



Université de la Manouba
École Nationale des Sciences de l'Informatique



RAPPORT DU PROJET DE CONCEPTION ET DE DÉVELOPPEMENT

Sujet : DoctSmart :Vers une aide à la prise de décision médicale

Auteurs :

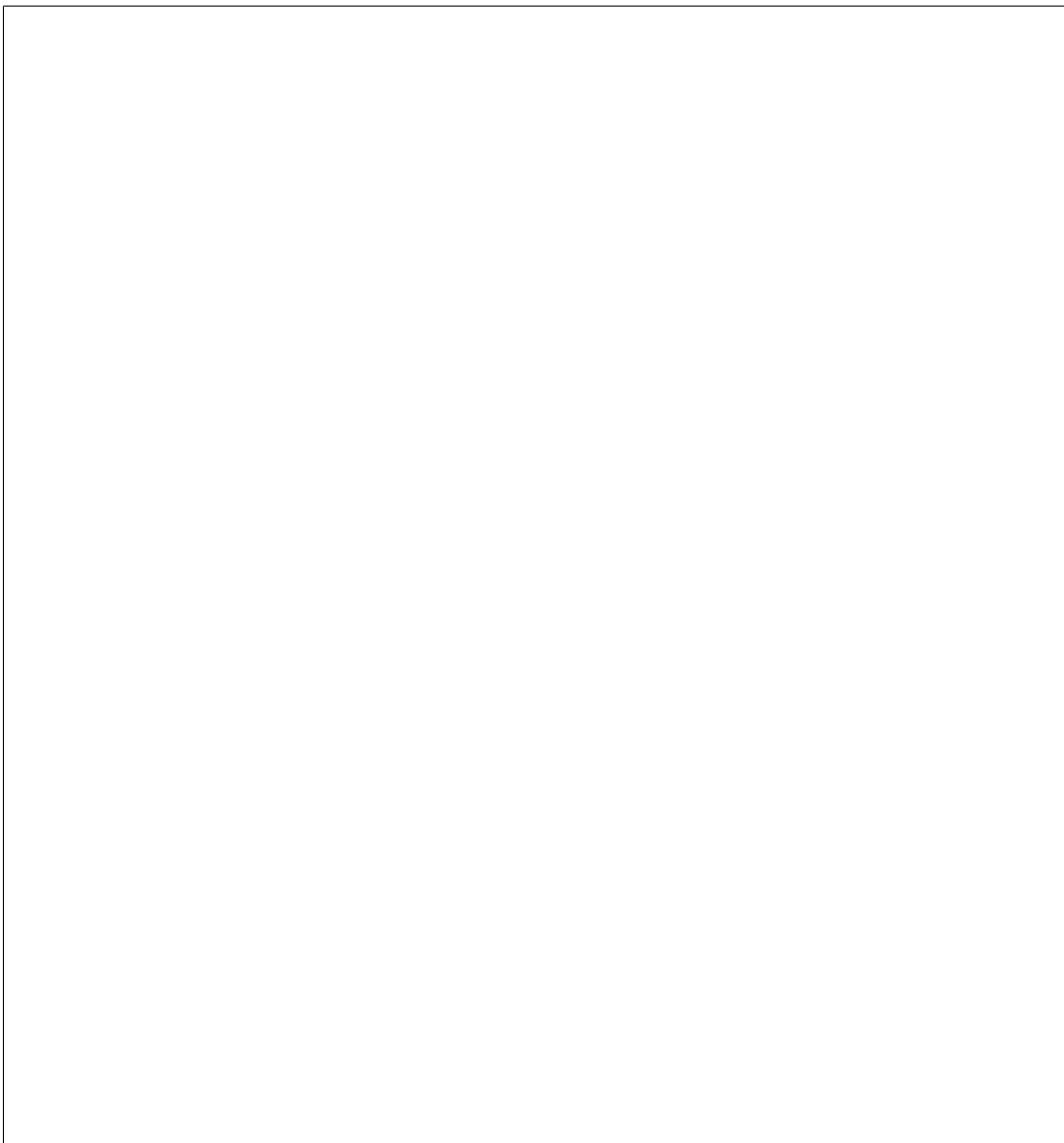
M. Mohamed MEZZI M. fedi KOUKI M Raslen OUARGHI

Encadrante :

Dr. Oumami FADOUA

Année Universitaire :2022 /2023

Appréciations et signature de l'encadrant

A large, empty rectangular box with a thin black border, occupying most of the page below the title. It is intended for handwritten text, such as appreciations or signatures.

Remerciements

Nous souhaitons exprimer toute notre reconnaissance et notre gratitude envers notre encadrante, le Dr Fadoua OUAMANI, pour ses précieuses directives, sa disponibilité et ses conseils avisés qui nous ont permis de mener à bien notre projet.

Nous tenons également à remercier chaleureusement les membres du jury pour l'honneur qu'ils nous ont fait en acceptant d'évaluer notre contribution, ainsi que l'administration et le corps enseignant de l'ENSI pour la formation de qualité que nous avons reçue tout au long de ces deux années.

Nous souhaitons également exprimer notre gratitude envers nos familles et nos camarades pour leur soutien constant et leur encouragement tout au long de ce projet.

Table des matières

Introduction	1
1 Présentation générale du projet	2
1.1 Introduction	2
1.2 Cadre de projet	2
1.3 Concepts clés	3
1.4 Problematique du projet	4
1.5 Étude de l'existant	4
1.5.1 Description de l'existant	4
1.5.2 Critique de l'existant	5
1.5.3 Solution préconisée	6
1.6 Méthodologie de développement adoptée	6
1.6.1 Etude comparative des Méthodologies	6
1.6.2 Sélection du processus de développement en CRISP	7
1.7 Formalisme(UML)	8
1.8 Conclusion	9
2 Analyse et spécification des besoins	10
2.1 Introduction	10
2.2 Analyse des besoins	10
2.2.1 Identification des acteurs	10
2.2.2 Besoins fonctionnels	11
2.2.3 Besoins non fonctionnels	11
2.3 Spécification des besoins	12
2.4 Conclusion	18
3 Conception	19
3.1 Introduction	19
3.2 Conception architecturale du système	19
3.2.1 Architecture physique	19
3.2.2 Architecture logique	21

3.3	Prétraitemenet des données	23
3.4	Les algorithmes de l'apprentissage automatique utilisés	23
3.4.1	Random Forest (forêts aléatoires)[14] :	23
3.4.2	K nearest neighbors (KNN)[15] :	24
3.4.3	Régression Logistique [16] :	26
3.4.4	Naïve bayésienne multinomiale [17] :	27
3.4.5	Les Métriques et les algorithmes d'évaluation	28
3.5	Conclusion	29
4	Realisation	30
4.1	Introduction	30
4.2	Environnements de travail	30
4.2.1	Environnements de développement matériel	30
4.2.2	Environnements de développement logiciel	31
4.3	Élaboration et mise au point des modèles	35
4.3.1	Préparation des données	35
4.3.2	Paramétrage et entrainement des modèles	36
4.3.3	Evaluation et choix du modèle à déployer	43
4.3.4	Déploiement du modèle	45
4.4	Conclusion	51
Conclusion et perspectives		52
Annexe		54
NETOGRAPHIE		56
Bibliographie		58

Table des figures

1.1	L'apprentissage automatique supervisé.[3]	3
1.2	MyChart application [4]	4
1.3	Healow application [5]	5
1.4	La méthodologie Crisp-DM [10]	7
1.5	Chronogramme	8
1.6	UML[12]	8
2.1	Diagramme de cas d'utilisation réceptioniste	12
2.2	Diagramme de cas d'utilisation médecin	13
2.3	Diagramme de cas d'utilisation patient	13
2.4	Diagramme de séquence Authentification	14
2.5	Diagramme de séquence Gérer les comptes	15
2.6	Diagramme d'activité gérer les comptes	16
2.7	Diagramme de séquence ajouter symptomes	17
3.1	Diagramme de déploiement [8]	20
3.2	Architecture du Frontend [13]	21
3.3	Architecture du Backend	22
3.4	Architecture du modèle ai	22
3.5	Random Forest Modèle	24
3.6	KNN modèle	26
3.7	Formule de précision	28
3.8	L'algorithme de validation croisée	29
4.1	Python	31
4.2	Pandas	31
4.3	Jupyter	32
4.4	Flask	32
4.5	NodeJS	33
4.6	React Js	33
4.7	Express Js	33
4.8	Mongo db	34

4.9	Code du nettoyage des données	35
4.10	Code du nettoyage des données	36
4.11	Diagramme d'activité GRID Search	37
4.12	Utilisation de Grid Search dans le modèle MNB	38
4.13	La visualisation des différentes valeurs de cross validation pour RF	39
4.14	Les paramètres optimales pour RF	39
4.15	La visualisation des différentes valeurs de cross validation pour KNN	40
4.16	Les paramètres optimales pour KNN	40
4.17	La visualisation des différentes valeurs de cross validation pour RL	41
4.18	Les paramètres optimales pour RL	41
4.19	La visualisation des différentes valeurs de cross validation pour MNB	42
4.20	Les paramètres optimales pour MNB	42
4.21	Diagramme d'activité de choix du modèle	43
4.22	Le pourcentage d'accuracy par modèle	44
4.23	Le pourcentage de cross validation accuracy par modèle	44
4.24	Le Logo de l'application	45
4.25	Les couleurs de base	46
4.26	La page d'accueil	46
4.27	Les différentes services	47
4.28	L'interface du choix : Login médecin ou réceptioniste	47
4.29	Le Login de réceptioniste	48
4.30	Le SignUp de médecin	48
4.31	Dashbord du médecin	49
4.32	Interface de sélection Des symptomes	49
4.33	Résultat du test médical envoyé Sur e-mail	50

Liste des tableaux

1.1	Critique de l'existant	5
1.2	Etude des différents Modèles	6
3.1	Tableau de paramètres RF.	24
3.2	Tableau de paramètres KNN.	25
3.3	Tableau de paramètres Régression Logistique.	27
3.4	Tableau de paramètres MNB.	28
4.1	Paramètre de la base de données	36
4.2	Gérer un rendez-vous tableau descriptif	54
4.3	Authentification tableau descriptif	55

Liste des sigles et acronymes

IA	<i>intelligence artificielle</i>
RF	<i>Random forest</i>
KNN	<i>K nearest neighbors</i>
MNB	<i>multinomial naive bayes</i>
ML	<i>Machine Learning</i>
CRISP-DM	<i>CRoss-Industry Standard Process for Data Mining</i>
UML	<i>Unified Modeling Language</i>
RGPD	Règlement Général sur la Protection des Données

Introduction Générale

Dans le domaine médical, chaque seconde est précieuse et peut faire la différence entre la vie et la mort. Il est donc impératif que la gestion hospitalière soit extrêmement précise et organisée afin d'assurer un traitement efficace et de qualité aux patients. Les technologies de l'IA ont révolutionné la façon dont nous abordons la santé en fournissant des solutions innovantes pour améliorer l'efficacité et la rapidité de la prise en charge des patients. Imaginez un monde où les patients peuvent obtenir un diagnostic rapide et précis sans avoir à attendre des heures dans les salles d'attente des hôpitaux.

Notre projet de conception et de développement vise à créer une plateforme facile d'utilisation, qui facilite l'accès aux informations et aux services médicaux pour les patients, tout en permettant une communication fluide entre le personnel médical et les patients. L'expérience utilisateur sera optimale, même pour les personnes qui ne sont pas familières avec les technologies numériques, afin de garantir une utilisation facile et intuitive de la plateforme. Notre application offre aux patients la possibilité de récupérer facilement leurs résultats d'analyses médicales en ligne, qui sont diagnostiqués grâce à l'apprentissage automatique pour prédire les résultats, ainsi qu'une évaluation complète de leur état de santé. La prise de rendez-vous est simplifiée grâce à des fonctionnalités innovantes, permettant une communication fluide entre le patient, le médecin et la réceptionniste.

Ce rapport est composé de cinq chapitres distincts. Tout d'abord, nous exposons et analysons les solutions existantes, tout en proposant notre propre solution. Ensuite, nous détaillons les besoins fonctionnels et non fonctionnels de notre projet. Nous abordons ensuite les étapes cruciales de la prédiction et de l'évaluation dans le cadre de l'apprentissage automatique, qui consistent respectivement à anticiper les résultats et à mesurer et améliorer les performances du modèle. Nous poursuivons avec la présentation de la conception de notre solution en détaillant les fonctionnalités et les technologies utilisées. Enfin, nous clôturons notre rapport en présentant la phase de réalisation et en illustrant son fonctionnement grâce à quelques exemples concrets.

Finalement, nous consacrons une conclusion générale à notre travail, dans laquelle nous résumons les objectifs initiaux du projet, faisons une synthèse du travail accompli et proposons des perspectives d'amélioration et de développement futur pour notre application.

Chapitre 1

Présentation générale du projet

1.1 Introduction

Une étude préliminaire approfondie est Essentielle pour la réussite de tout projet, car elle permet de comprendre les conceptes clés et de bâtir un cadre solide pour l'accomplissement des objectifs fixés. Dans le but de mener a terme notre projet , Nous allons adopter une démarche méthodique et rigoureuse, en nous appuyant sur une analyse critique des travaux existants. Cela nous permettra d'identifier la problématique à traiter dans le cadre de notre projet, soutenue par une méthodologie claire et un chronogramme précis qui reflètera les étapes et leur durée afin d'organiser et mener à bien le travail.

1.2 Cadre de projet

Ce projet est l'aboutissement de notre deuxième année d'études à l'École Nationale des Sciences de l'Informatique, il nous offre la possibilité de mettre en pratique les connaissances que nous avons acquises en génie logiciel, en présentant les principes de base de l'apprentissage de machine (ML). En fait, notre objectif est de créer une plateforme web intelligente gratuite intitulée DoctSmart dont l'objectif est de développer un modèle de diagnostic précis pour améliorer la qualité des soins et fournir une assistance médicale efficace, contribuant ainsi au bon fonctionnement des hôpitaux et à la prestation de soins de qualité supérieure aux patients.

1.3 Concepts clés

Définition 1(Intelligence artificielle [1]) : L'intelligence artificielle repose essentiellement sur la création et le développement de modèles et de machines qui couvrent des tâches normalement réservées à l'homme, à savoir la conception, la prise de décision et enfin l'action. Cela se fait à partir d'une grande quantité de données et de capacité de traitement. il est largement utilisé dans différents domaines, de l'éducation à la finance, de la santé au transport.

Définition 2 (Machine Learning[2]) : Le Machine Learning utilise des algorithmes pour Analyser des données avec rigueur et en extraire des modèles statistiques qui peuvent être utilisés pour anticiper des résultats futurs ou résoudre des défis concrets.

Définition 3 (Apprentissage supervisé[2]) : L'apprentissage supervisé est une méthode d'apprentissage automatique qui consiste à fournir un ensemble de données catégorisées pour Construire un système de prédiction capable de faire des prédictions précises sur des données inconnues. Les exemples d'apprentissage sont généralement organisées en couple (caractéristiques, réponse attendue) qui sont utilisées comme guide pour prédire de nouveaux exemples.

Définition 4 : (Apprentissage non supervisé[2]) : L'apprentissage non supervisé permet d'organiser automatiquement des données non étiquetées en catégories ou en groupes. Les deux défis courants dans cette approche sont :

- Le clustering : qui consiste à regrouper des données similaires en sous-groupes homogènes sans étiquetage préalable.
- La réduction de dimension : permet de simplifier la représentation d'un ensemble de données en éliminant les dimensions redondantes ou non significatives tout en conservant la variance des données. afin de faciliter leur analyse et leur traitement.

