**Rapport**

**4ème année**

**Ingénierie Informatique et Réseaux**

**Sous le thème**

Mise en place d’une application de gestion de cabinet dentaire

**Réalisé par :**

BOUX Hamza

ENNOUNI Ayoub

**Encadré par :**

M Omar El Midaoui

# Table des matières

[Dédicaces i](#_Toc130848826)

[Remerciement ii](#_Toc130848827)

[Résumé iii](#_Toc130848828)

[Abstract iv](#_Toc130848829)

[Table des matières v](#_Toc130848830)

[Liste des figures vii](#_Toc130848831)

[Liste des tableaux viii](#_Toc130848832)

[Liste des abréviations ix](#_Toc130848833)

[Introduction Générale 1](#_Toc130848834)

[Chapitre 1 : Contexte générale 3](#_Toc130848835)

[Introduction 4](#_Toc130848836)

[1. Présentation du projet 4](#_Toc130848837)

[2. Organisation du projet 5](#_Toc130848838)

[2.1 Processus de Développement 5](#_Toc130848839)

[2.2 Planification Opérationnelle 6](#_Toc130848840)

[Conclusion 10](#_Toc130848841)

[Chapitre 2 : Analyse & Conception 11](#_Toc130848842)

[Introduction 12](#_Toc130848843)

[1. Description des besoins fonctionnels 12](#_Toc130848844)

[1.1 Périmètre de projet 12](#_Toc130848845)

[1.2 Description de la structure générale du site 12](#_Toc130848846)

[1.3 Fonctionnalités de projet 12](#_Toc130848847)

[2. Description des besoins ergonomiques 13](#_Toc130848848)

[2.1 Définir les règles ergonomiques 13](#_Toc130848849)

[2.2 Formaliser une charte ergonomique 13](#_Toc130848850)

[3. Description des besoins graphiques 13](#_Toc130848851)

[4. Descriptions des besoins techniques 13](#_Toc130848852)

[4.1 Outils de développement 13](#_Toc130848853)

[5. Modélisation UML 13](#_Toc130848854)

[5.1. Vue Fonctionnelles (diagrammes des cas d’utilisation) 14](#_Toc130848855)

[5.2. Vue Dynamique (Diagramme de séquence) 15](#_Toc130848856)

[5.3. Vue Logique (Diagramme de classes) 15](#_Toc130848857)

[6. Implémentation //facultatif 15](#_Toc130848858)

[6.1. Schéma de la base de données 15](#_Toc130848859)

[6.2. Maquettes 15](#_Toc130848860)

[Conclusion 16](#_Toc130848861)

[Chapitre 3 : Interfaces de l’application 17](#_Toc130848862)

[Introduction 18](#_Toc130848863)

[1. Interfaces 18](#_Toc130848864)

[Conclusion 18](#_Toc130848865)

[Conclusion Générale 19](#_Toc130848866)

[Références 21](#_Toc130848867)

[Annexe 1 : questionnaire (exemple) 23](#_Toc130848868)

# Liste des figures

[Figure.1 : légende qui présente une idée sur le contenu de la figure 4](#_Toc130848869)

[Figure.2 : Le processus de développement en Y 5](#_Toc130848870)

[Figure.3 : Le diagramme de Gantt 8](#_Toc130848871)

[Figure.4 : Le diagramme de Pert 9](#_Toc130848872)

[Figure.5 : Diagramme d’études des cas –Administrateur 15](#_Toc130848873)

[Figure.6 : Diagramme d’études des cas de l’Administrateur de l’organisme 15](#_Toc130848874)

[Figure.7 : Diagramme d’études des cas du client 15](#_Toc130848875)

[Figure.8 : Maquette de la page d'accueil 16](#_Toc130848876)

Introduction Générale

Le secteur médical, et en particulier la gestion des cabinets dentaires, connaît une transformation significative grâce à l'évolution des technologies de l'information et de la communication. Les cabinets dentaires doivent désormais adopter des solutions informatisées pour améliorer l'efficacité de leurs opérations quotidiennes, réduire les erreurs et offrir des soins de qualité supérieure. Dans ce contexte, le but de ce projet est de développer une application de gestion de cabinet dentaire en utilisant des technologies modernes telles que Spring Boot, Spring Data, Spring MVC et Thymeleaf. Cette application vise à simplifier et automatiser les processus administratifs, améliorer la gestion des dossiers patients, optimiser la planification des rendez-vous et faciliter la communication entre le cabinet et ses patients.

Pour atteindre ces objectifs, l'application est structurée en plusieurs modules, notamment la gestion des patients, la gestion des factures, la gestion des dossiers médicaux, la gestion des utilisateurs, etc. Elle est destinée à différents types d'utilisateurs, tels que les dentistes, les secrétaires, chacun ayant accès à des fonctionnalités spécifiques après authentification.

Le développement de cette application nécessite une étude approfondie des besoins fonctionnels et non fonctionnels, une documentation des technologies utilisées, une analyse et une conception de l'application, ainsi que la mise en œuvre proprement dite de l'application.

Ce rapport est organisé en trois chapitres, chacun traitant d'un aspect spécifique du projet de développement de l'application de gestion de cabinet dentaire. Dans le premier chapitre, nous introduirons le contexte du projet, en présentant brièvement le problème à résoudre ainsi que les objectifs visés. Il détaillera également l'organisation du projet, en expliquant le processus de développement utilisé pour sa réalisation.

