**Sommaire**

**Introduction Générale**

**Chapitre 1: Description du projet**

1- Contexte du projet

2- Objectifs du projet

3-Cahier de charges

**Chapitre 2: Analyse et Conception**

1- Identifier les acteurs

2- Diagramme MCD et de classes

**Chapitre 3: Réalisation**

1- Outils de développement

2- Développement

3- Présentation de l’application

**Conclusion**

**Introduction générale:**

Pour améliorer la gestion de ses patients, l’informatisation de leurs informations dans une base de données devient une necessité pour tout docteur surtout avec les dossiers en papier qui deviennent obsolètes, moins efficaces et plus couteux.

Notre application vise à faciliter la gestion d’un cabinet médical que ça soit concernant la sauvegarde et modification de leurs données ou bien la prise des rendez-vous.

Ce rapport présente l’ensemble entier des étapes du développement de cette application, divisé en 3 chapitres qui détaillent chaqun une partie du processus:

Le premier chapitre intitulé “Description du projet” est consacré à introduire le projet de manière detaillé ainsi que ses objectifs.

Le deuxième chapitre intitulé “Analyse et conception’’ vise à identifier les acteurs ainsi que de donner les diagrammes de classes et MCD sur lesquelles se base l’application.

Le troisième et dernier chapitre intitulé “Réalisation” donne les étapes de réalisation et finalisation de l’application ainsi que l’environement de développements et les outils utilisés.

En général, ce rapport facilite la navigation et la compréhension de la conception du projet final.

**Chapitre I**

**I-1- Contexte du projet:**

Lors du travail au sein du cabinet medical, le docteur se retrouve en face de plusieurs problèmes qui puissent entraver et ralentir le rhytme de son travail:

-Difficultés de gestion des rendez-vous et encombrements au sein du cabinet.

-Perte des fiches médicales, informations et rapports des patients.

-Dépassement du nombre des patients par jour (10).

Ce projet s’inscrit dans le cadre d’application du langage JAVA pour la création d’une application de gestion d’un cabinet médical à visée de résolution des problèmes surmentionnés.

**I-2- Objectifs du projet:**

-Facilitation de la gestion des patients au sein du cabinet medical.

-Sauvegarde, modification et suppression des données des patients.

-Possibilité de prendre des rendez-vous à limite régulière de 10 patients MAX.

-Stockage des données et des rendez-vous dans une base de données pour l’utilisation à long terme sans perte des informations.

-Accès à un calendrier interactif pour la confirmation des rendez-vous lors d’une date donnée.

-Possibilité d’écriture et de modification des rapports.

**I-3- Cahier de charges:**

**L’équipe de développement: -**Soufiane Essahli

-Zineb Zekhnini

**Objectifs du projet:** L’optimisation et facilité de la gestion d’un cabinet medical spécifique de manière simplifiée et optimale sans avoir recours à des fiches non électroniques.

**Plateformes:** Application de bureau.

**Fonctionnalités:**

-Sauvegarde des données.

-Accès limité à l’application par login et mot de passe.

-Interface graphique interactive et facile à naviguer.

-Facilité d’accès rapide aux données via biais de moteur de recherche.

-Historique.

**Les service tiers:** L’application doit impérativement être connectée à une base de données pour assurer un bon fonctionnement et une sauvegarde des données.