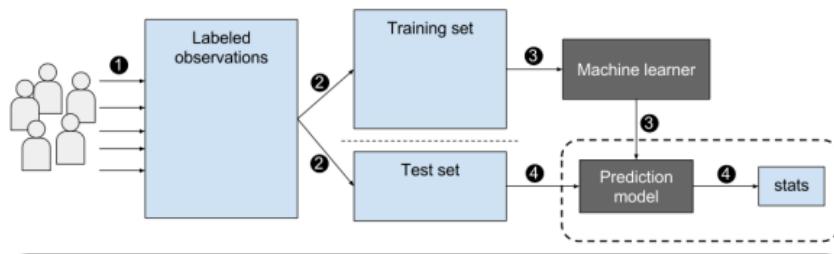


FIGURE 1.1 – L'apprentissage automatique supervisé.[3]

1.4 Problématique du projet

Chaque minute compte dans le domaine médical et une erreur peut engendrer de graves conséquences. La gestion hospitalière doit donc être précise et organisée afin de garantir un traitement efficace et de qualité aux patients. Une étude approfondie a été menée dans ce cadre et nous avons constaté que la procédure d'obtention d'informations sur la maladie devient de plus en plus difficile et complexe, entraînant ainsi une perte de temps. Enfin, le personnel peut rencontrer des difficultés de communication et d'échange d'informations.

1.5 Étude de l'existant

1.5.1 Description de l'existant

Avant de commencer à développer le projet, nous allons prendre le temps d'analyser les applications existantes qui ont déjà tenté de résoudre notre problématique. Nous allons examiner attentivement les avantages et les lacunes des applications existantes pour apprendre de ces éléments et ainsi améliorer notre projet en conséquence.

Mychart : MyChart est une plateforme en ligne qui offre aux patients un accès à leur dossier médical électronique, y compris les résultats de tests, les rendez-vous, les médicaments prescrits et les antécédents médicaux. Les patients peuvent dialoguer avec leur équipe médicale via la plateforme pour poser des questions sur leur santé et demander la prolongation de leurs prescriptions.

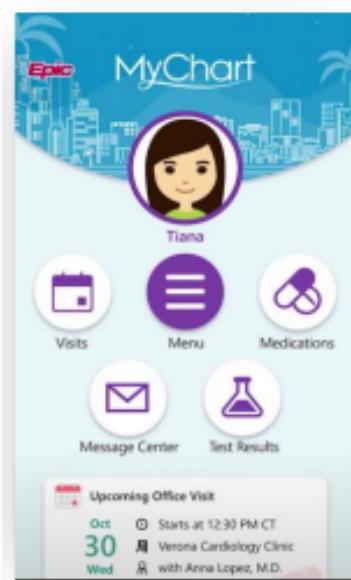


FIGURE 1.2 – MyChart application [4]

Healow : est une Plateforme de santé en ligne Qui permet une communication entre les patients et le Personnel hospitalier , de prendre des rendez-vous en ligne, de remplir des formulaires de santé et de consulter leur dossier médical électronique Pour garantir un suivi médical optimal.



FIGURE 1.3 – Healow application [5]

1.5.2 Critique de l'existant

	Platform	Coûts	Facilité d'utilisation
Mychart	IOS,Android et Web	Payante	Compliqué et complexe
Healow	IOS,Android et Web	Payante	Tortueux

TABLE 1.1 – Critique de l'existant.

1.5.3 Solution préconisée

Après avoir évalué les différentes solutions disponibles, Nous avons constaté que la plupart d'entre elles requièrent des frais ou des coûts supplémentaires. De plus, l'utilisation de ces applications peut être difficile et il peut être compliqué de comprendre les résultats des tests médicaux sans l'avis d'un professionnel de santé. Afin de réaliser notre objectif de projet, Nous proposons une application web gratuite basée sur les algorithmes d'apprentissage automatique pour assister efficacement les médecins dans la prise de décision et minimiser les erreurs des diagnostics

1.6 Méthodologie de développement adoptée

1.6.1 Etude comparative des Méthodologies

Le choix de la bonne méthodologie pour notre projet est essentiel pour atteindre nos objectifs avec succès, c'est pourquoi nous devons évaluer attentivement les différentes options qui s'offrent à nous et choisir celle qui convient le mieux à nos besoins spécifiques.

Méthodes	Avantages	Inconvénients
En cascade [6]	Modèle de prévision Simple et fiable Facile à mettre en oeuvre	Longs cycles de planification Planification à long terme Manque de souplesse Manque de réactivité Peu adaptable aux changement
En V [7]	favorise la collaboration et la communication favorise la détection précoce des erreurs une meilleure traçabilité des exigences du projet.	Plus difficile que le modèle en cascade Basé sur la planification Plus difficile à mettre en oeuvre
En Spirale [8]	Modèle plus évolutif que prédictif Processus d'amélioration continue Amélioration continue.	Manque de recul au départ Nécessite plus d'effort Complexité
En CRISP [9]	Adaptable Collaborative Orientée résultats .	Lourdeur Compliquée Limitée

TABLE 1.2 – Etude des différents Modèles.

1.6.2 Sélection du processus de développement en CRISP

Le CRISP-DM[9] est une méthode largement utilisée dans le domaine de la fouille de données et de l'analyse, qui permet de gérer efficacement les différentes étapes du développement d'un logiciel en adoptant une approche itérative. Chaque étape de la méthodologie doit être vérifiée et validée pour garantir la qualité du produit final. Le modèle en cascade est à éviter car il peut présenter un risque d'effet de tunnel, tandis que la méthode Scrum est souvent difficile à appliquer sans un Scrum Master et un backlog Product. Le modèle en spirale est quant à lui réservé aux projets à haut risque et innovants. [2] En revanche, le modèle CRISP-DM apparaît comme une méthode efficace pour valider les cinq étapes clés de la méthodologie, offrant ainsi une planification anticipée des étapes de conception, de réalisation et de test, assurant une qualité optimale pour le produit final.

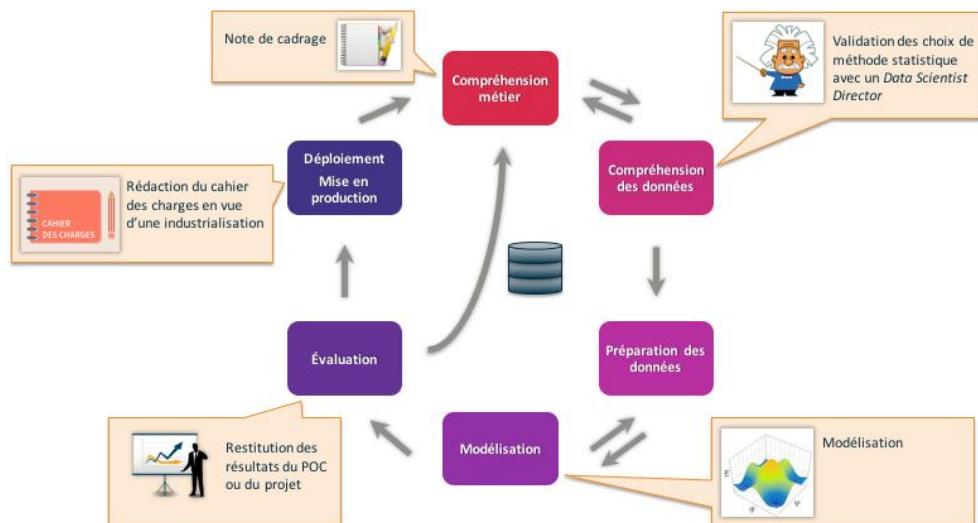


FIGURE 1.4 – La méthodologie Crisp-DM [10]

Le chronogramme de la méthode en CRISP est divisé en 6 étapes clés qui permettent une planification précise et une gestion efficace du projet.

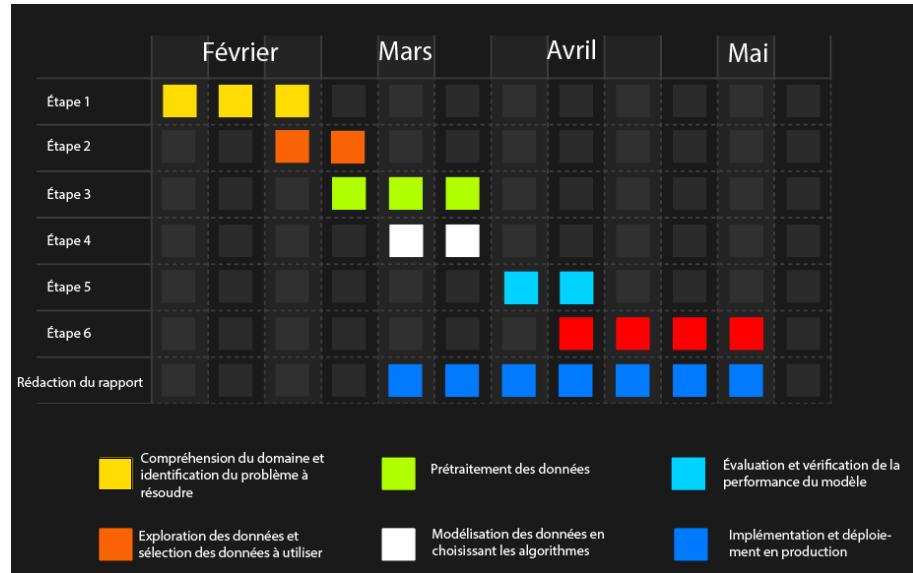


FIGURE 1.5 – Chronogramme

1.7 Formalisme(UML)

L'UML [11], acronyme de Unified Modeling Language ou Langage de Modélisation Uniifié, c'est un langage visuel de modélisation qui a été conçu pour l'architecture, la conception et l'implémentation de systèmes logiciels complexes En termes de leur structure ainsi que de leur comportement. Il offre une sémantique et une syntaxe riches pour assurer une communication efficace entre les membres de l'équipe de développement et assurer la qualité du logiciel.UML utilise des diagrammes pour visualiser la structure et le comportement des objets d'un système en facilitant la compréhension des systèmes logiciels sans recourir au code informatique



FIGURE 1.6 – UML[12]

1.8 Conclusion

En résumé, ce chapitre a abordé la problématique du projet et a examiné les différents états de lieu pour identifier les limites de l'existant. Nous avons également étudié et comparé les applications existantes pour trouver des solutions qui amélioreront l'existant et contourneront ses limites. Enfin, nous avons comparé différentes méthodologies afin de choisir la stratégie qui convient le mieux à notre projet.

Chapitre 2

Analyse et spécification des besoins

2.1 Introduction

Le premier stade du processus de développement consiste à analyser et à définir les exigences du projet. Il s'agit d'une phase où les exigences de l'application sont examinées en détail afin d'établir une description complète des besoins à satisfaire. Ensuite, nous procérons à la modélisation UML pour visualiser d'une manière claire et précise les informations en utilisant les diagrammes UML appropriés.

2.2 Analyse des besoins

Dans cette partie, nous allons identifier et décrire les besoins et les exigences du système, en associant à chaque acteur les différentes actions, tâches et fonctionnalités impliquées dans la spécification des besoins y compris les besoins non fonctionnels pour construire un système qui répond aux attentes et satisfait les utilisateurs.

2.2.1 Identification des acteurs

La plateforme sera accessible à une multitude d'acteurs :

- 1. Réceptionniste** : Ce membre du personnel sera chargé de gérer les patients et les médecins dans le système, il sera également en charge de planifier les rendez-vous pour les patients.
- 2. Médecin** : Ce rôle est important pour notre projet, car les médecins seront responsables de la saisie des symptômes présentés par le patient, pour permettre à l'IA de faire un diagnostic.
- 3. Patient** : c'est l'utilisateur final et la raison d'être de notre projet de gestion d'hôpital. Ils ont la possibilité d'accéder à leur dossier médical et de donner leur opinion.

2.2.2 Besoins fonctionnels

La plateforme offre diverses fonctionnalités afin de satisfaire ses acteurs

Pour le receptioniste La plateforme doit offrir la possibilité de :

1.S'enregistrer en créant un compte, puis de se connecter en saisissant leur email et leur mot de passe.

2.Gérer un medecin :dispose de la possibilité d'ajouter,supprimer, consulter et de modifier les données d'un médecin.

3.Gérer un patient : dispose de la possibilité d'ajouter,supprimer,consulter et de modifier Les données d'un patient.

4.Planifier et organiser des rendez-vous pour les patients.

5.Consulter et éditer les rendez-vous préalablement fixés.

Pour le médecin

La plateforme doit offrir au médecin la possibilité de :

1.S'authentifier.

2.Visualiser la liste des patients ajoutés par la réceptionniste.

3.Saisir les symptômes présentés par le patient.

4.Visualiser l'historique médical des patient.

Pour le patient

La plateforme doit offrir au patient la possibilité de :

1. Donner son avis.

2.2.3 Besoins non fonctionnels

Notre solution doit également répondre aux containtes de qualités suivants pour assurer une meilleure interaction avec les utilisateurs et le bon fonctionnement du systeme.

Facilité d'utilisation :La conception de l'interface utilisateur doit etre facile, intuitive et conviviale pour les utilisateurs.

Sécurité :l'application doit garantir la confidentialité des informations sensible des patients et de leur santé.

Robustesse :Assurer une disponibilité de manière continue et sans interruption des services

2.3 Spécification des besoins

Diagramme de cas d'utilisation

Dans cette section, nous utilisons des diagrammes de cas d'utilisation pour décrire les interactions entre les acteurs et le système.

La figure 2.1 représente le diagramme de cas d'utilisation du réceptionniste, qui inclut les fonctionnalités suivantes : la gestion des comptes, la consultation du tableau de bord et la gestion des rendez-vous pour les patients.