Le deuxième chapitre sera consacré à l'étude des besoins, incluant une analyse approfondie des besoins fonctionnels et non fonctionnels. Nous aborderons également les exigences techniques et ergonomiques nécessaires pour garantir l'efficacité et la convivialité de l'application. Ensuite, nous présenterons une documentation relative aux technologies utilisées, notamment Spring Boot, Spring Data, Spring MVC et Thymeleaf, et la conception du système d'information modélisé en UML.

Le troisième chapitre exposera en détail les différentes interfaces de l'application et leur fonctionnement, comprenant une description approfondie de chacune d'elles. Ce rapport vise à fournir une analyse complète et des recommandations pratiques pour le développement d'une application de gestion de cabinet dentaire, en mettant en lumière son importance dans l'amélioration des processus opérationnels et des soins aux patients.

Chapitre 1 : Contexte générale

Introduction

Ce chapitre traite du contexte général du projet, des objectifs à atteindre, et de l'organisation du processus de réalisation. Nous aborderons les problèmes rencontrés dans la gestion des cabinets dentaires, l'importance de proposer des solutions innovantes, et les objectifs du projet. Enfin, nous expliquerons l'organisation du processus de réalisation.

### Présentation du projet

#### Etude de l’existant

La gestion des cabinets dentaires repose encore largement sur des méthodes traditionnelles et des systèmes informatiques obsolètes, ce qui entraîne des inefficacités et des erreurs humaines. De nombreuses solutions de gestion de cabinet existent sur le marché, telles que Dentrix, Eaglesoft, et Open Dental, offrant des fonctionnalités de gestion de dossiers patients, de planification de rendez-vous, et de gestion des stocks. Cependant, ces solutions présentent également des lacunes importantes. Par exemple, certaines ne sont pas suffisamment intuitives pour les utilisateurs non techniques ou manquent de fonctionnalités de personnalisation adaptées aux besoins spécifiques des cabinets dentaires. D'autres plateformes souffrent de limitations en termes d'intégration avec d'autres systèmes ou de support client inadéquat. Dans le cadre de ce projet, nous proposons de répondre à ces lacunes en développant une application de gestion de cabinet dentaire qui automatisera les processus administratifs, améliorera la gestion des dossiers patients, optimisera la planification des rendez-vous, et facilitera la communication entre le cabinet et ses patients. Ces améliorations visent à résoudre certains des problèmes rencontrés par les solutions existantes et à offrir une gestion plus efficace et personnalisée pour les cabinets dentaires.

### Organisation du projet

### [Processus de Développement](#_Toc355001711)

* **Processus 2TUP**

Le processus 2TUP (Two Track Unified Process) est une méthode de développement logiciel qui se base sur le processus unifié (UP) et qui a été spécialement conçue pour les projets de développement web et mobiles. Il est caractérisé par une approche itérative et incrémentale qui se divise en deux voies : La première voie est dédiée à la réalisation fonctionnelle et la seconde voie à la réalisation technique.

La figure suivante représente les étapes de développement des trois branches du processus 2TUP.