**Chapitre 2**

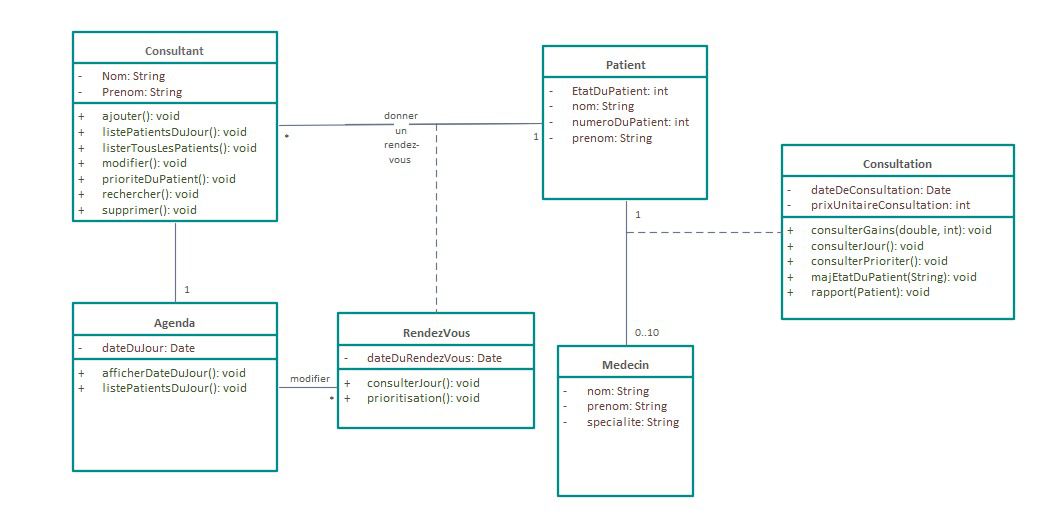
**II-1- Identifier les acteurs**

Il existe deux acteurs principaux dans notre application:

**Le docteur:** Il a une accessibilité totale du programme mais il ne se charge en général que de la rédaction et modification du rapport et la vérification et finalisation des données des patients.

**Le secrétaire/infirmier:** à comme tâche de donner et vérifier les rendez-vous pour le docteur ainsi que la sauvegarde initiale des données des patients.

**II-2- Diagramme MCD et diagramme de classes**



**Chapitre 3**

**III-1- Outils de développement:**

Nous avons utilisé les outils suivants:

**IntelliJ IDEA:**

également appelé « IntelliJ », « IDEA » ou « IDJ » est un environnement de développement intégré destiné au développement de logiciels informatiques reposant sur la technologie Java. Il est développé par JetBrains (anciennement « IntelliJ ») et disponible en deux versions, l'une communautaire, open source, sous licence Apache 2 et l'autre propriétaire, protégée par une licence commerciale. Tous deux supportent les langages de programmation Java, Kotlin, Groovy et Scala.

**SceneBuilder:**

Scene Builder est un outil interactif de conception d'interface graphique pour JavaFX. Créé par Oracle, il permet de construire rapidement des interfaces utilisateurs sans avoir besoin de (savoir) coder.

Le logiciel est décliné en deux versions : l'une (8.x) destiné à JavaFX 8 et l'autre (9.0 et +) pour JavaFX 9 et plus.

**MySQL Workbench 8.0 CE:**

MySQL Workbench est un logiciel de gestion et d'administration de bases de données MySQL créé en 2004

**MySQL:**

C’est un système de Gestion de Bases de Données (SGBD) fonctionne sous linux et Windows C’est un logiciel libre sous licence GPL, performant de point de vue stockage de données volumineuses.

**Javafx:**

JavaFX est un framework et une bibliothèque d'interface utilisateur issue du projet OpenJFX, qui permet aux développeurs Java de créer une interface graphique pour des applications de bureau, des applications internet riches et des applications smartphones et tablettes tactiles.

Créé à l'origine par Sun MicroSystems, puis développé par Oracle après son rachat et ce, jusqu'à la version 11 du JDK, c'est depuis lors à la communauté OpenJFX que revient la poursuite de son développement1.

Cette bibliothèque a été conçue pour remplacer Swing et AWT, qui ont été développés à partir de la fin des années 90, pour pallier les défauts de ces derniers et fournir de nouvelles fonctionnalités (dont le support des écrans tactiles).

Le cycle de sortie d'une nouvelle version de JavaFX correspond à celui de Java, soit tous les 6 mois.