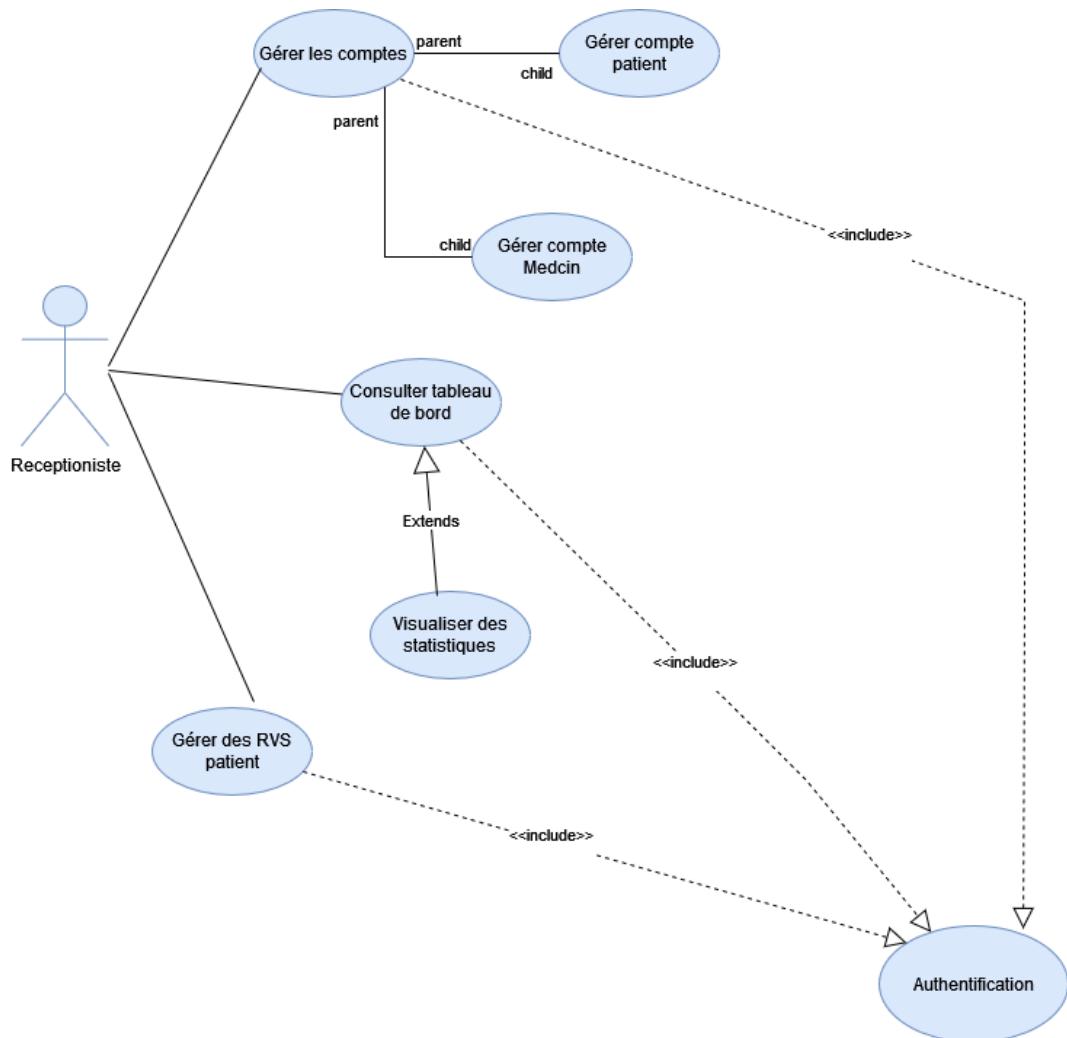


FIGURE 2.1 – Diagramme de cas d'utilisation réceptioniste

La figure 2.2 représente le diagramme de cas d'utilisation du médecin, qui comprend les fonctionnalités suivantes : consulter la liste des patients afin de leur ajouter des symptômes et visualiser leur historique.

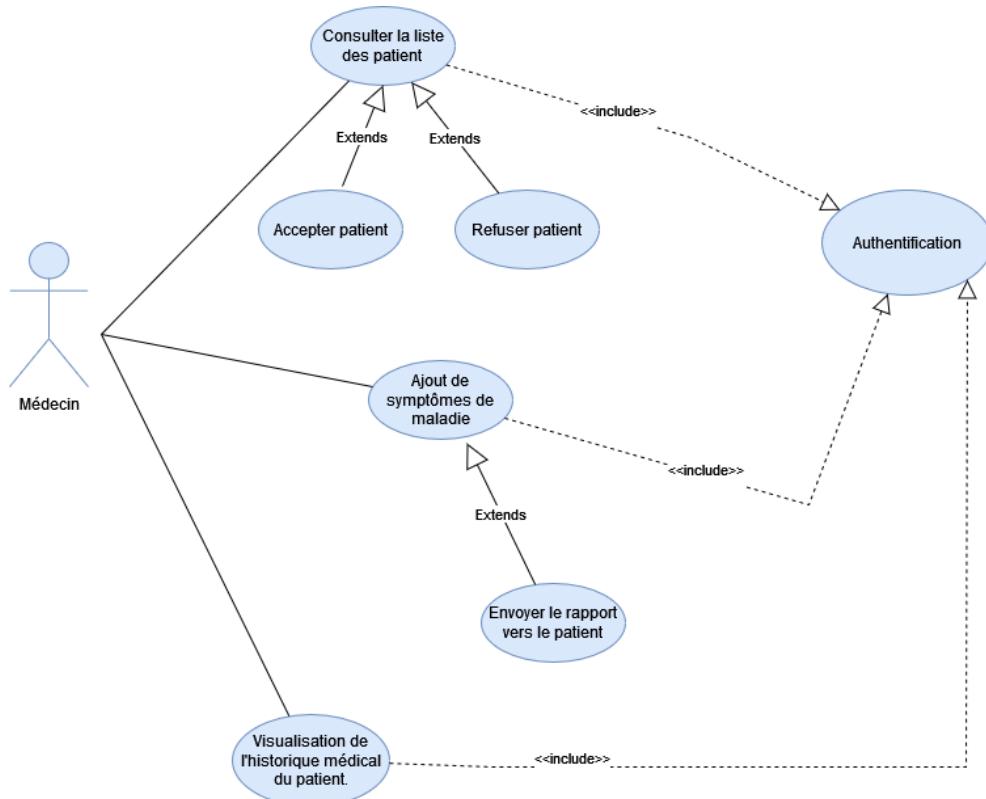


FIGURE 2.2 – Diagramme de cas d'utilisation médecin

La figure 2.3 représente le diagramme de cas d'utilisation du patient, qui a la possibilité de donner son avis dans la plateforme



FIGURE 2.3 – Diagramme de cas d'utilisation patient

Description des cas d'utilisation :

Authentication : L'étape d'authentification est critique pour la sécurité de notre plateforme, et nous avons donc mis à disposition un tableau descriptif et un diagramme de séquence détaillant les différentes étapes de ce processus. (Voir annexe)

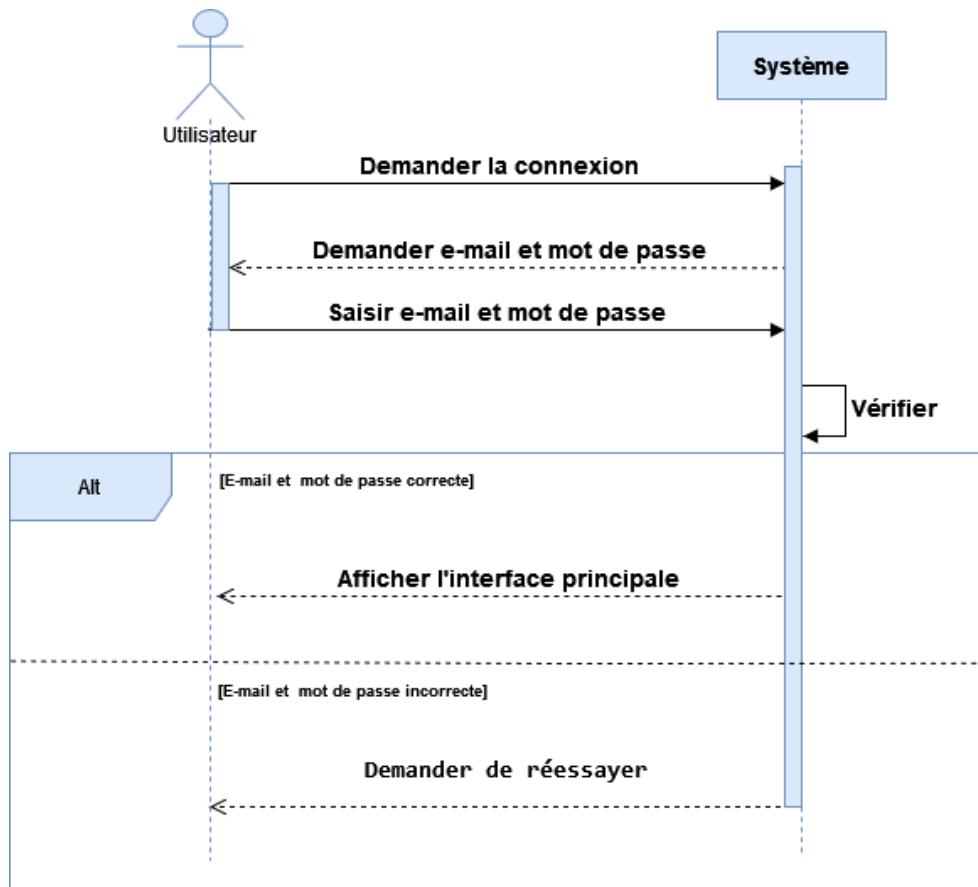


FIGURE 2.4 – Diagramme de séquence Authentification

Gérer les comptes : Pour mieux comprendre la fonctionnalité de gestion des comptes sur notre plateforme, nous avons créé un diagramme de séquence qui illustre les interactions entre le système et le réceptionniste pour la gestion des comptes.

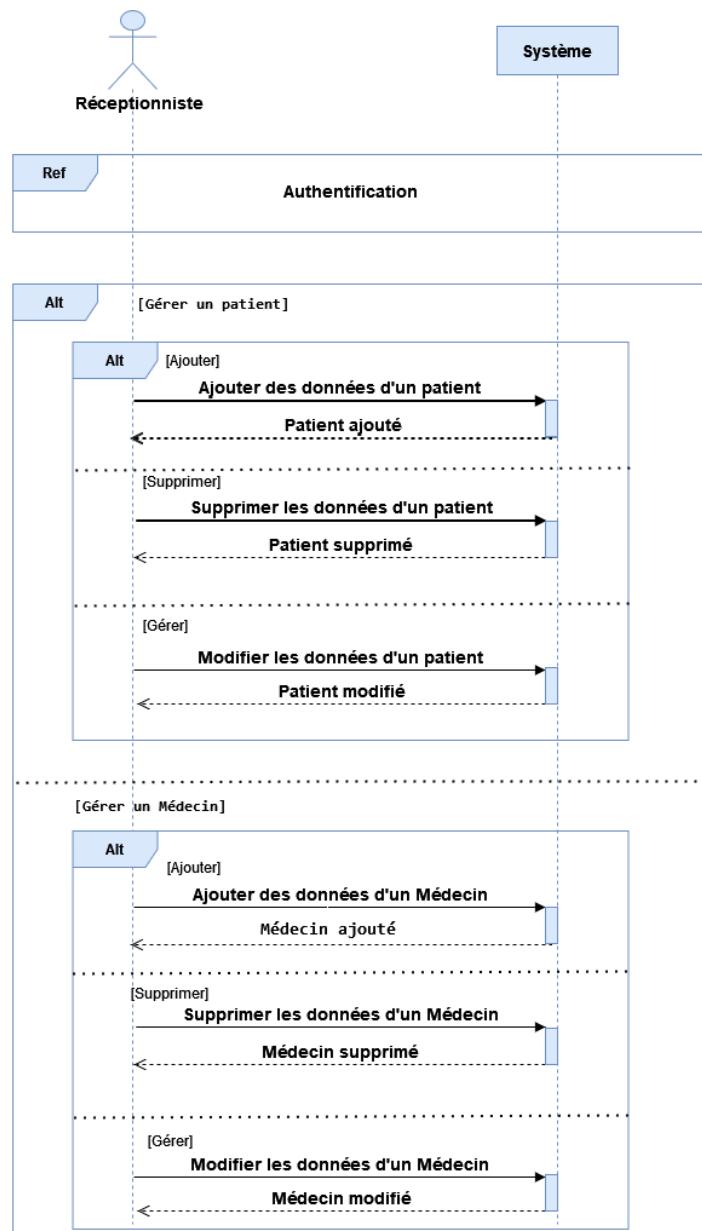


FIGURE 2.5 – Diagramme de séquence Gérer les comptes

Gérer les rendez vous : Pour mieux comprendre la fonctionnalité de gestion des rendez-vous sur notre plateforme, nous avons créé un tableau descriptif et un diagramme d'activité détaillant les différentes étapes à suivre pour gérer efficacement les rendez-vous. (Voir annexe)

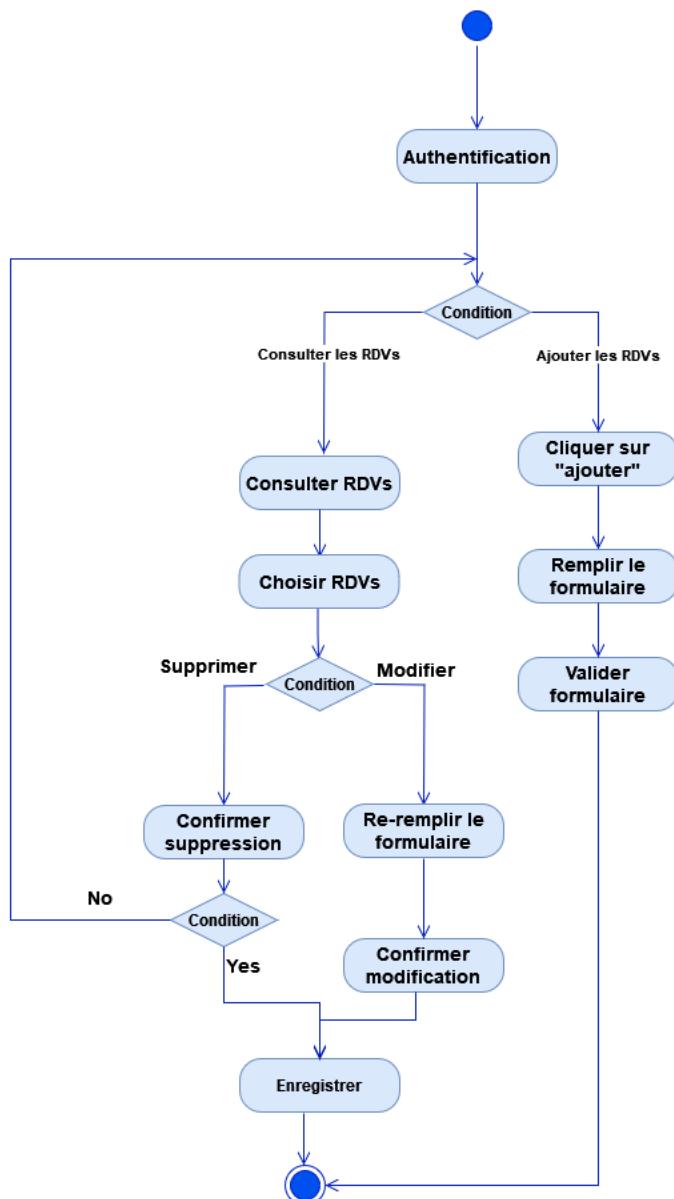


FIGURE 2.6 – Diagramme d'activité gérer les comptes

Ajouter les symptômes : Nous avons décrit les étapes que le médecin doit suivre pour ajouter les symptômes d'un patient dans notre système en utilisant un tableau descriptif. En outre, nous avons créé un diagramme de séquence illustrant la communication entre le front-end, le back-end et le modèle d'IA lors de cette opération.