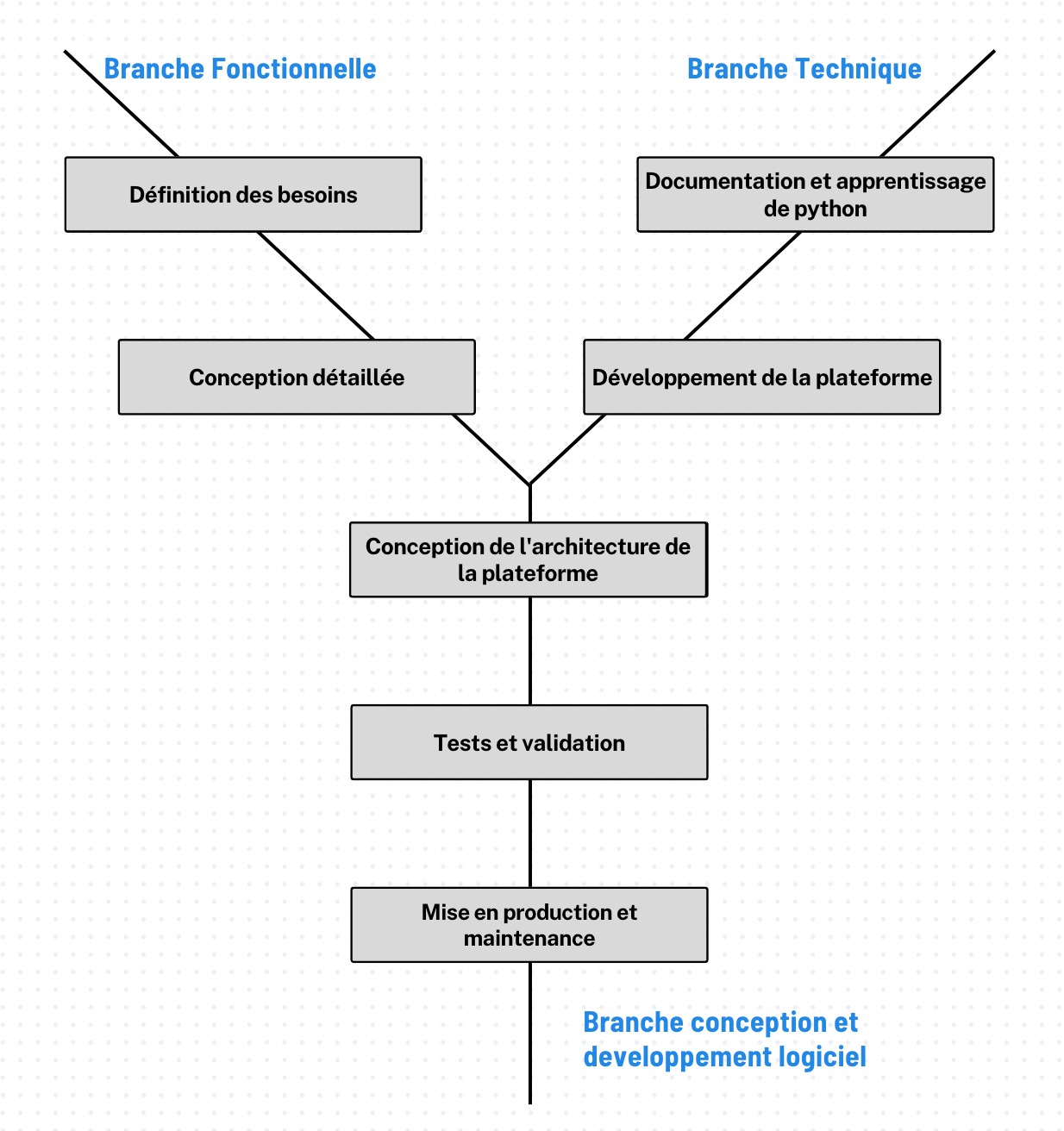


Figure.1 : Le processus de développement en Y

Conclusion

Dans ce chapitre on a présenté le contexte général du projet ainsi que les problèmes à résoudre et les objectifs à atteindre. Nous avons également décrit l’organisation du projet en utilisant le processus 2TUP.

L’objectif de ce chapitre était de fournir un aperçu général du projet et de présenter les problèmes à résoudre ainsi que les objectifs à atteindre. Le prochain chapitre se concentrera sur l’analyse approfondie des besoins fonctionnels et non fonctionnels, la documentation du langage utilisé ainsi la conception détaillée de ce projet.

Chapitre 2 : Analyse & Conception

Introduction

Dans ce deuxième chapitre de notre rapport, nous nous concentrons sur l’analyse approfondie des besoins fonctionnels et non fonctionnels de notre projet d’application de cabinet dentaire. Ces besoins fourniront une base solide pour la conception de notre solution, en identifiant clairement ce que notre application doit réaliser du point de vue des fonctionnalités attendues et des contraintes à prendre en compte.

La première partie de ce chapitre sera consacrée à l’identification et à la description détaillée des besoins fonctionnels. Ces besoins fonctionnels représenteront les fonctionnalités spécifiques que notre application doit offrir pour répondre aux attentes des utilisateurs. Nous examinerons en profondeur chaque module et chaque acteur de l’application afin de définir clairement leurs responsabilités et leurs interactions, et de déterminer les fonctionnalités indispensables pour la réussite de notre projet.

Ensuite nous entamerons la partie conception où nous traduirons les analyses effectuées précédemment en modèles et en diagrammes. Nous utiliserons des outils tels que les diagrammes de classes. Cette phase de conception nous permettra d’avoir une vision claire et organisée de notre solution, favorisant ainsi une implémentation efficace et cohérente.

Ce chapitre joue un rôle essentiel dans la planification et la réalisation de notre projet d’application de cabinet dentaire, en jetant les bases nécessaires pour développer une solution robuste, fonctionnelle et en adéquation avec les attentes des utilisateurs et les contraintes du système.

## Description des besoins fonctionnels

### Périmètre de projet

Ce projet a pour objectif de développer une application de gestion de cabinet dentaire en utilisant les technologies Spring Boot, Spring Data, Spring MVC et Thymeleaf. L'objectif principal est de créer une plateforme qui automatise les tâches liées à la gestion des différents acteurs du cabinet dentaire, à savoir les dentistes, les secrétaires. L'automatisation de ces tâches vise à simplifier et optimiser les processus administratifs, améliorer la gestion des dossiers patients, et optimiser la planification des rendez-vous, etc…. Grâce à cette application, les dentistes pourront gérer les dossiers médicaux des patients. Les secrétaires pourront coordonner les rendez-vous avec les patients, gérer les factures, etc. En fournissant une plateforme centralisée et conviviale, cette application vise à améliorer l'efficacité opérationnelle et la qualité des soins dans les cabinets dentaires.

## Description des besoins ergonomiques

### Besoins techniques

### L’application doit être développée en utilisant le langage de programmation Java et le Framework Spring Boot.

### La base de données doit être sécurisée et fiable.

### L’application doit être compatible avec différents navigateurs web.

### Besoins ergonomiques

### L’interface utilisateur doit être intuitive et facile à utiliser.

### Le design doit être attractif et s’adapter à différents appareils.

### Besoins de performance

### L’application doit être rapide et réactive.

### Les temps de chargement doivent être optimisés.

### La disponibilité du service doit assurer en tout temps.

### Besoins de sécurité

### L’application doit être sécurisée contre les attaques externes.

### Les données des utilisateurs et des cours doivent être protégées et confidentielles.

## Descriptions des besoins techniques

### Outils de développement

Pour mener à bien le projet, nous avons choisi de développer une application en utilisant le Framework Spring Boot et Java comme langage de programmation, donc nous avons eu recours aux Outils et Framework suivant :