**Java:**

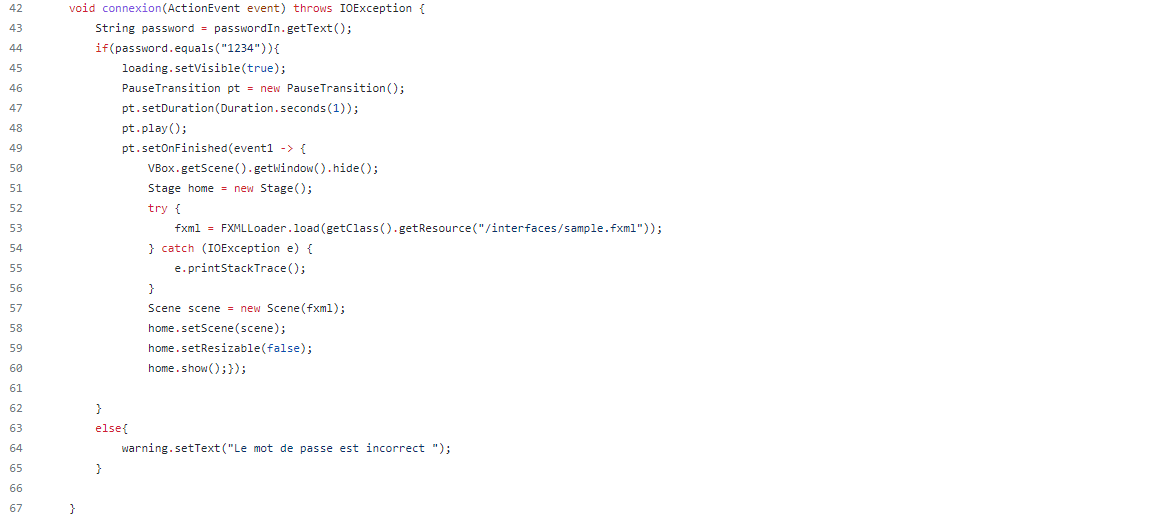
Java est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld.

La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java.

Une particularité de Java est que les logiciels écrits dans ce langage sont compilés vers une représentation binaire intermédiaire qui peut être exécutée dans une machine virtuelle Java (JVM) en faisant abstraction du système d'exploitation.

**III-2- Développement:**

*Methode de connexion:*



Ligne 42: Initialisation de la méthode.

Ligne 43: Saisie du mode passe par l’utilisateur

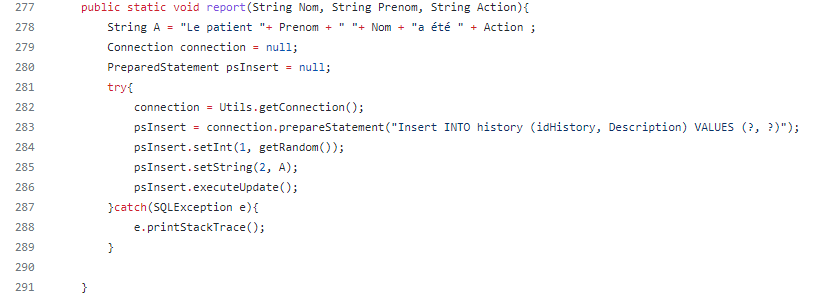
Ligne 44: Condition de connexion si l’entrée de l’utilisateur est égale au mot de passe fixé.

Ligne 45-48: Initialization de l’animation de chargement et de transition de pause pour 1s avant la fin de la connexion.

Ligne 49-62: Masquage de l’interface de login et apparition de l’interface principale (menu de l’application).

Ligne 63-65: Au cas de l’incompatibilité du mot de passe entré par l’utilisateur, il y’a affichage d’un avertissement.