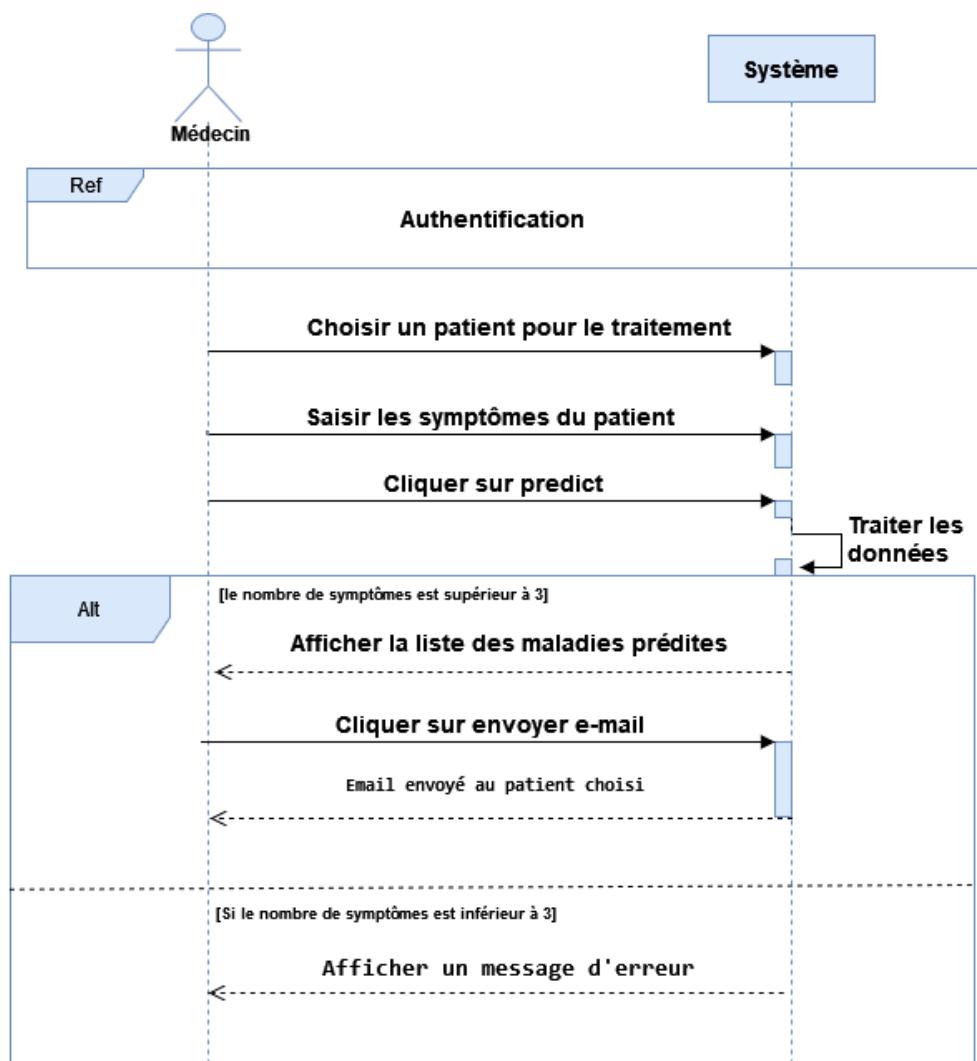


FIGURE 2.7 – Diagramme de séquence ajouter symptomes

2.4 Conclusion

Dans ce chapitre, nous avons analysé les besoins fonctionnels et non fonctionnels par acteur pour concevoir et réaliser les fonctionnalités et services de notre plateforme. Nous avons également créé des diagrammes pour mettre en évidence les interactions nécessaires à la compréhension de l'application.

Chapitre 3

Conception

3.1 Introduction

Dans ce chapitre, nous avons également divisé la conception en deux parties clés. La première partie se concentre sur l'architecture du système, en détaillant l'architecture physique et logique, ainsi que le diagramme de paquetages pour visualiser les composants du système. La seconde partie se concentre sur l'apprentissage automatique, en présentant les modèles que nous avons utilisés ainsi que les métriques que nous avons appliqués pour évaluer ces modèles.

3.2 Conception architecturale du système

3.2.1 Architecture physique

Pour la mise en oeuvre de notre application, nous nous allons la déployer sur une architecture client serveurs dont les composants sont les suivants :

Client léger : C'est l'interface qui permet à l'utilisateur d'accéder à l'application sans installation de logiciel supplémentaire pour l'y accéder.

Serveur Web : Est responsable de gérer les requêtes et les réponses entre les clients et l'application, y compris la base de données. Il est chargé de récupérer et de stocker des données pour construire la logique métier et faciliter la communication entre les parties prenantes. En tant que tel, le serveur web joue un rôle central dans l'élaboration de stratégies métier efficaces, telles que la gestion des données, l'optimisation des performances et la sécurité des applications web.

Serveur de base de données : C'est Le quatrième niveau de notre architecture, Cette couche est responsable de stocker et de récupérer les données en assurant leur sécurité et leur intégrité.

Nous avons opté pour ce style architectural pour les raisons suivantes :

Séparation des tâches : La division des tâches entre les différents niveaux de l'application permet de séparer les responsabilités et les fonctionnalités, ce qui facilite le développement et la maintenance de l'application.

Scalabilité : La scalabilité d'une architecture physique à 4 niveaux correspond à sa capacité à évoluer et à s'adapter aux charges de travail croissantes ou diminuantes en permettant l'ajout ou la suppression des ressources matérielles telles que des serveurs ou des espaces de stockage.

Maintenance : L'architecture 4 tiers facilite la maintenance du système en permettant une gestion plus facile des mises à jour et des correctifs.

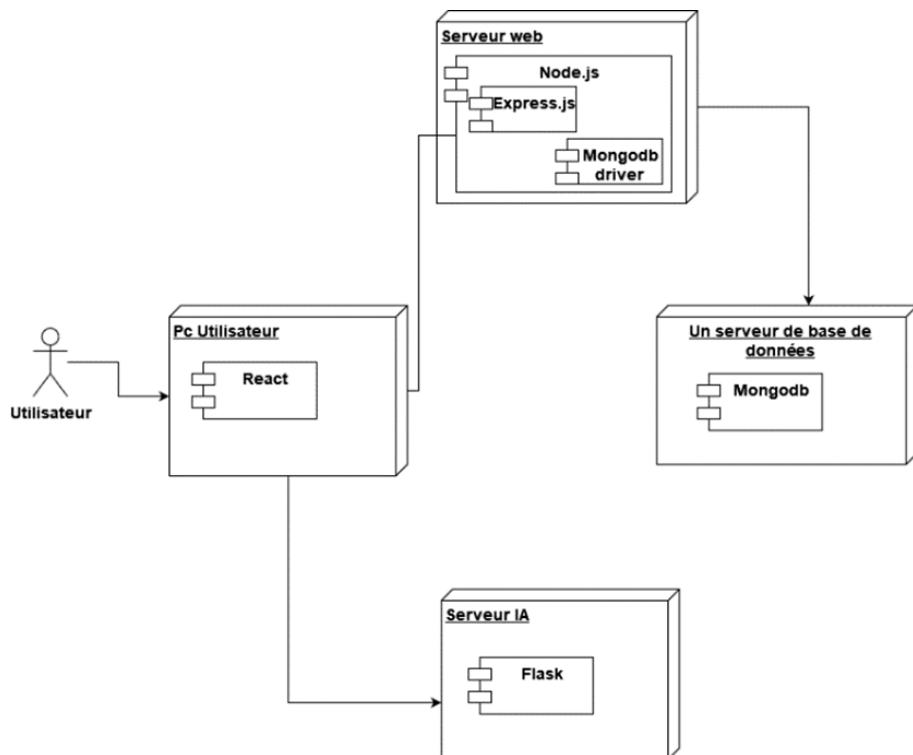


FIGURE 3.1 – Diagramme de déploiement [8]

3.2.2 Architecture logique

Front-End Hooks

l'architecture logique des Hooks facilite l'organisation du code en blocs de fonctionnalités modulaires et indépendants cela permet de faciliter la distinction entre la logique de présentation et la logique fonctionnelle dans le but d'améliorer la lisibilité et la facilité de compréhension du code.

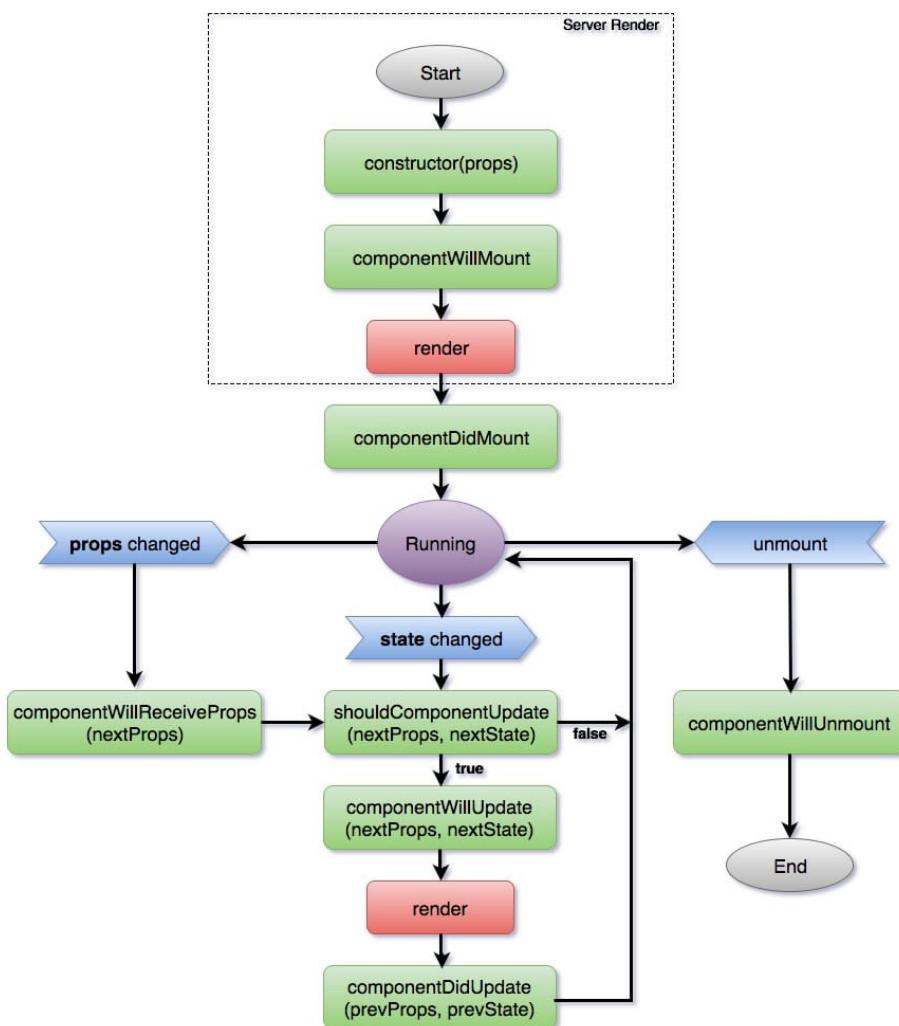


FIGURE 3.2 – Architecture du Frontend [13]

Back-End

Le BackEnd est la partie invisible de l'application qui traite et gère les données et les fonctionnalités de l'application en arrière-plan et Les utilisateurs n'ont pas accès directe au BackEnd. On a opté pour une architecture à trois couches

Les contrôleurs : recevoir les requêtes qui proviennent du FrontEnd et de les acheminer vers la couche métier pour les traiter.

La couche métier : contient la logique principale de l'application et elle est responsable de la gestion des processus métier nécessaires pour satisfaire les besoins des utilisateurs.

La couche d'accès aux données : charge d'extraction et du traitement des données stockées dans la base de données.



FIGURE 3.3 – Architecture du Backend

Modèle IA pipeline

L'architecture en pipeline est une Suite et enchaînement d'actions qui sert à formater et transformer des données en données traitées et utiles. On a utilisé cette logique pour organiser, structurer et faciliter le processus de traitement. Nous avons décidé d'utiliser cette structure en 4 étapes pour garantir la mise en place d'application déployé.

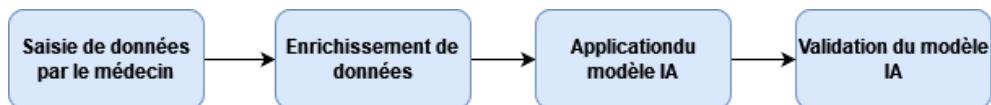


FIGURE 3.4 – Architecture du modèle ai

La saisie de données : cette étape consiste à entrer les différentes informations requises en les séparant par des virgules pour les distinguer les unes des autres.

Enrichissement de données : L'enrichissement des données consiste à ajouter des synonymes et des termes associés à la liste initiale d'entrées en utilisant des outils tels que WordNet.

Application : Le modèle doit être entraîné. Cela implique l'utilisation d'un algorithme d'apprentissage automatique pour analyser les données d'entrée.

Visualisation des résultats : Obtenir les résultats triés par ordre de probabilité des maladies les plus probables.

3.3 Prétraitemenet des données

La création d'un modèle de machine learning est un processus composé de plusieurs étapes précises et bien définies, à savoir le pré-traitement des données. Cette tâche d'opération est considérée comme la tâche la plus critique de ce processus car elle met l'accent sur les données afin de les traiter et de les convertir sous la forme la plus adéquate. Cela se fait en réduisant leur taille ainsi qu'en trouvant des relations entre eux, en les normalisant et en supprimant les valeurs aberrantes. Celui-ci intègre plusieurs technologies, à savoir la transformation et la réduction des données afin de sélectionner le modèle le plus adapté en améliorant la qualité des données.

3.4 Les algorithmes de l'apprentissage automatique utilisés

3.4.1 Random Forest (forêts aléatoires)[14] :

L'algorithme d'apprentissage supervisé Random Forest, également connu sous le nom de forêts aléatoires, il combine plusieurs techniques d'apprentissage pour résoudre des problèmes complexes et améliorer les performances du modèle. C'est un algorithme qui fusionne les résultats de plusieurs arbres de décision pour obtenir un résultat unique. Sa facilité d'utilisation a renforcé son adoption car il est capable de résoudre à la fois des problèmes de classification et de regression.

Les avantages :

-Haute précision : Les Random Forest offrent une grande précision en raison de leur capacité à réduire le surapprentissage (overfitting) et à gérer les données manquantes et bruyantes.

-Robustesse aux données manquantes : Les Random Forest peuvent traiter des ensembles de données avec des valeurs manquantes sans avoir besoin de remplacer les données manquantes.

-Robustesse aux valeurs aberrantes : Les Random Forest peuvent gérer des valeurs aberrantes (outliers) et des valeurs extrêmes (extremes).

-Adaptabilité aux grands ensembles de données : Les Random Forest peuvent traiter de grands ensembles de données avec de nombreux attributs.

-Facilité de mise en œuvre : Les Random Forest sont relativement faciles à mettre en œuvre.

Les inconvénients :

- Manque d'interprétabilité : Les RF sont souvent considérés comme des "boîtes noires" car il est difficile d'interpréter les décisions prises par l'algorithme.
- Temps de traitement plus long : Les RF peuvent prendre plus de temps pour s'entraîner que certains autres algorithmes d'apprentissage automatique.
- Besoin de données normalisées : Les RF fonctionnent mieux lorsque les données sont normalisées, ce qui peut nécessiter un prétraitement supplémentaire des données.
- Risque de sur-ajustement : Comme pour tous les modèles d'apprentissage automatique, les RF peuvent sur-ajuster les données d'entraînement, ce qui peut entraîner une mauvaise performance sur de nouvelles données.

Les paramètres du modèle RF :

Dans ce tableau, nous présentons les différents paramètres qui ont été ajustés pour le modèle de forêt aléatoire

Paramètre	définition
n-estimators	entier, optionnel (par défaut=100) : Le nombre d'estimateurs dans la forêt.
max-depth	entier, (par défaut=None) La limite de profondeur de l'arbre.

TABLE 3.1 – Tableau de paramètres RF.

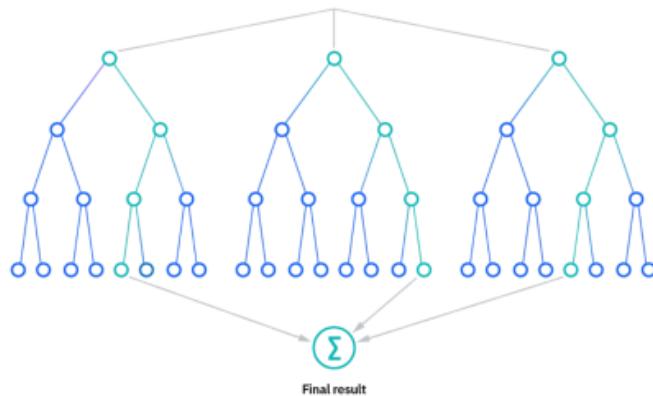


FIGURE 3.5 – Random Forest Modèle

3.4.2 K nearest neighbors (KNN)[15] :

L'algorithme des k-plus proches voisins, également connu sous le nom de KNN ou k-NN est l'un des algorithmes d'apprentissage supervisé les plus simples, utilisé pour résoudre

des problèmes de classification et de régression. Son mode de fonctionnement consiste à classifier les nouveaux points de données en se basant sur la similitude avec les points de données voisins, le KNN classe les nouveaux points de données.