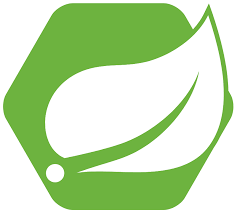
* Java :



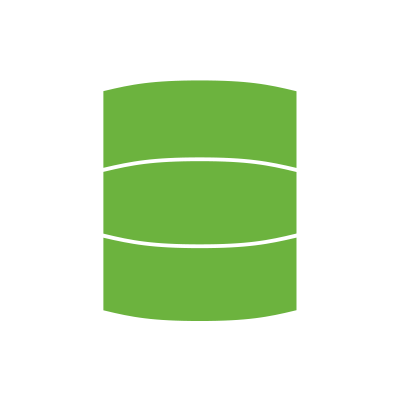
Langage de programmation polyvalent largement utilisé pour le développement backend. Il offre une grande portabilité et une syntaxe claire, ce qui en fait un choix populaire pour les applications web.

* ****Spring Boot**** :

Framework Java qui simplifie le développement d'applications web en fournissant des fonctionnalités prêtes à l'emploi pour la configuration, la sécurité, et la gestion des dépendances. Il permet de créer rapidement des applications web robustes et évolutives.



* Spring Data :



Module de Spring qui facilite l'accès aux données en fournissant une abstraction au-dessus des technologies de persistance de données, telles que les bases de données relationnelles et NoSQL. Il simplifie le code en évitant la répétition des opérations CRUD (Create, Read, Update, Delete).

* ****Spring MVC :****

Modèle-Vue-Contrôleur (MVC) est un motif de conception couramment utilisé pour développer des applications web. Spring MVC fournit une structure pour organiser le code en séparant les préoccupations (logique métier, présentation, contrôle) et en facilitant la création de vues dynamiques.



* ****Thymeleaf :****

Moteur de template pour les applications web Java. Il permet d'intégrer facilement du contenu dynamique dans les pages HTML en utilisant des attributs spécifiques. Thymeleaf facilite la création de vues côté serveur en rendant le code HTML plus lisible et maintenable.



## Modélisation UML

La modélisation UML (Unified Modeling Language) est une méthode standardisée utilisée dans le domaine du développement logiciel pour représenter graphiquement les différents aspects d’un système. UML offre un langage visuel pour décrire les composants, les interactions et les fonctionnalités d’un système, ce qui facilite la compréhension et la communication entre les membres de l’équipe de développement. L’objectif principal de l’utilisation d’UML est de créer un modèle clair et précis du système à développer, facilitant ainsi la conception, la compréhension et la maintenance du projet.

UML représente une variété de vues qui permettent de représenter les différents aspects d’un système. Ces vues sont regroupées en deux catégories : les vues statiques et les vues dynamiques.

Les vues statiques, représentant le système physiquement, incluent les diagrammes d’objets, les diagrammes de classes et les diagrammes de déploiement. Ces diagrammes permettent de visualiser la structure du système, les relations entre les entités, les dépendances et les interactions entre les différents composants.

Les vues dynamiques, illustrant le fonctionnement du système, comprennent les diagrammes de séquence, les diagrammes de collaboration, les diagrammes d’états-transitions et les diagrammes d’activités. Ces diagrammes mettent l’accent sur le déroulement des processus, les flux de données, les évènements et les changements d’états au sein du système.

L’utilisation d’UML présente plusieurs avantages pour la conception d’un projet : Des descriptions graphiques, des vues différentes à des étapes différentes, et un regroupement des descriptions.

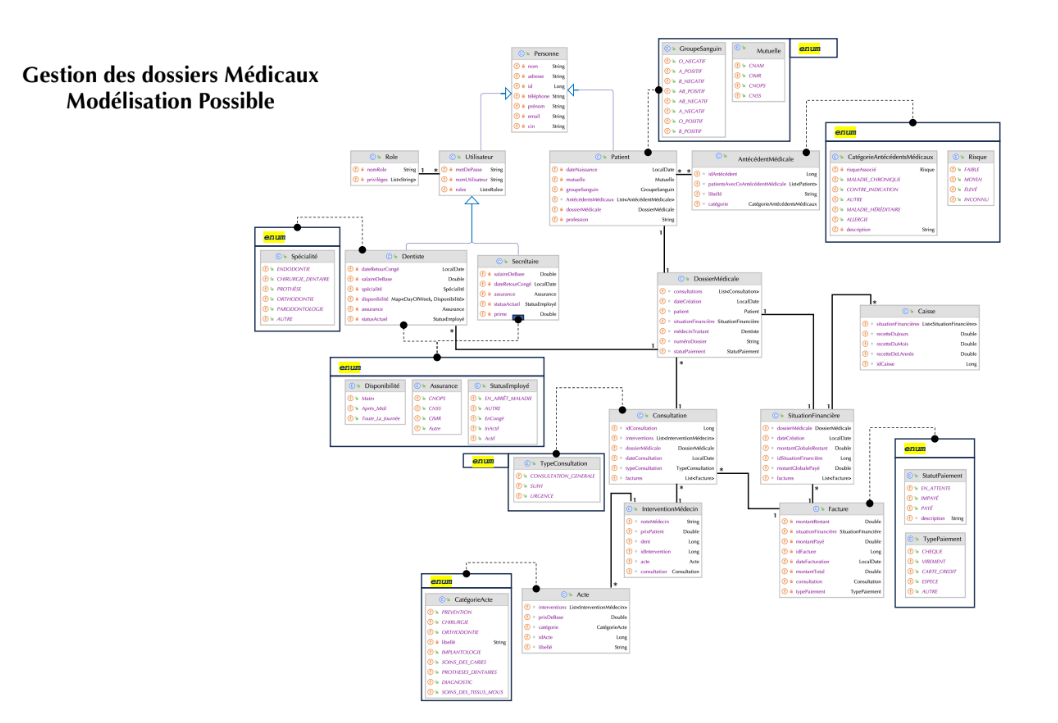
**StarUML** est un outil de génie logiciel dédié à la modélisation UML et édité par la société coréenne MKLabs. Il est multiplateforme et fonctionne sous Windows, Linux et MacOs. La dernière version gère l’ensemble des diagrammes définis par UML2, ainsi que plusieurs diagrammes SysML, l’organigramme, les diagrammes de flux de données, et les diagrammes entité-association.





