01, 2010.
http://www.asipsante.fr/index.php?option=com_content&task=view&id=18&Itemid=161.
141. **R. V. Labaree**
"Research Guides: Organizing Your Social Sciences Research Paper: 5. The Literature Review," 2009.
142. **M. Bates**
"Models of natural language understanding," Proc Natl Acad Sci USA, vol. 92, no. 22, p. 9977, Oct. 1995, doi: 10.1073/pnas.92.22.9977.
143. **A. Vaswani et al.**
Attention Is All You Need. 2017.
144. **J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova**
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2019.
145. **S. Idreos, O. Papaemmanouil, and S. Chaudhuri**
"Overview of data exploration techniques," in Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, 2015, pp. 277–281.
146. **Upton and Cook**
Oxford Dictionary of Statistics. OUP, 2006.
147. **Hinton, Geoffrey, Sejnowski, and Terrence**
Unsupervised Learning: Foundations of Neural Computation. MIT Press, 1999.

148. **IBM**
“Getting Started with Querying,” IBM, 2020.
<https://cloud.ibm.com/docs/discovery?topic=discovery-getting-started-with-querying>.
149. **H. S. Dar, M. I. Lali, M. U. Din, K. M. Malik, and S. A. C. Bukhari**
“Frameworks for Querying Databases Using Natural Language: A Literature Review,” arXiv preprint arXiv:1909.01822, 2019.
150. **J. H. Friedman**
“Stochastic gradient boosting,” Computational statistics & data analysis, vol. 38, no. 4, pp. 367–378, 2002.
151. **D. F. Specht**
“A general regression neural network,” IEEE transactions on neural networks, vol. 2, no. 6, pp. 568–576, 1991.
152. **S. Dreiseitl and L. Ohno-Machado,**
“Logistic regression and artificial neural network classification models: a methodology review,” Journal of biomedical informatics, vol. 35, no. 5–6, pp. 352–359, 2002.
153. **I. Goodfellow et al.**
“Generative adversarial nets,” in Advances in neural information processing systems, 2014, pp. 2672–2680.
154. **A. Krizhevsky, I. Sutskever, and G. E. Hinton**
“Imagenet classification with deep convolutional neural networks,” in Advances in neural information processing systems, 2012, pp. 1097–1105.
155. **T. Mikolov, S. Kombrink, L. Burget, J. Černocký, and S. Khudanpur**
“Extensions of recurrent neural network language model,” in 2011 IEEE international conference on acoustics, speech and signal processing (ICASSP), 2011, pp. 5528–5531.
156. **V. Larivière, S. Haustein, and P. Mongeon**
“The Oligopoly of Academic Publishers in the Digital Era,” PLOS ONE, vol. 10, no. 6, pp. 1–15, 2015, doi: 10.1371/journal.pone.0127502.
157. **A. E. Jinha**
“Article 50 million: an estimate of the number of scholarly articles in existence,” Learned Publishing, vol. 23, no. 3, pp. 258–263, 2010, doi: 10.1087/20100308.

158. **IETF**
“The text/markdown Media Type – RFC 7763.” Internet Engineering Task Force (IETF), 2016, [Online]. Available: <https://tools.ietf.org/html/rfc7763>.
159. **Leslie. Lamport**
LATEX: a document preparation system. Reading, Mass.: Addison–Wesley Pub. Co., 1986.
160. **B. Fredson**
Four Faces of Bibliography. Papers of the Bibliographical Society of Canada, 1971.
161. **Wikipedia**
“Reference software,” Wikipedia, 2018. https://en.wikipedia.org/wiki/Reference_software.
162. **INNS Pub**
“Standards of a Quality Research,” [Online]. Available: <https://innspub.net/standards-of-a-quality-research/>.
163. **P. M. Wortman**
“Judging research quality,” The handbook of research synthesis, pp. 97–109, 1994.
164. **L. Chen, M. A. Babar, and B. Nuseibeh**
“Characterizing Architecturally Significant Requirements,” IEEE Software, vol. 30, no. 2, pp. 38–45, 2013.
165. **R. Soley**
“How to Deliver Resilient, Secure, Efficient, and Easily Changed IT Systems in Line with CISQ Recommendations.” 2013, [Online]. Available: http://www.omg.org/CISQ_compliant_IT_Systemsv.4-3.pdf.
166. **D. Zubrow**
Measuring Software Product Quality: the ISO 25000 Series and CMMI. 2004.
167. **Z. D. Kelemen, G. Bényász, and Z. Badinka**
“A measurement based software quality framework,” 2014, doi: 10.13140/2.1.3484.3528.
168. **2017 IEEE International Conference on Software Architecture**
“A Catalog of Security Architecture Weaknesses.” 2017 IEEE International Conference on Software Architecture, 2017, [Online]. Available: <https://design.se.rit.edu/papers/cawe-paper.pdf>.