*Methode pour l’Historique:*

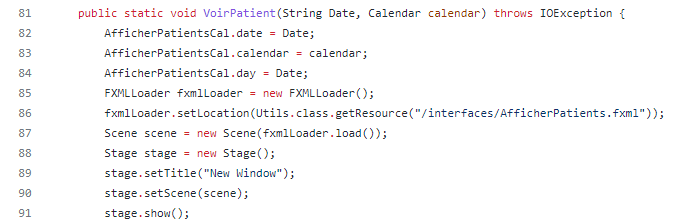


Ligne 277: initialization de la méthode.

Ligne 278: Creation du String qui détaille l’opération appliquée sur le malade (ajout, suppression, modification...)

Ligne 279-289: connexion à la database et insertion du String crée dans la table de Description pour son affichage dans la page historique et assignation aléatoire d’une valeur pour l’idHistory.

*Methode pour visualiser les patients qui ont rendez vous en une certaine date:*

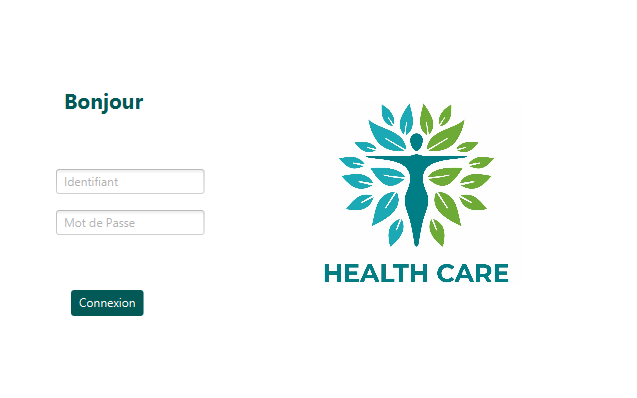


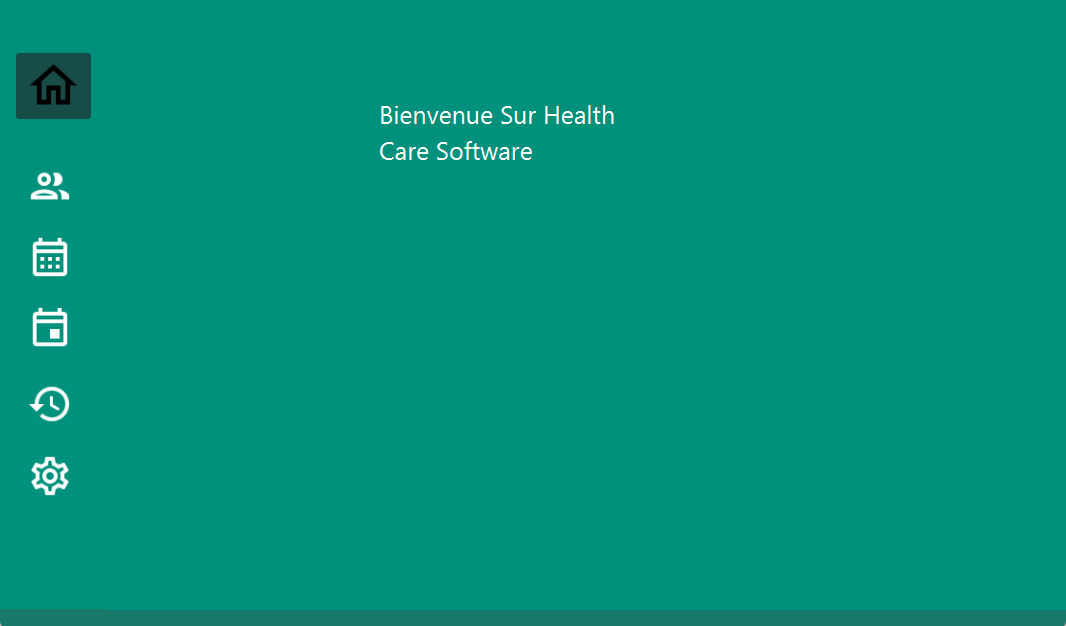
Ligne 81: Initialization de la méthode ‘VoirPatient’ avec les paramètres String et Calendar.