Les avantages :

- Sa simplicité et sa facilité d'implémentation.
- Son efficacité pour les données non linéaires.
- L'absence de paramétrage complexe nécessaire.
- Sa considération comme un algorithme d'apprentissage "paresseux", car il ne nécessite pas de formation explicite du modèle, ce qui permet de l'utiliser facilement sur des données en constante évolution.

Les inconvénients :

- Sensibilité aux données aberrantes
- Temps de calcul plus longs pour les ensembles de données volumineux
- Affecté par la présence de variables bruyantes et de redondance dans les données
- Le choix de la valeur K peut être un défi et doit être effectué avec soin pour obtenir des résultats optimaux.

Les paramètres du modèle KNN :

Dans ce tableau, nous présentons les différents paramètres qui ont été ajustés pour le modèle de forêt aléatoire

Paramètre	définition
n-neighbors	entier , (par défaut =5) Nombre de voisins utilisés pour les requêtes kneighbors.
weights	‘uniform’, ‘distance’, (par défaut =’uniform’) Fonction de pondération utilisée pour la prédiction.

TABLE 3.2 – Tableau de paramètres KNN.

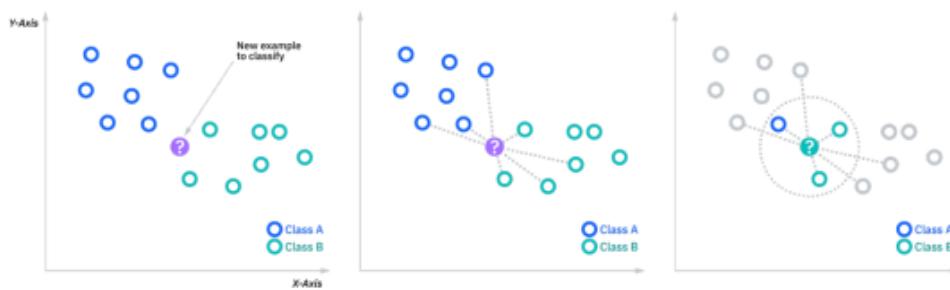


FIGURE 3.6 – KNN modèle

3.4.3 Régression Logistique [16] :

La régression logistique, également connue sous le nom de modèle logit, est méthode de classification et d'analyse prédictive populaire basée sur des statistiques. Elle est utilisée pour réaliser des prévisions de la probabilité d'occurrence d'un événement, la classification consistant à déterminer si une personne a voté ou non en fonction d'un ensemble de variables indépendantes. Comme la sortie est une probabilité, la variable dépendante est bornée entre 0 et 1.

Les avantages :

- La régression logistique est facile à comprendre et à interpréter, même pour les non-spécialistes des statistiques.
- Elle est également facile à mettre en œuvre dans la plupart des langages de programmation, notamment grâce aux packages de logiciels tels que Scikit-Learn et Statsmodels pour Python, ou encore le package glm pour R.
- Elle fournit une estimation de la probabilité d'appartenance à chaque classe, ce qui peut être utile dans diverses applications, telles que la prédiction de la probabilité de défaut de paiement d'un prêt hypothécaire.
- La régression logistique peut être étendue pour inclure des variables continues et catégorielles dans le modèle.

Les inconvénients :

- La régression logistique suppose une relation linéaire entre les variables explicatives et la variable dépendante, ce qui peut ne pas être valide dans certains cas.
- Elle peut également être sensible à la présence de valeurs aberrantes et de données manquantes.
- Comme pour la plupart des méthodes de régression, il est important de faire attention à la multicollinéarité entre les variables explicatives, qui peut entraîner des estimations biaisées et des intervalles de confiance incorrects.

Les paramètres du modèle RL :

Dans ce tableau, nous présentons les différents paramètres qui ont été ajustés pour le modèle de forêt aléatoire

Paramètre	définition
penalty	'l1', 'l2', 'elasticnet', None (par défaut ='l2') Préciser la norme de la pénalité.
C	float(par défaut =1.0) est un paramètre essentiel qui permet de gérer la puissance de la régularisation dans un modèle.

TABLE 3.3 – Tableau de paramètres Régression Logistique.

3.4.4 Naïve bayésienne multinomiale [17] :

une méthode d'apprentissage probabiliste qui est principalement utilisée en traitement du langage naturel (NLP), l'algorithme se fonde sur le théorème de Bayes pour prédire l'étiquette d'un texte, tel au'un courriel ou un article de presse. L'algorithme calcule la probabilité de chaque étiquette pour un exemple donné, puis retourne l'étiquette ayant la plus forte probabilité. Ce classificateur est une famille d'algorithmes qui partagent un même principe de base , ce principe repose sur l'hypothèse que chaque caractéristique utilisée pour la classification est indépendante des autres caractéristiques.

Les avantages :

- Faible coût de calcul : Le modèle est rapide à entraîner et à prédire, même pour des ensembles de données massifs.
- Bonne performance avec des ensembles de données de petite et moyenne taille : Les algorithmes Naïve Bayes ont tendance à fonctionner bien avec des ensembles de données de petite et moyenne taille.
- Facilité de mise en œuvre : facilité à mettre en œuvre et à comprendre, ce qui le rend accessible même pour les débutants. Le modèle fonctionne bien même si le nombre de caractéristiques est élevé.

Les inconvénients :

- Assumption de l'indépendance : Le modèle Naïve Bayes suppose que toutes les fonctionnalités sont indépendantes les unes des autres, ce qui peut ne pas être vrai dans la pratique.
- Sensibilité aux données manquantes : Le modèle peut être sensible aux données manquantes car il suppose que toutes les fonctionnalités sont présentes dans chaque instance.

-Limitations de classification : Le modèle peut avoir des difficultés à classer des cas qui n'ont pas été observés dans l'ensemble de données d'entraînement.

Les paramètres du modèle MNB :

Dans ce tableau, nous présentons les différents paramètres qui ont été ajustés pour le modèle de forêt aléatoire

Paramètre	définition
fit prior	Boolean Option pour déterminer si les probabilités a priori de chaque classe doivent être apprises (True) ou si une probabilité uniforme doit être utilisée (False).
alpha	réel, correspond au facteur de lissage additive est utilisé pour régulariser les probabilités des classes et prévenir le surapprentissage du modèle.

TABLE 3.4 – Tableau de paramètres MNB.

3.4.5 Les Métriques et les algorithmes d'évaluation

Précision (Accuracy)[18] : La précision est une métrique qui permet de quantifier la proportion de prédictions positives correctes par rapport à l'ensemble des prédictions positives réalisées plus la précision est haute, moins le modèle de Machine Learning se trompe en donnant des résultats positifs.

$$precision = \frac{Vrai\ Positif}{Vrai\ Positif + Faux\ Positif}$$

FIGURE 3.7 – Formule de précision

Cross validation[19] : L'une des techniques utilisées pour tester l'efficacité d'un modèle de Machine Learning, la Cross-Validation est une technique couramment utilisée en Machine Learning pour comparer plusieurs modèles et identifier celui qui convient le mieux à un problème donné.

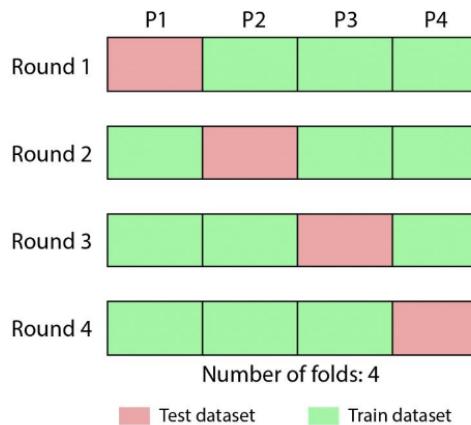


FIGURE 3.8 – L’algorithme de validation croisée

3.5 Conclusion

Au cours de ce chapitre, nous avons posé les bases pour la phase de réalisation à venir. Nous avons présenté une vue d’ensemble détaillée de l’architecture logique et physique sous-jacente à notre système, ainsi que les différents algorithmes d’apprentissage automatique utilisés, en mettant l’accent sur l’importance des métriques pour choisir le modèle adéquat. Enfin, nous avons également partagé des maquettes pour donner une idée concrète de la façon dont nous envisageons de commencer le processus de développement.

Chapitre 4

Realisation

4.1 Introduction

Dans ce chapitre, nous allons plonger dans les coulisses du développement de notre solution, pour découvrir les secrets de sa mise en œuvre. Vous pourrez suivre pas à pas les différentes étapes de réalisation, du choix des technologies à l'implémentation concrète du projet, dans ce chapitre, nous allons fournir une description détaillée des différentes étapes de développement, des choix de technologies et d'implémentation nous allons également inclure des exemples de code et des captures d'écran pour mieux illustrer le fonctionnement de la solution.

4.2 Environnements de travail

4.2.1 Environnements de developpement materiel

Poste n°1 : ASUS TUF A17

Processor : AMD Ryzen 7 4800H with Radeon Graphics 2.90 GHz

Graphics : NVIDIA GeForce RTX 3050 Laptop GPU GDDR6 @ 4GB (128 bits)

Memory : Total system memory :16GB

Poste n°2 : ASUS X550V

Processor : Intel® Core™ i5-10300H Processor 2.4 GHz (8M Cache, up to 4.1 GHz, 4 cores)

Graphics : Intel® UHD Graphics 630, NVIDIA® GeForce® GTX 1650, 4GB GDDR5

Memory : Total system memory :16GB

4.2.2 Environnements de développement logiciel

Python : Python est un langage de programmation très apprécié des développeurs pour sa simplicité, sa lisibilité, sa flexibilité et sa bibliothèque standard complète. Il est multiplateforme, dispose de bibliothèques populaires pour l'analyse de données et l'apprentissage automatique, et bénéficie d'une communauté active et dynamique qui fournit de nombreux outils et ressources pour améliorer le code.

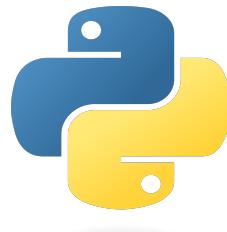


FIGURE 4.1 – Python

Pandas : Pandas est une bibliothèque open-source qui permet la manipulation et l'analyse de données en Python. Elle est performante, flexible et facile à utiliser, et permet de charger, manipuler et fusionner des données efficacement. Pandas est un outil de haut niveau pour l'analyse en Python, et son nom vient de "Panel Data", ce qui désigne les ensembles de données incluant des observations sur de multiples périodes temporelles. Pandas facilite le travail des scientifiques de données et des développeurs.



FIGURE 4.2 – Pandas

Jupyter : Jupyter est un environnement de développement interactif qui présente de nombreux avantages pour les développeurs, tels que la possibilité d'écrire, d'exécuter et de visualiser du code dans un même environnement, la compatibilité avec plusieurs langages de programmation, des extensions et widgets pour ajouter des fonctionnalités supplémentaires, et une communauté active de soutien. Jupyter est un outil puissant et flexible pour les développeurs souhaitant créer du code interactif et reproductible.



FIGURE 4.3 – Jupyter

Flask : Flask est un micro-framework web Python qui offre de nombreux avantages, notamment sa simplicité, sa flexibilité et sa modularité. Il est facile à apprendre et à utiliser, permettant de créer des applications web personnalisées et évolutives en fonction des besoins spécifiques des développeurs. Flask est également compatible avec une grande variété d'outils et de bibliothèques, ce qui facilite l'intégration avec d'autres technologies. Enfin, Flask est open-source et dispose d'une communauté active qui fournit un soutien continu et des mises à jour régulières, en faisant un choix judicieux pour les développeurs cherchant à créer des applications web simples et flexibles en Python.



FIGURE 4.4 – Flask

Node.js : Node.js est un environnement d'exécution JavaScript côté serveur qui permet aux développeurs de créer des applications hautement évolutives et performantes en utilisant une seule langue de programmation. Il est basé sur un modèle d'entrée/sortie non bloquant, ce qui permet une gestion efficace des connexions multiples et une meilleure utilisation des ressources. Node.js dispose également d'une grande bibliothèque de modules et d'outils, est open-source et dispose d'une communauté active pour un soutien continu et des mises à jour régulières. En somme, Node.js est un choix judicieux pour les développeurs qui cherchent à créer des applications web performantes et évolutives en utilisant JavaScript sur le côté serveur.



FIGURE 4.5 – NodeJS

React Js : React.js est une bibliothèque JavaScript pour la création d'interfaces utilisateur (UI). Elle utilise des composants réutilisables pour faciliter la création d'interfaces utilisateur complexes. Elle optimise les performances en ne mettant à jour que les parties de l'interface qui ont changé. Elle est compatible avec une variété d'outils et est open-source avec une communauté active. En somme, elle est idéale pour la création d'interfaces utilisateur réutilisables, performantes et évolutives en utilisant JavaScript.

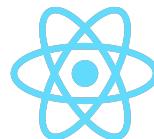


FIGURE 4.6 – React Js

Express Js : Express.js est un framework web JavaScript simple, flexible et modulaire utilisé pour créer des applications web côté serveur rapidement. Il utilise des fonctions middleware pour traiter les requêtes et les réponses. Il est compatible avec de nombreux composants tels que les moteurs de templates et les outils de gestion de sessions et dispose d'une grande variété d'outils et de bibliothèques pour une intégration facile avec d'autres technologies. Enfin, il est open-source et dispose d'une communauté active fournissant un soutien continu et des mises à jour régulières.



FIGURE 4.7 – Express Js

MongoDB : MongoDB est une base de données NoSQL appréciée pour sa capacité à stocker et gérer des données semi-structurées ou non structurées. Elle utilise une structure de données souple, appelée BSON, pour stocker les données dans des documents semblables à des fichiers JSON. MongoDB est réputée pour sa capacité à gérer de gros volumes de données avec une scalabilité horizontale. Elle est également compatible avec de nombreux langages de programmation et possède une communauté active open source qui fournit un support continu.



FIGURE 4.8 – Mongo db

4.3 Élaboration et mise au point des modèles

4.3.1 Préparation des données

Ensemble de données utilisé

Le jeu de données sur les maladies et leurs symptômes a été obtenu à partir de deux sources : une partie manuellement à partir du site <https://www.nhs.uk/conditions/> et une autre partie depuis l'ensemble de données disponible sur Kaggle à <https://www.kaggle.com/datasets/itachi9604/disease-symptom-description-dataset>.