**ArgoUML** est un logiciel libre de création de diagrammes UML. Programmé en Java, il est édité sous licence EPL 1.0. Il est multilingue, supporte la génération de code et l’ingénierie inverse.

### Diagramme de classes



Conclusion

La conception d’une nouvelle application et le choix de la technologie appropriée sont des éléments essentiels dans le processus de développement. La phase de conception permet de définir les besoins et les contraintes du système, tandis que le choix de la technologie détermine la façon dont ces besoins seront satisfaits. Une bonne coordination entre la conception et le choix technologique garantit la création d’une application fonctionnelle, adaptée aux attentes des utilisateurs et prête à être déployée avec succès.

Chapitre 3 : Interfaces de l’application

Introduction

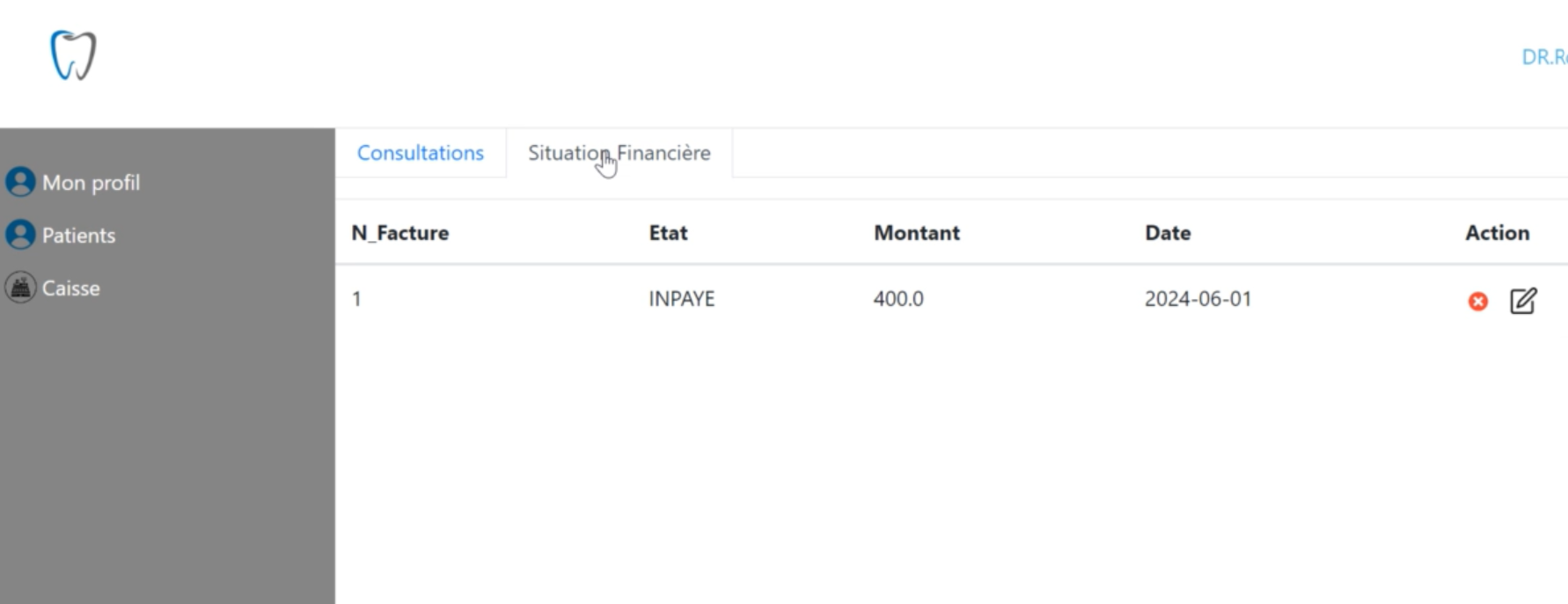
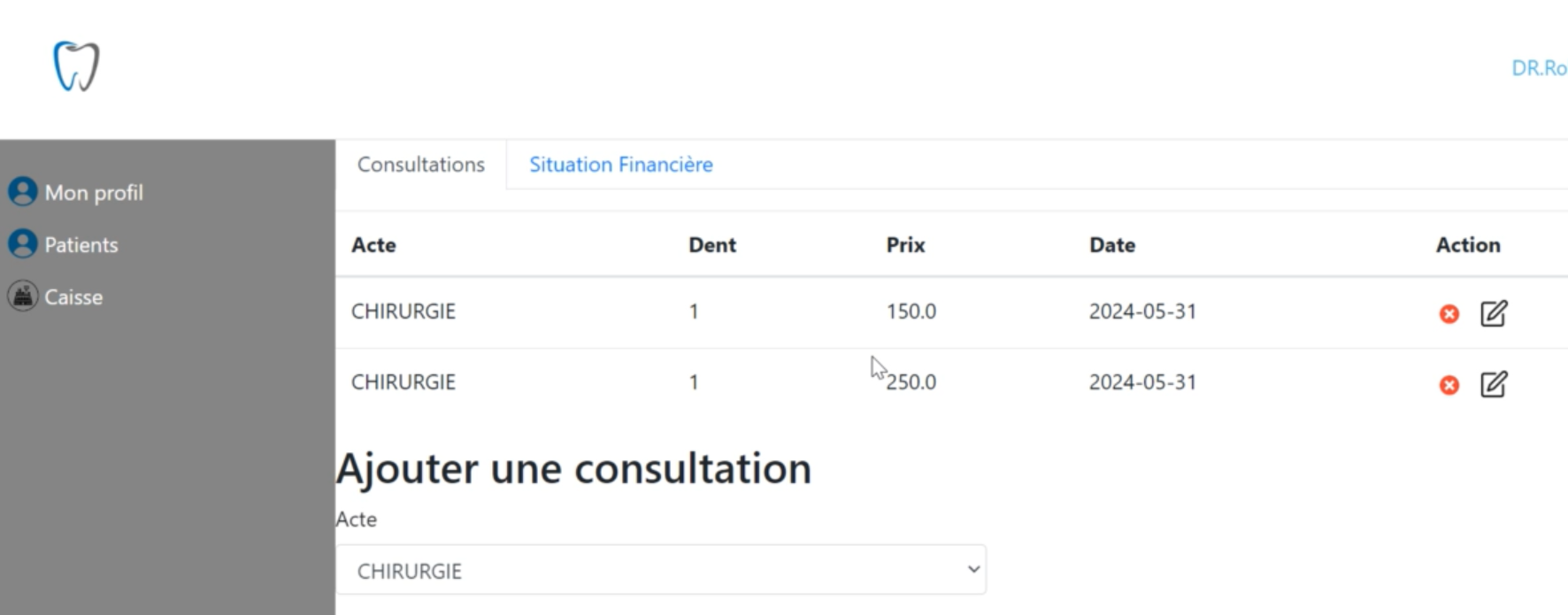
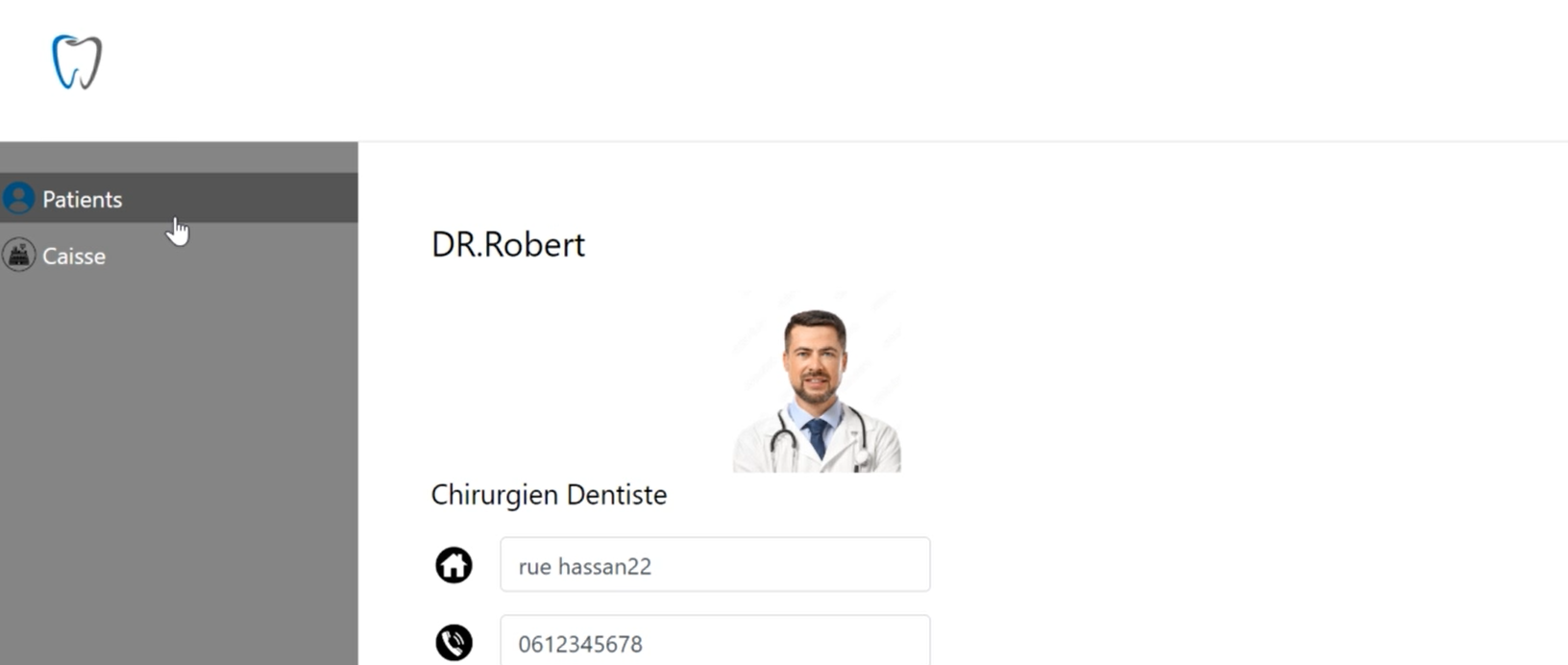
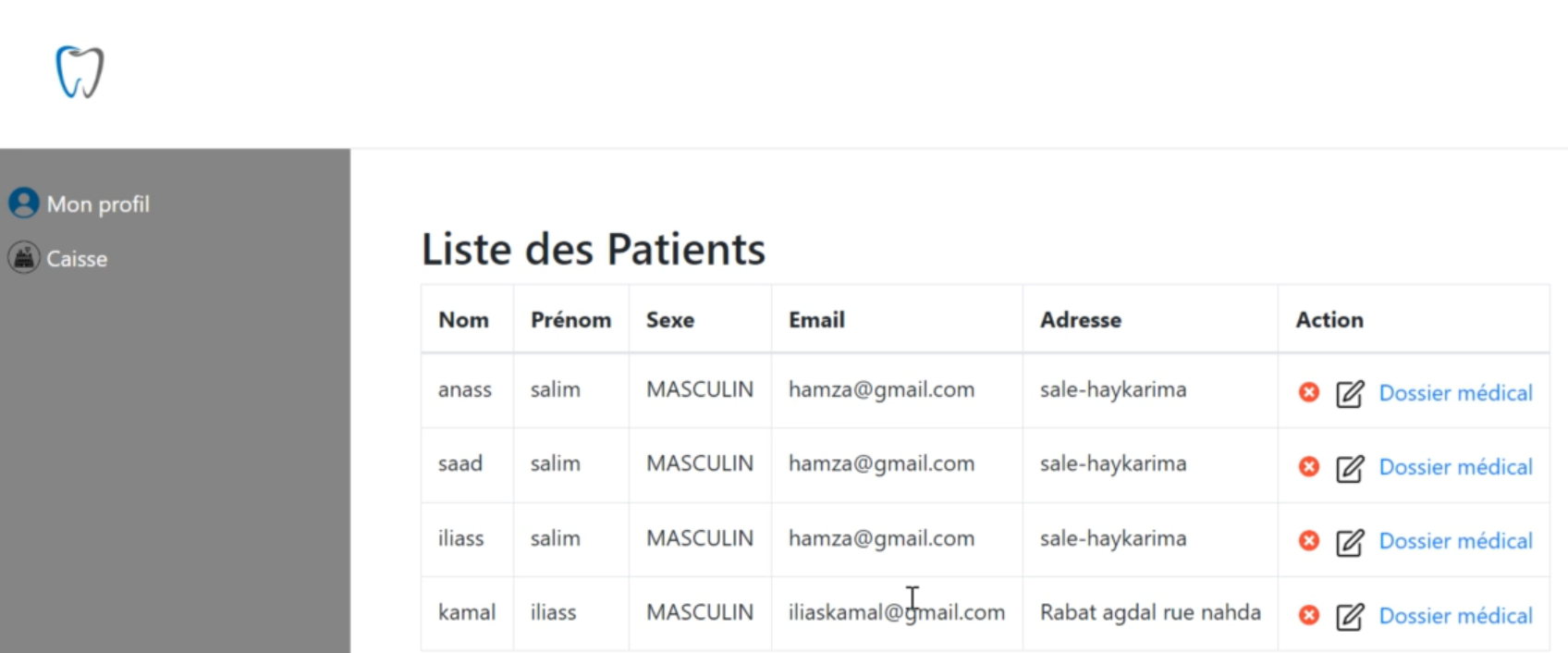
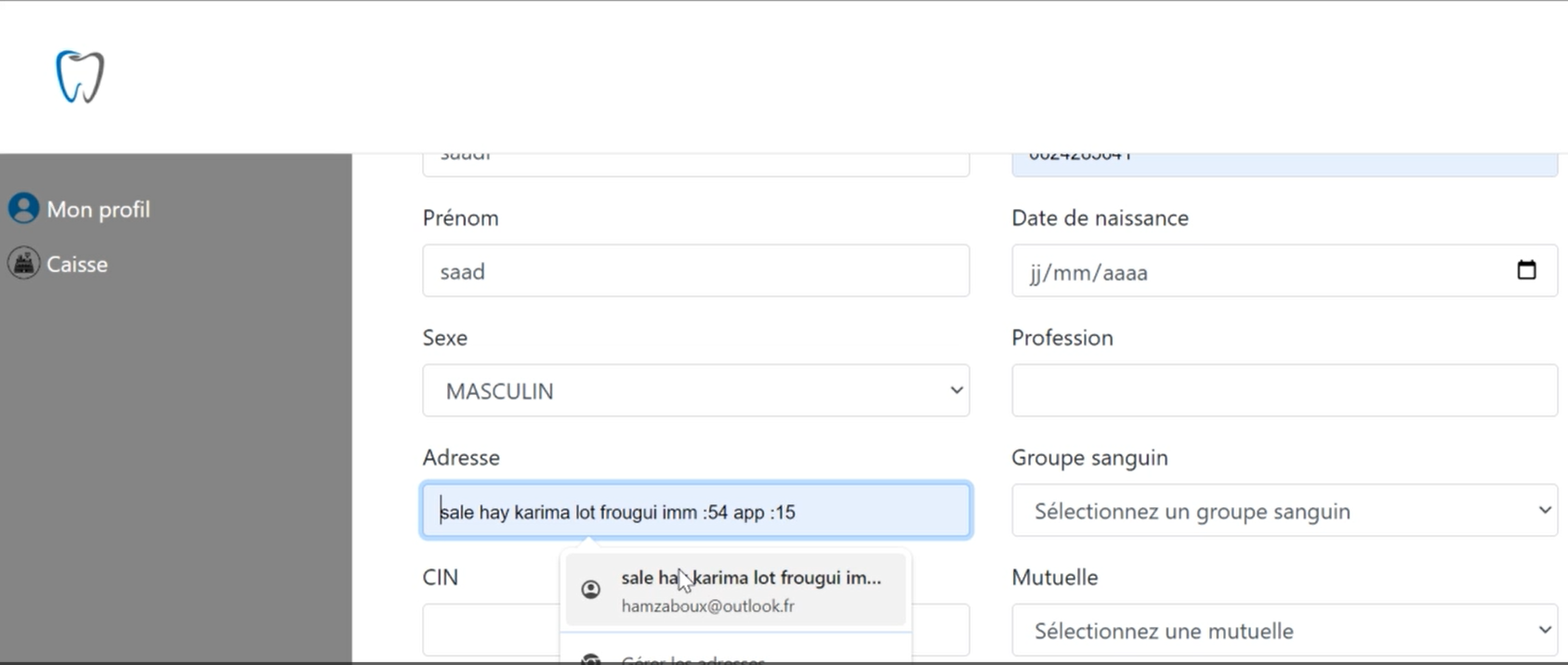
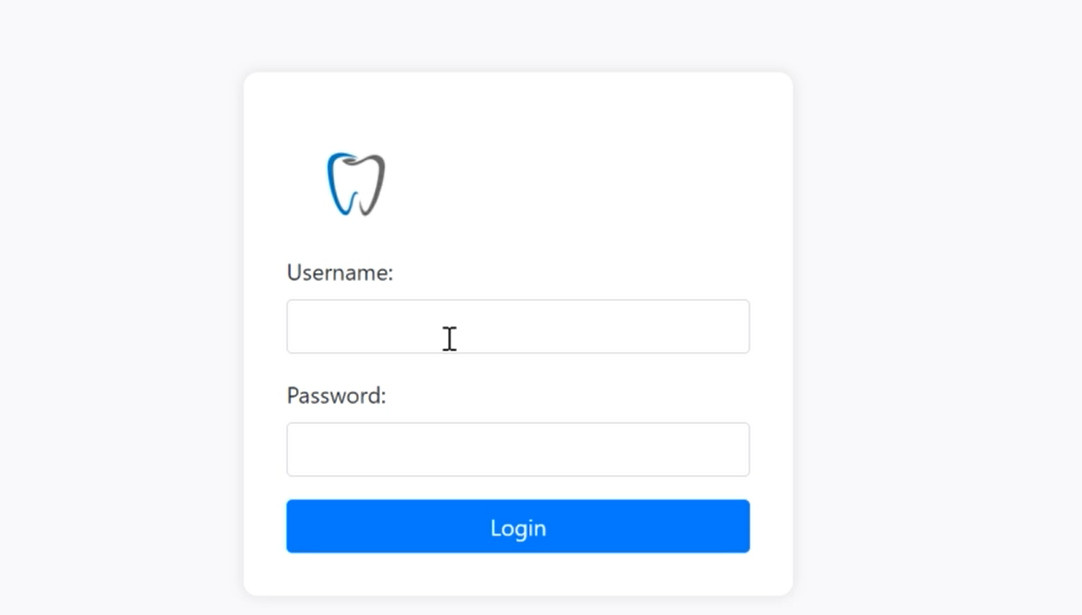
Ce Chapitre met en lumière les résultats de la programmation et du développement de la solution en présentant les interfaces de l’application.

## Interfaces

Cette application contient 3 espaces principaux,

* Espace de liste des patients,
* Espace d’authentification,
* Espace pour la consultation de dossiers médicaux
* Espace de suivi de la situation financière

Les images suivantes montrent un aperçu de notre application :



Conclusion

Nous avons examiné en détail les interfaces et les fenêtres spécifiques à chaque utilisateur. De plus on a exploré les diverses fonctionnalités offertes par ces interfaces, mettant ainsi l’accent sur les détails et les spécialités de chacune d’entre elles.

Conclusion Générale

Le développement d'une application de gestion de cabinet dentaire utilisant des technologies modernes telles que Java, Spring Boot, Spring Data, Spring MVC et Thymeleaf offre une solution innovante pour améliorer l'efficacité opérationnelle des cabinets dentaires. En automatisant les processus administratifs, en simplifiant la gestion des dossiers patients et en optimisant la planification des rendez-vous, cette application vise à offrir une expérience utilisateur améliorée tant pour les dentistes que pour les patients. L'intégration de ces technologies permet de créer une application web robuste, évolutive et sécurisée, répondant ainsi aux besoins spécifiques du domaine médical.