Ligne 82-84: Assignation des valeurs aux paramètres d’après la classe ‘AfficherPatientsCal’

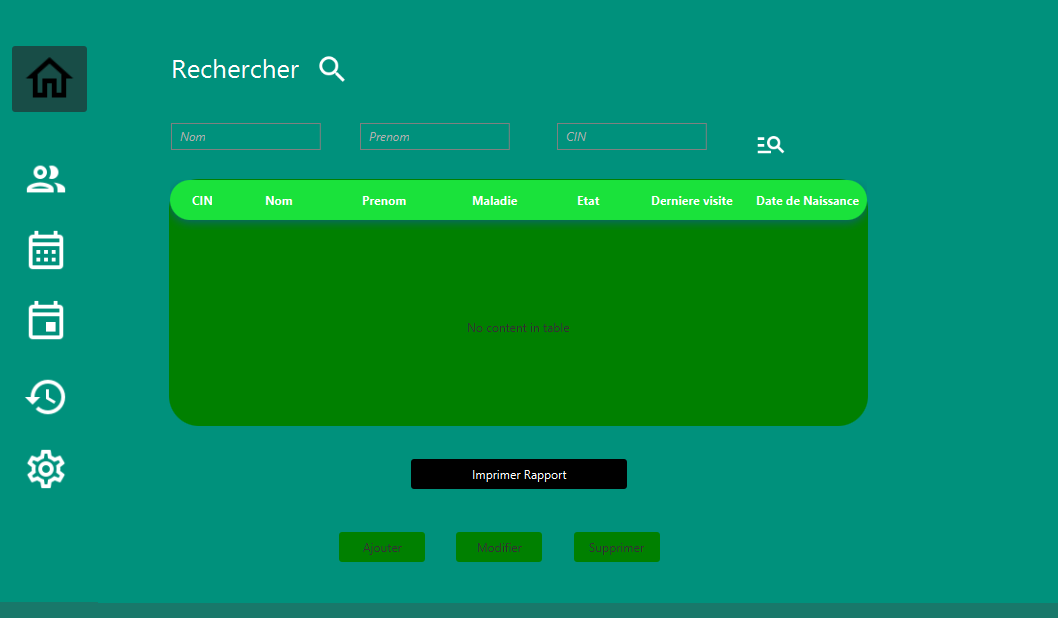
Ligne 85-91: Apparition d’une nouvelle fenêtre qui détaille les patients qui ont rendez-vous lors du jour spécifié.

**III-3- Présentation du projet**

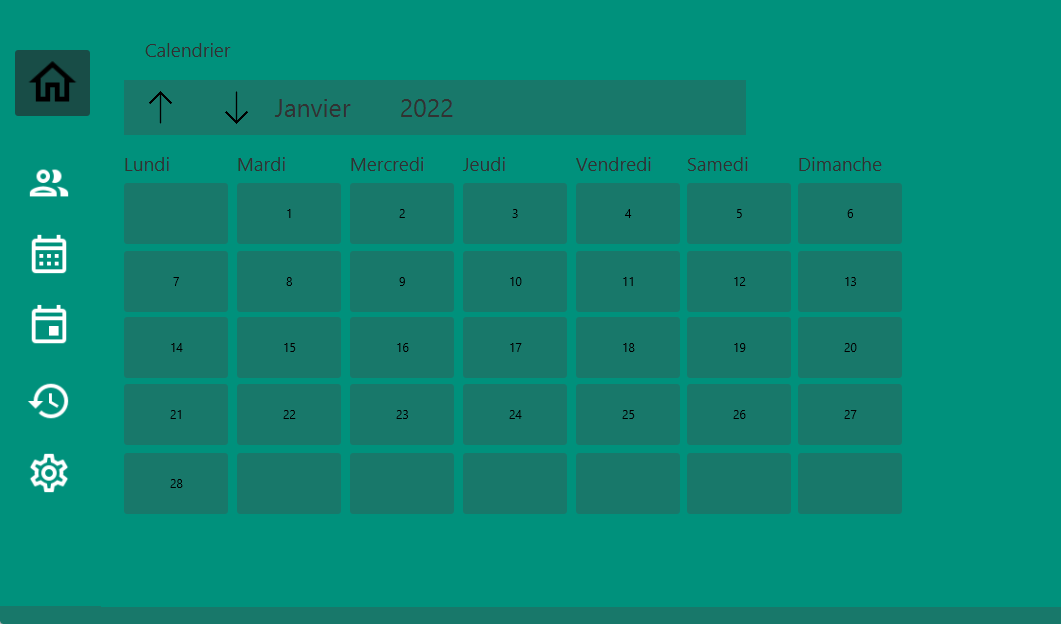
C’est la première interface qui s’affiche après le lançement de l’application. L’utilisateur doit entrer ses identifiants pour se connecter et accéder à l’interface suivante.