Nettoyage des données

Dans cette étape de prétraitement de données, nous avons effectué une série d'opérations complexes pour nettoyer et normaliser les symptômes, supprimer les données redondantes, transformer la forme de la base de données et enfin la rendre binaire. Les travaux ont été effectués consciencieusement et avec rigueur, afin d'assurer la qualité et la cohérence des données pour les prochaines étapes de l'analyse.

```
: import pandas as pd
import re
import unicodedata

# Chargement du fichier CSV
df = pd.read_csv('before.csv')

# Suppression des liens dans les symptômes
df['Disease'] = df['Disease'].apply(lambda x: re.sub(r'http\S+', '', x))

# Suppression des caractères spéciaux dans les symptômes (sauf virgules et un seul espace)
df['Disease'] = df['Disease'].apply(lambda x: re.sub(r'[^\\w\\s,]|(?<! ) (?= )', '', x))

# Conversion des textes en majuscule en minuscule
df = df.apply(lambda x: x[1:].str.lower())
# Normalisation des mots dans les symptômes
df['Symptoms'] = df['Symptoms'].apply(lambda x: unicodedata.normalize('NFKC', x))

# Export du fichier nettoyé en CSV
df.to_csv('data_nettoyé.csv', index=False)
```

FIGURE 4.9 – Code du nettoyage des données

Nous avons essayé de transformer notre base de données en plaçant les maladies dans la première colonne et les symptômes dans la première ligne, en remplissant chaque cellule avec une valeur binaire de 0 ou 1 pour indiquer la présence ou l'absence de chaque symptôme pour chaque maladie



FIGURE 4.10 – Code du nettoyage des données

Nombre de Maladie	261
Nombre de Symptomes	489

TABLE 4.1 – Paramètre de la base de données

4.3.2 Paramétrage et entraînement des modèles

Lorsqu'on utilise des modèles d'apprentissage automatique pour résoudre des problèmes de classification ou de régression, il est important de choisir le modèle le plus adapté à la

tâche à accomplir. Cependant, une fois que le modèle est choisi, il est également important d'ajuster les paramètres du modèle pour améliorer ses performances. Après avoir identifié les paramètres qui ont une influence significative sur les performances du modèle, nous avons utilisé la méthode de la recherche en grille (grid search) pour tester plusieurs valeurs de paramètres afin de visualiser les résultats à l'aide de graphiques.

Grid search : est une méthode courante d'optimisation des hyperparamètres en machine learning elle implique de spécifier une liste de valeurs pour chaque hyperparamètre à optimiser. Ensuite, pour chaque combinaison dans la grille, le modèle est entraîné et évalué.

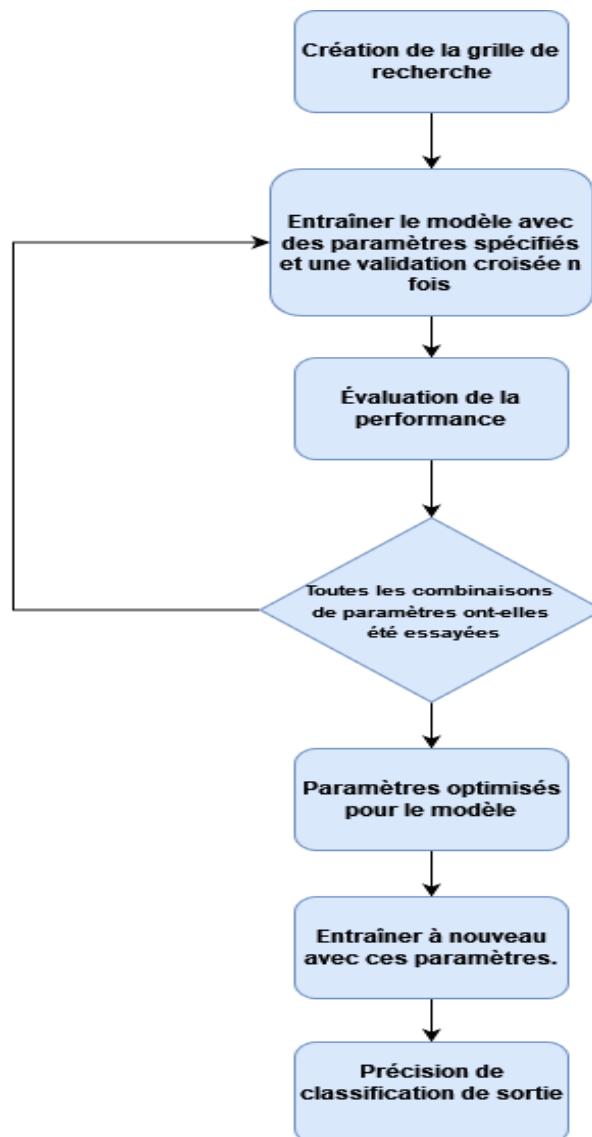


FIGURE 4.11 – Diagramme d'activité GRID Search

```
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# define the parameter grid to search over
param_grid = {
    'alpha': np.arange(0.1, 1.0, 0.1),
    'fit_prior': [True, False],
}

# create the GridSearchCV object
mnb_grid = GridSearchCV(MultinomialNB(), param_grid, cv=5)

# fit the grid to the data
mnb_grid.fit(x_train, y_train)

# get the best estimator and its parameters
best_mnb = mnb_grid.best_estimator_
best_params = mnb_grid.best_params_

# print the best parameters
print(f"Best parameters: {best_params}")

# create a grid of alpha values and fit_prior values
alpha_vals = np.arange(0.1, 1.0, 0.1)
fit_prior_vals = [True, False]
grid = np.zeros((len(alpha_vals), len(fit_prior_vals)))

# Loop over the alpha and fit_prior values and calculate the cross-validation score
for i, alpha in enumerate(alpha_vals):
    for j, fit_prior in enumerate(fit_prior_vals):
        mnb = MultinomialNB(alpha=alpha, fit_prior=fit_prior)
        scores_mnb = cross_val_score(mnb, X, Y, cv=5)
        score = scores_mnb.mean()
        grid[i, j] = score

# plot the grid as a heatmap
sns.heatmap(grid, xticklabels=fit_prior_vals, yticklabels=alpha_vals, annot=True, fmt='%.2f')
plt.xlabel("fit_prior")
plt.ylabel("alpha")
plt.title("Cross Validation Accuracy Grid")
plt.show()
```

FIGURE 4.12 – Utilisation de Grid Search dans le modèle MNB

Random Forest (forêts aléatoires) :

Nous avons exploré différents paramètres pour affiner notre modèle de Random Forest, en mettant l'accent sur ceux qui ont le plus d'impact, tels que le nombre d'estimateurs (n-est-vals) et la profondeur maximale (max-depth-vals). Les résultats de nos expériences sont présentés dans une figure captivante, qui met en évidence les différences significatives entre les valeurs de la validation croisée.

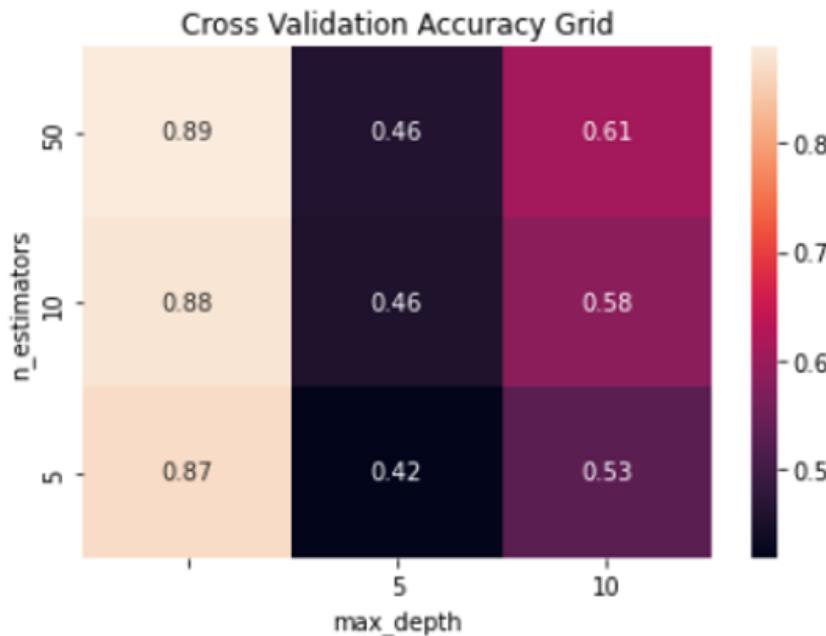


FIGURE 4.13 – La visualisation des différentes valeurs de cross validation pour RF

Grâce à cette visualisation, nous sommes maintenant en mesure de choisir les valeurs de paramètres les plus appropriées, les réglages qui donnent lieu aux résultats de validation croisée les plus satisfaisants.

```
Best parameters: {'max_depth': None, 'n_estimators': 50}
```

FIGURE 4.14 – Les paramètres optimales pour RF

K nearest neighbors (KNN) :

Nous avons procédé à une investigation de divers paramètres afin d'optimiser notre modèle KNN, en nous concentrant en particulier sur ceux qui ont le plus d'influence, tels que "weights" et "n-neighbors". Les résultats de nos expérimentations sont illustrés dans une figure captivante qui met en évidence les écarts significatifs entre les différentes valeurs obtenues lors de la validation croisée.

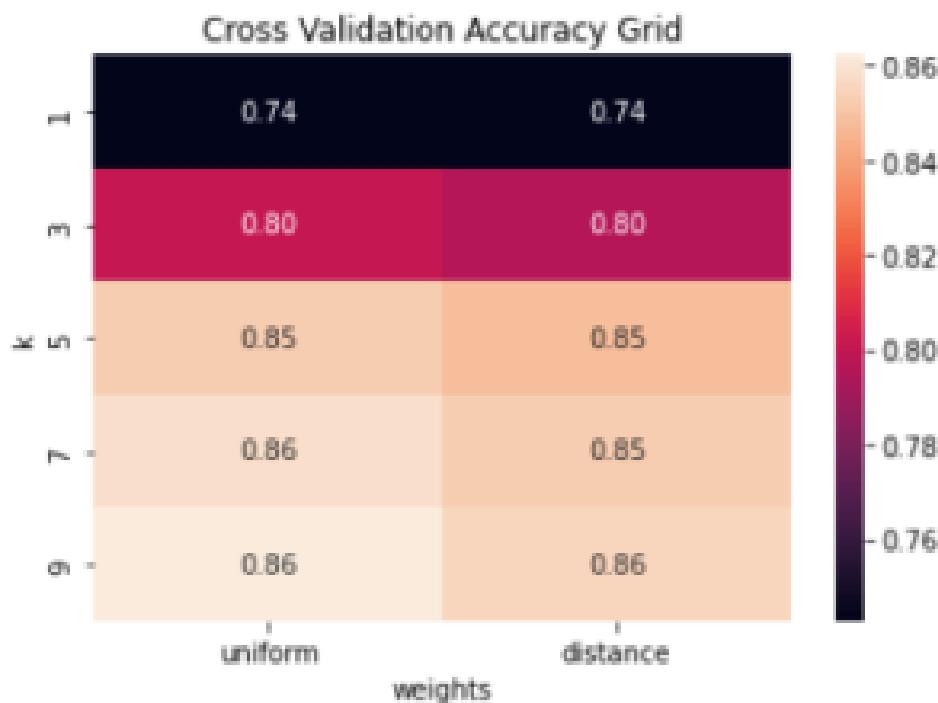


FIGURE 4.15 – La visualisation des différentes valeurs de cross validation pour KNN

Cette visualisation nous a permis de sélectionner les valeurs de paramètres les plus adéquates, c'est-à-dire celles qui ont donné lieu aux résultats de validation croisée les plus satisfaisants. Nous sommes donc désormais en mesure de choisir les réglages optimaux pour notre modèle KNN .

```
Best parameters: {'n_neighbors': 9, 'weights': 'uniform'}
```

FIGURE 4.16 – Les paramètres optimales pour KNN

Régression Logistique :

Nous avons effectué une exploration de différents paramètres pour optimiser notre modèle de régression logistique, en nous concentrant notamment sur les paramètres ayant le plus d'influence, tels que "C" et "penalty". Les résultats de nos expériences ont été présentés sous la forme d'une figure attractive qui met en évidence les variations significatives entre les différentes valeurs obtenues lors de la validation croisée.

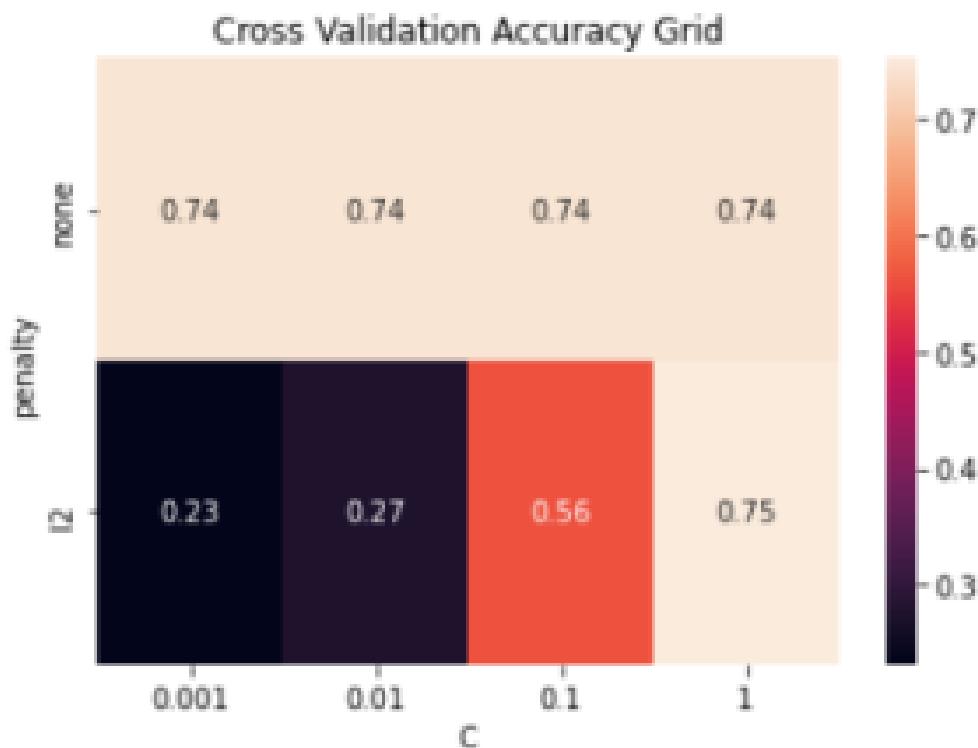


FIGURE 4.17 – La visualisation des différentes valeurs de cross validation pour RL

Grâce à cette visualisation, nous sommes maintenant en mesure de choisir les valeurs de paramètres les plus appropriées, les réglages qui donnent lieu aux résultats de validation croisée les plus satisfaisants.

```
Best parameters: {'C': 0.001, 'penalty': 'none'}
```

FIGURE 4.18 – Les paramètres optimales pour RL

Naïve bayésienne multinomiale :

Nous avons effectué des expériences pour affiner notre modèle de Bayes Naïf multinomial (MNB), en mettant l'accent sur les paramètres les plus importants, tels que "alpha" et "fit prior". Les résultats de ces expériences sont présentés de manière captivante sous forme d'une figure qui met en évidence les variations significatives entre les différentes valeurs obtenues lors de la validation croisée.