169. **D. Johnsson, D. Deogun, and D. Sawano**
Secure by Design. 2019.
170. **SecurityArchitecture.org**
“Definitions: IT Security Architecture,” SecurityArchitecture.org, 2006.
<http://www.opensecurityarchitecture.org/cms/definitions/it-security-architecture>.
171. **NullSweep**
“Integrating Security with Agile Development,” 2019. <https://nullsweep.com/integrating-security-with-agile-development/>.
172. **C. Jannsen**
“Security Architecture,” Techopedia. Janalta Interactive Inc., 2014.
<http://www.techopedia.com/definition/72/security-architecture>.
173. **A. Woodie**
“Why ONI May Be Our Best Hope for Cyber Security Now,” 2016.
<http://www.datanami.com/2016/05/09/oni-may-best-hope-cyber-security-now/>.
174. **Reuters**
“Firms lose more to electronic than physical theft,” Reuters, 2010.
<https://www.reuters.com/article/2010/10/18/us-crime-fraud-idUSTRE69H25820101018>.
175. **Bezzi and Michele**
“Privacy and Identity Management for Life,” 2011, doi: 10.1007/978-3-642-20317-6.
176. **F. Strou**
“What Is Identity and Access Management (IAM)? Webopedia Definition,” 2019.
<https://www.webopedia.com/TERM/I/iam-identity-and-access-management.html>.
177. **Kingdom of Morocco**
“Ordre des médecins du maroc. Code de déontologie,” Bulletin officiel, 1953, p. 828.
178. **Kingdom of Morocco**
“Ordre des chirurgiens-dentistes. Code de déontologie. Bulletin officiel n° 3364 du 1er jourmada I 1397,” 1997, p. 523.
179. **Kingdom of Morocco**
“Code de la couverture médicale de base Loi N° 65-00, bulletin officiel N° 5058 -16 Ramadan 1423,” 2002, p. 1333.

180. French Health Departement

“Direction des Hôpitaux et Soins Ambulatoires. Le dossier médical,” 2006.

181. P. Roger

“Le dossier médical dans les établissements de santé. Médecine & Droit,” vol. 2000, 40 vols., 2000, pp. 10–14.

182. S. Ababou

“Elaboration d’un cahier des charges du dossier patient minimal commun,” 2006.

183. French Republic

Ministère de la santé Code de la santé publique, Décret n° 2002–780. 2002.

184. French Republic

Ministère de la santé. Code de la santé publique, décret n° 462–2003. 2003.

185. French Republic,

Ministère de la santé Code de la santé publique, décret n°2002–194 du 11 février 2002 relatif aux actes professionnels et à l’exercice de la profession d’infirmier. 2002.

186. Kingdom of Morocco

Code pénal, Article 446. 1962.

187. Kingdom of Morocco,

Décret n° 2–05–733 du 11 jourmada II 1426 (18 juillet 2005) pris pour l’application de la loi n° 65–00 portant code de la couverture médicale de base. 2005.

188. T. Casagrande

“L’accès direct au dossier médical : Principes juridiques et réalités pratiques,” Médecine & Droit, 2005, pp. 50–5



كلية الطب
والصيدلة - مراكش
FACULTÉ DE MÉDECINE
ET DE PHARMACIE - MARRAKECH

Année: 2020

Thèse N° 218

Informatisation du dossier médical pour automatiser la recherche scientifique médicale

THÈSE

PRÉSENTÉE ET SOUTENUE PUBLIQUEMENT LE 12/11/2020

PAR

Mr. Zakaria CHABIHI

Né le 09 Décembre 1994 à Dammam, Arabie Saoudite

POUR L'OBTENTION DU DOCTORAT EN MÉDECINE

MOTS CLES

Dossier de santé électronique – Système de gestion hospitalière – Système de gestion de contenu – Entrepôt de données cliniques – Ontologie – Big Data – Science des données – Analyse des données – Interrogation des données – Intelligence artificielle

JURY

Mr. A.R. EL ADIB

Professeur d'anesthésie-réanimation

PRESIDENTE

Mr. F.M.R. MAOULAININE

Professeur de pédiatrie

RAPPORTEUR

Mme. L. ADARMOUCH

Professeur de médecine communautaire

JUGES

أطروحة رقم 218

سنة: 2020

رقمنة السجل الطبي من أجل تمكين البحث العلمي الطبي المستمر

الأطروحة

قدمت ونوقشت علانية يوم 12/11/2020

من طرف

السيد زكريا شبیه

المزاداد يوم 09 دجنبر 1994 بالدمام، المملكة السعودية العربية

لنيل شهادة الدكتوراه في الطب

الكلمات الأساسية

السجل الصحي الإلكتروني - نظام إدارة المستشفيات - نظام إدارة المحتوى - مستودع البيانات
السريرية - الأنطولوجيا - البيانات الضخمة - علم البيانات - تحليل البيانات - الاستعلام عن البيانات -
الذكاء الاصطناعي

اللجنة

الرئيسة

المشرف

القضاة

{

السيد. أ.غ. الأديب
استاذ في الإنعاش و التخدير

السيد. ف.م.ر. ماء العينين
استاذ في طب الأطفال

السيدة. ل. الدرמוש
استاذة في طب المجتمع