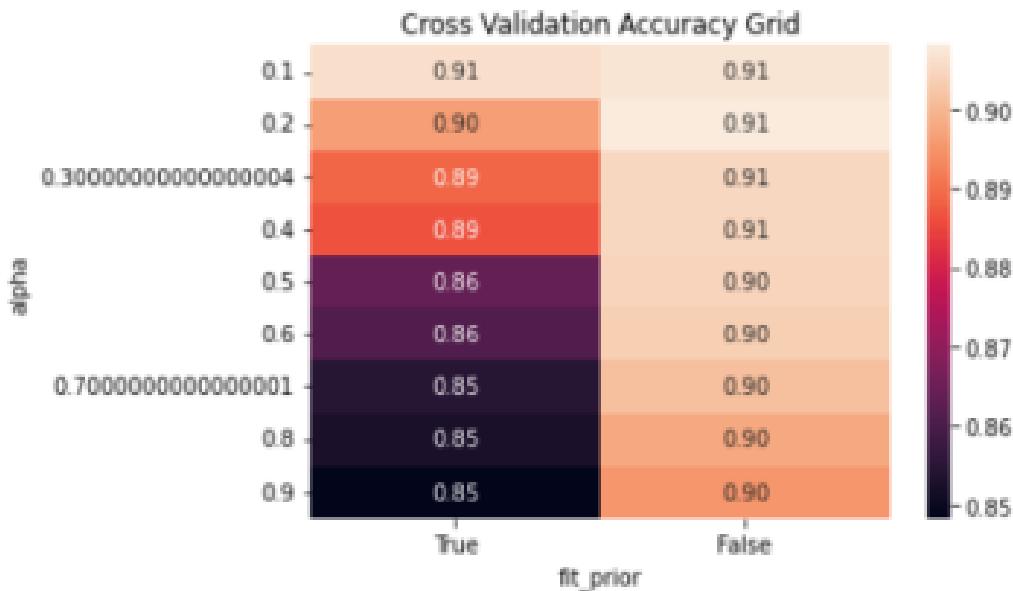


FIGURE 4.19 – La visualisation des différentes valeurs de cross validation pour MNB

Grâce à cette visualisation, nous sommes maintenant en mesure de choisir les valeurs de paramètres les plus appropriées, les réglages qui donnent lieu aux résultats de validation croisée les plus satisfaisants.

```
Best parameters: {'alpha': 0.2, 'fit_prior': False}
```

FIGURE 4.20 – Les paramètres optimales pour MNB

4.3.3 Evaluation et choix du modèle à déployer

La sélection du modèle est une étape essentielle et centrale de l'apprentissage automatique, qui requiert de choisir le modèle optimal parmi une pléthore de candidats. Cette étape clé permet de déterminer le modèle qui offre les meilleures performances pour l'ensemble de données pour garantir des résultats de qualité.

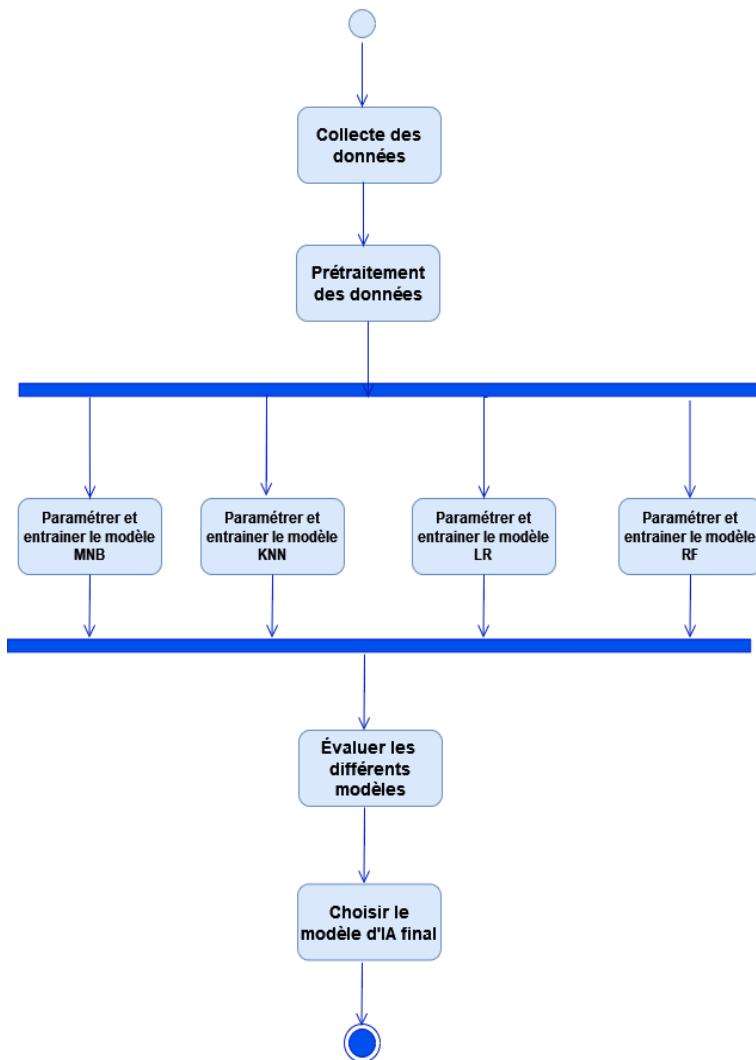


FIGURE 4.21 – Diagramme d'activité de choix du modèle

Pour sélectionner le modèle adéquat, nous avons travaillé sur la précision (accuracy) et la validation croisée, puis avons comparé les résultats et les avons visualisés à l'aide d'un diagramme en barres pour choisir le modèle ayant les meilleures performances.

Avec accuracy :

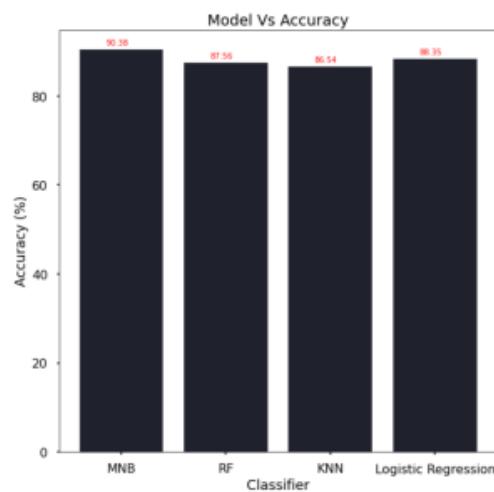


FIGURE 4.22 – Le pourcentage d'accuracy par modèle

Avec cross validation accuracy :

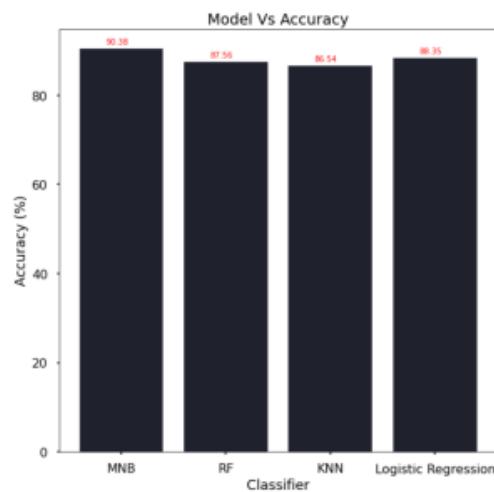


FIGURE 4.23 – Le pourcentage de cross validation accuracy par modèle

Pour sélectionner le modèle adéquat, nous avons travaillé avec la validation croisée qui consiste à diviser l'ensemble des données en plusieurs parties, appelées "folds". Cette technique est plus fiable et robuste pour évaluer la performance des modèles d'apprentissage automatique et choisir le modèle le plus performant pour notre tâche.

4.3.4 Déploiement du modèle

Cette section présente des scénarios d'exécution illustrés par des captures d'écran pour expliquer plus en détail le fonctionnement de notre application. Elle comprend également une explication sur le choix du logo et des couleurs utilisées.

Explication du logo



FIGURE 4.24 – Le Logo de l'application

Doct : le terme "Doct" dans le nom de l'entreprise fait référence à notre objectif de fournir des services de santé de qualité à nos utilisateurs.

Smart : "Smart" fait référence à l'utilisation de la technologie pour améliorer l'expérience utilisateur et fournir des solutions intelligentes dans le domaine de la santé.

Le stéthoscope : Il symbolise notre engagement envers la santé et notre désir de fournir des services de qualité dans ce domaine.

Charte Graphique

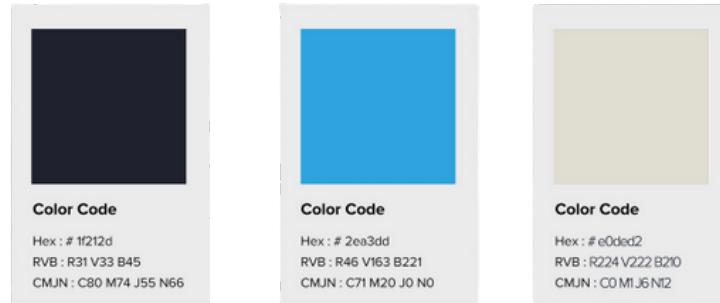


FIGURE 4.25 – Les couleurs de base

Les couleurs choisies, le bleu marine et le bleu clair, évoquent la confiance, la fiabilité et la sérénité. Le bleu marine est une couleur sobre et professionnelle, souvent utilisée dans le domaine de la santé pour représenter la confiance et la sécurité. Le bleu clair, quant à lui, est une couleur apaisante qui évoque la tranquillité et la fraîcheur. Ensemble, ces couleurs reflètent notre engagement à fournir des services de qualité, fiables et apaisants pour nos utilisateurs.

les différentes interfaces

Lorsque l'utilisateur ouvre l'application pour la première fois, il est dirigé vers cette page, qui fait office de page d'accueil.

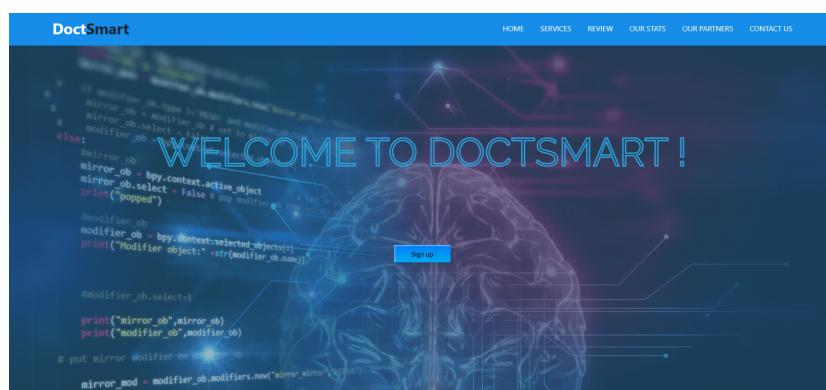


FIGURE 4.26 – La page d'accueil

Une variété de services sont généralement proposés pour répondre aux besoins de soins de santé des patients.

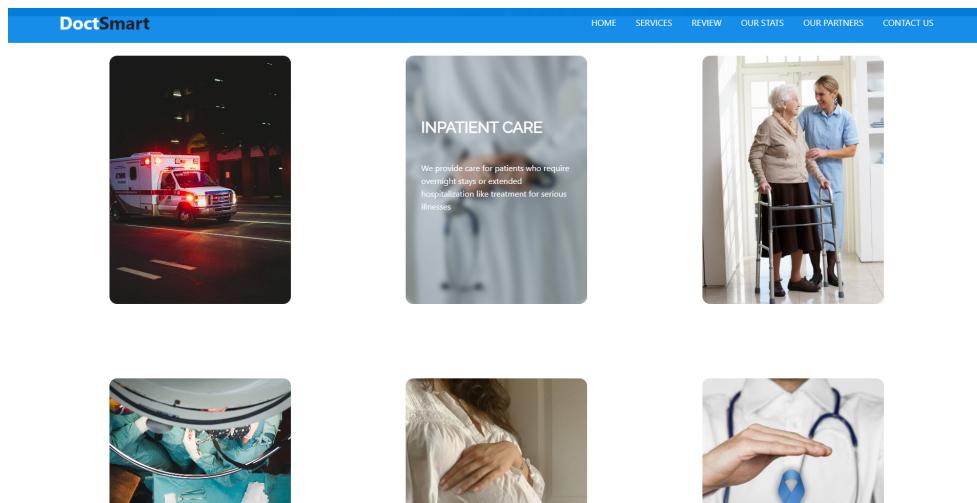


FIGURE 4.27 – Les différentes services

Lorsque vous accédez à l'interface de choix après avoir ouvert l'application, vous êtes présenté avec deux options : "Login Médecin" ou "Login Réceptionniste".

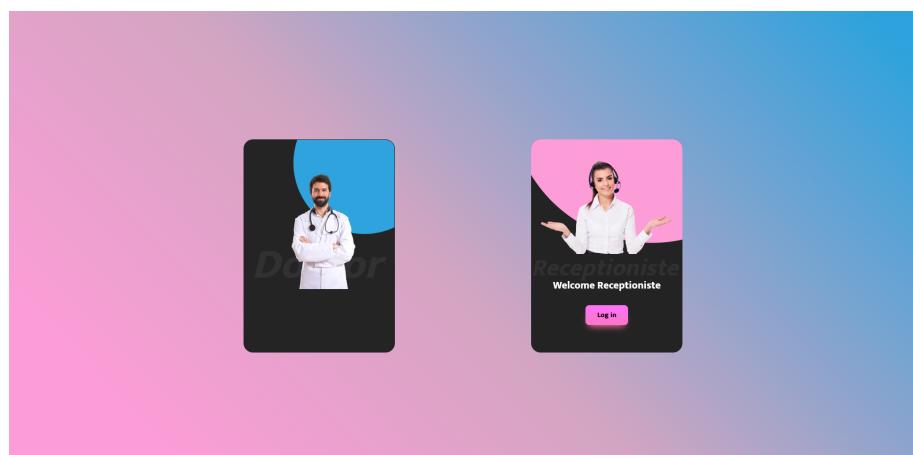


FIGURE 4.28 – L'interface du choix : Login médecin ou réceptionniste

Lorsque vous choisissez l'option "Login Réceptionniste" sur l'interface de choix de l'application, vous serez redirigé vers la page de connexion dédiée aux réceptionnistes.

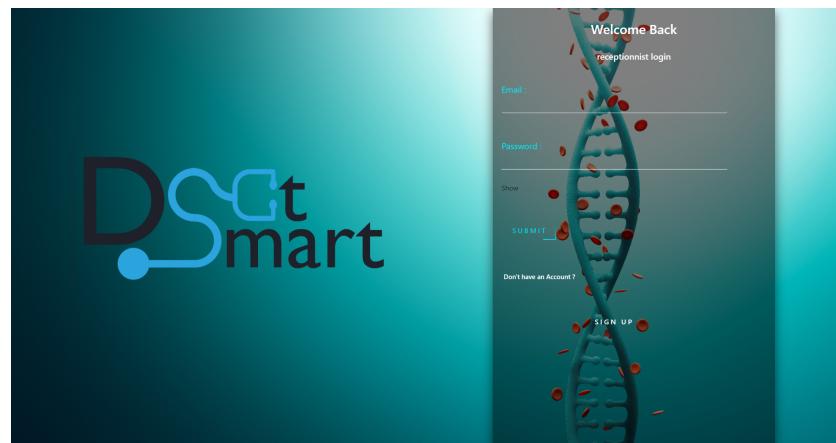


FIGURE 4.29 – Le Login de réceptioniste

Lorsque vous choisissez l'option "SignUp Médecin" sur l'interface de choix de l'application, vous serez redirigé vers la page d'inscription dédiée aux médecins.

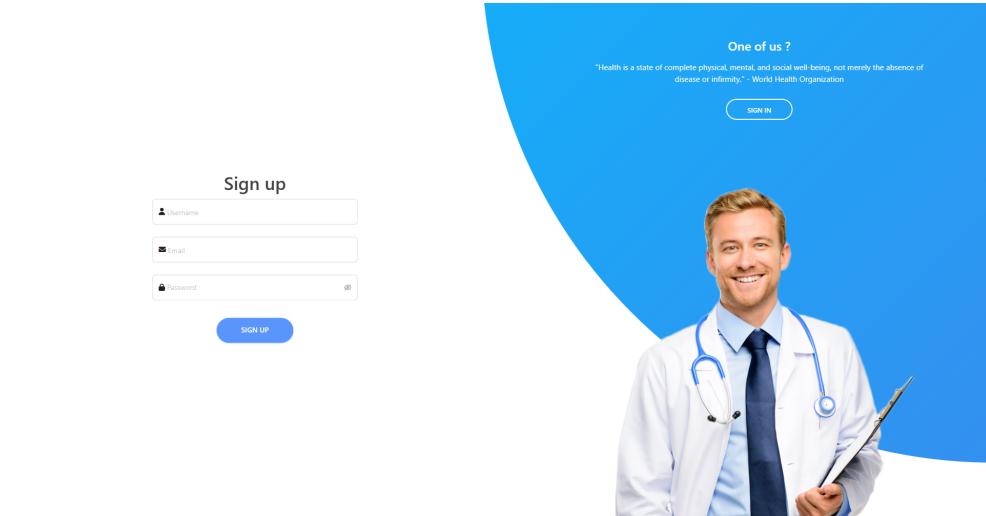


FIGURE 4.30 – Le SignUp de médecin

Le tableau de bord du médecin est une interface spécifique réservée aux médecins lorsqu'ils se connectent à leur compte. Il offre un aperçu des informations et des fonctionnalités essentielles dont les médecins ont besoin pour gérer leurs activités cliniques et interagir avec les patients. L'interface de sélection des symptômes est une fonctionnalité couramment



FIGURE 4.31 – Dashboard du médecin

utilisée dans les applications médicales ou les systèmes de diagnostic en ligne.

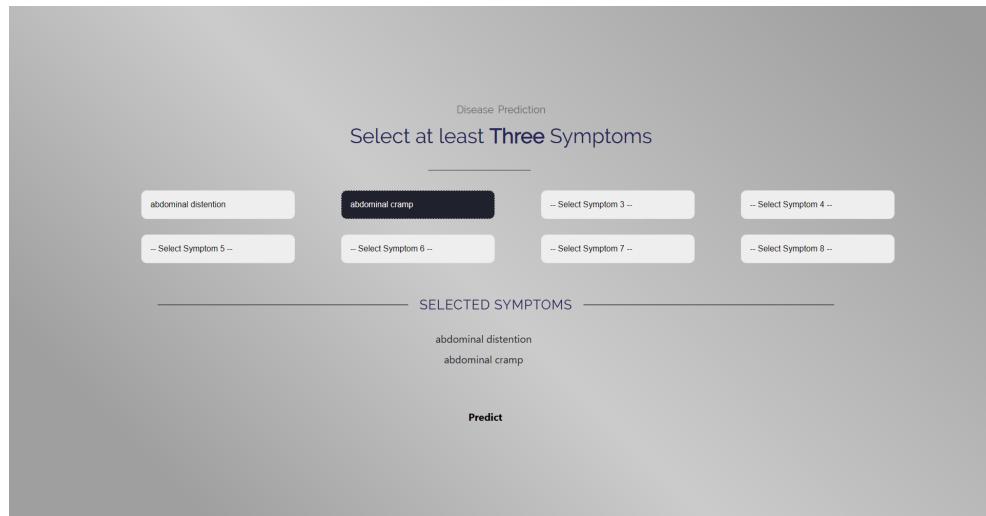


FIGURE 4.32 – Interface de sélection Des symptomes

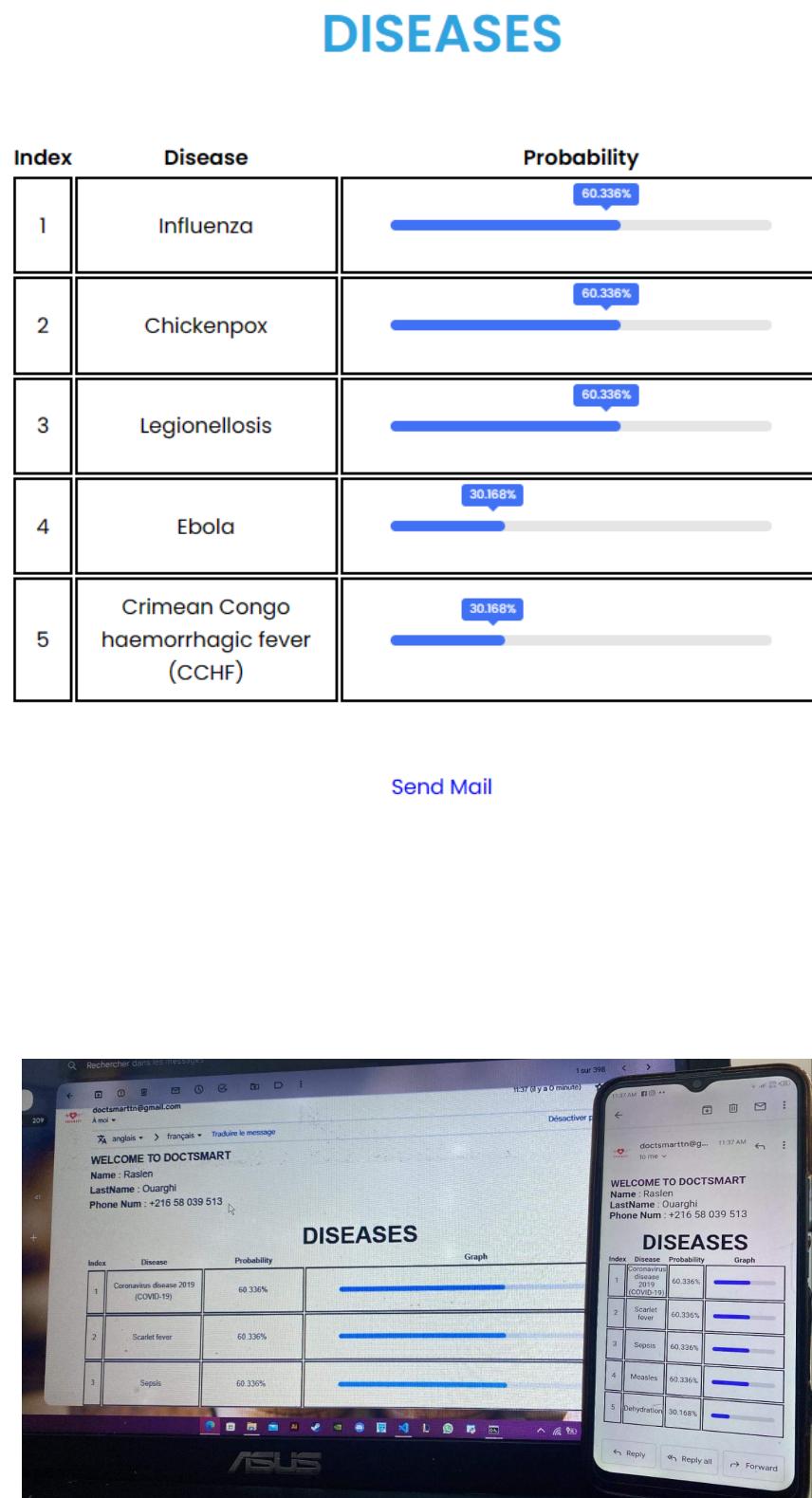


FIGURE 4.33 – Résultat du test médical envoyé Sur e-mail

4.4 Conclusion

Dans ce chapitre, nous avons examiné les différents langages de programmation et bibliothèques utilisés dans la partie développement de notre environnement afin de sélectionner les outils les plus appropriés pour créer notre application. Nous avons identifié les paramètres essentiels pour choisir le modèle le plus adapté et avons optimisé notre approche de développement en conséquence. Ensuite, nous avons présenté des captures d'écran pour illustrer les différentes fonctionnalités de notre application.

Conclusion et perspectives

Nous avons pour but de concevoir et développer une application web qui utilise les techniques de machine learning. Cette application vise à simplifier la communication entre les patients et le personnel médical, afin d'améliorer l'efficacité et la rapidité de la prise en charge des patients. L'objectif de cette application est d'optimiser les processus de traitement en utilisant les techniques d'apprentissage automatique pour améliorer la qualité des soins.

Ce rapport décrit en détail les différentes étapes que nous avons entreprises pour atteindre le résultat attendu de notre projet. Nous avons débuté en présentant la problématique à résoudre, les objectifs à atteindre, ainsi que le cadre de travail. Ensuite, nous avons effectué une étude approfondie des solutions existantes et présenté notre propre solution. Cette étape nous a permis de spécifier les exigences du système et d'analyser les fonctionnalités à fournir avant de passer à la phase de conception. Enfin, pour conclure notre rapport, nous avons exposé la phase de réalisation où nous avons présenté les outils technologiques que nous avons utilisés pour le développement de notre application. Nous avons également inclus des captures d'écran illustrant les différentes fonctionnalités mises en place.

Ce projet nous a permis de mettre en pratique nos connaissances théoriques et pratiques, tout en apprenant de nouvelles technologies de développement telles que React, Flask, Node.js, MongoDB, ainsi que des notions de machine learning. Bien que nous ayons rencontré des difficultés tout au long de ce projet, nous avons finalement réussi à atteindre notre objectif initial.

Dans la dernière partie du rapport, nous avons examiné différentes perspectives pour améliorer la rapidité de consultation, le traitement des données médicales et les conseils dans une application web. Voici un résumé de ces perspectives :

1. Interface utilisateur réactive : L'application web devrait avoir une interface utilisateur intuitive, conviviale et réactive, permettant aux utilisateurs de naviguer facilement et d'accéder rapidement aux fonctionnalités pertinentes.

2. Communication en temps réel : L'intégration de fonctionnalités de communication en temps réel, telles que la messagerie instantanée ou les chats vidéo, permet aux utilisateurs de consulter rapidement les médecins et d'obtenir des réponses en temps réel.
3. Gestion efficace des données médicales : Un système de gestion des données médicales efficace et sécurisé doit être mis en place, permettant le stockage sécurisé des dossiers médicaux, un accès rapide aux antécédents médicaux des patients et le partage d'informations pertinentes avec les médecins.
4. Algorithmes de tri et de priorisation : Des algorithmes de tri et de priorisation doivent être développés pour classer les demandes des utilisateurs en fonction de leur gravité ou de leur urgence, optimisant ainsi le temps de traitement des médecins et garantissant une réponse rapide aux besoins les plus critiques.
5. Système de recommandation : L'utilisation de techniques de machine learning permet de créer un système de recommandation personnalisé qui propose des conseils médicaux pertinents aux utilisateurs en fonction de leurs symptômes, antécédents médicaux et autres facteurs pertinents.
6. Compatibilité multi-appareils : L'application web doit être compatible avec différents appareils (ordinateurs de bureau, tablettes, smartphones) et différents systèmes d'exploitation, permettant ainsi une consultation rapide et facile pour les utilisateurs où qu'ils se trouvent.
7. Sécurité et confidentialité : Des mesures de sécurité robustes doivent être mises en place pour protéger les données médicales sensibles des utilisateurs, en respectant les réglementations de confidentialité des données, telles que le RGPD, afin de garantir la confidentialité des informations personnelles.

Il est important de collaborer étroitement avec des experts médicaux et des professionnels du développement d'applications web pour concevoir et mettre en œuvre ces fonctionnalités de manière optimale.

Annexe

L'annexe de ce rapport contient une section dédiée aux cas d'utilisation étudiés, où vous pourrez trouver une série de fiches descriptives détaillées, chacune décrivant un cas d'utilisation spécifique.

Nom du CU	Gérer un rendez-vous
Acteur	Réceptioniste
Objectif	Donner la possibilité à un réceptioniste de gérer les rendez-vous des patients (ajouter, modifier, consulter et supprimer).
Précondition	Le réceptioniste doit être connecté au système.
Définition des enchainements	<ol style="list-style-type: none"> 1. Le système affiche le calendrier des rendez-vous. 2. Le réceptioniste peut choisir de créer un nouveau rendez-vous. 3. Le système affiche une fenêtre dédiée pour saisir les informations relatives au rendez-vous. 4. Le réceptioniste saisit les informations du rendez-vous. 5. Le réceptioniste peut modifier les informations. 6. Le réceptioniste peut consulter les détails d'un rendez-vous existant.
Exception(s)	Données manquantes : le rendez-vous ne sera pas ajouté.
Post conditions	Le système enregistre le rendez-vous et le calendrier du médecin est mise à jour.

TABLE 4.2 – Gérer un rendez-vous tableau descriptif

Nom du CU	S'authentifier
Acteur	Médecin,Réceptioniste
Objectif	Permettre à l'utilisateur de se connecter en toute sécurité au système
Précondition	Serveur Opérationnel
Définition des enchainements	<ol style="list-style-type: none"> 1. L'utilisateur lance une interface de connexion. 2. Le système présente les champs "Email" et "Mot de passe". 3. l'utilisateur entre son email et son mot de passe. 4. Le système vérifie les informations de connexion 5. Si les informations sont valides , le système donne accès à l'utilisateur à l'interface principale.
Exception(s)	Mot de passe incorrect Adresse e-mail incorrecte serveur hors ligne
Post conditions	l'utilisateur peut se connecter à l'interface principale du système

TABLE 4.3 – Authentication tableau descriptif

NETOGRAPHIE

-<https://www.ibm.com/fr-fr/topics/artificial-intelligence/>[1][Consulté 9-Avril-2023].

-The Hundred- Page Machine Learning Book/ [2][Consulté 9-Avril-2023].

-https://www.codecademy.com/learn/paths/machine-learning?utm_id=t_kwd-790967083291loc-227:ag_1265538519950069:cp_370379642:n_o:d_c&msclkid=3ed5511989dd114dac0835433e14utm_source=bing&utm_medium=cpc&utm_campaign=ROW/ [3][Consulté 10-Avril-2023].

- <https://play.google.com/store/apps/details?id=epic.mychart.android/> [4][Consulté 10-Avril-2023].

- <https://play.google.com/store/apps/details?id=com.ecw.healow/> [5][Consulté 10-Avril-2023].

- <https://blog-gestion-de-projet.com/modele-en-cascade/> [6][Consulté 11-Avril-2023].

- [https://blog-gestion-de-projet.com/modele-en-cascade/\[7\]](https://blog-gestion-de-projet.com/modele-en-cascade/)[Consulté 11-Avril-2023].

- <https://www.ionos.fr/startupguide/productivite/modele-en-spirale/> [8][Consulté 11-Avril-2023].

- <https://www.ibm.com/docs/fr/spss-modeler/saas?topic=dm-crisp-help-overview/>[9][Consulté 11-Avril-2023].

- <https://en.blog.businessdecision.com/the-key-to-data-science-success-is-the-crisp> 12-Avril-2023].

- <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/> [12-Avril-2023].

- <https://studysection.com/blog/history-of-unified-modeling-language-uml/>[12][Consulté 12-Avril-2023].

- <https://dev.to/lakshyabatman/cheatsheet-to-react-lifecycle-hooks-part-1-597d/>[13][Consulté 13-Avril-2023].

- <https://www.ibm.com/topics/random-forest/>[14][Consulté 13-Avril-2023].
- <https://www.ibm.com/topics/knn/>[15][Consulté 13-Avril-2023].
- <https://www.ibm.com/topics/logistic-regression/>[16][Consulté 13-Avril-2023].
- <https://www.ibm.com/topics/naive-bayes/>[17][Consulté 13-Avril-2023].
- <https://inside-machinelearning.com/recall-precision-f1-score/?fbclid=IwAR1RvsFkXFuXRpqYxB0/>[18][Consulté 14-Avril-2023].
- https://datascientest.com/cross-validation?fbclid=IwAR26Y_SPU_moQjYzshGUxejM1quYRc4u5kHWriI[19][Consulté 14-Avril-2023].

Bibliographie