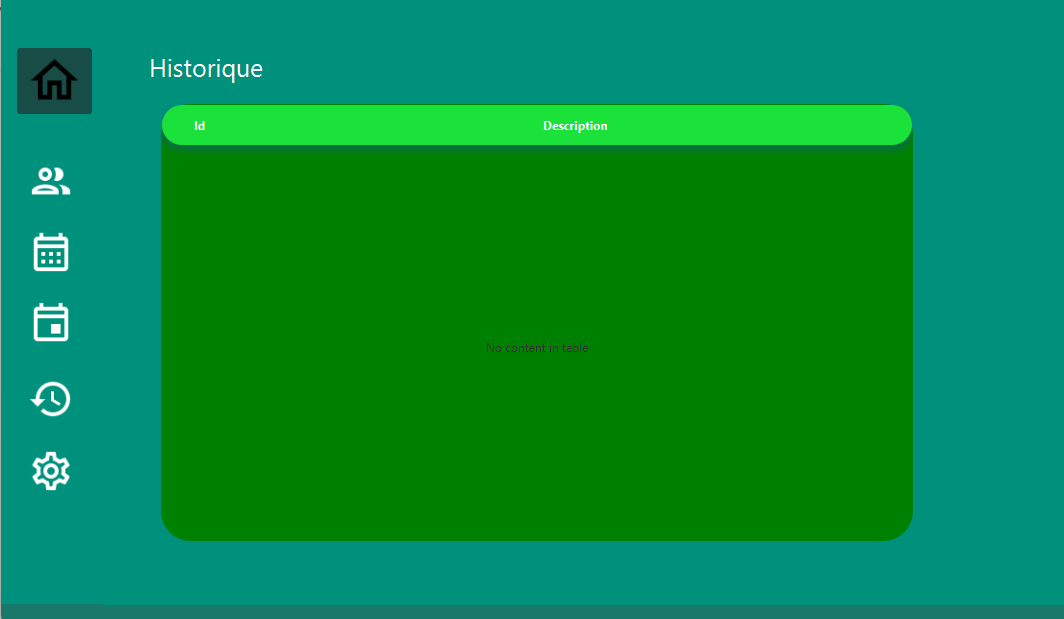
C’est la seconde interface après la connexion. Ici l’utilisateur peut accéder aux autres fonctionnalités du programmes: Visualisation de la liste de patient, calendrier, prise de rendez-vous, Historique et paramètres.



C’est l’interface des patients, ici l’utilisateur peut soit ajouter, modifier, supprimer ou bien afficher les informations des patients sauvegardés dans la base de données. Il peut ainsi utiliser le moteur de recherche pour rechercher un patient spécifique via les critères: Nom, Prénom et/ou CIN.



Celle là, c’est l’interface calendrier où l’utilisateur peut afficher la liste des patients qui ont rendez-vous durant un jour spécifique.



D’après cette interface, l’utilisateur à droit de voir toutes les actions et modifications qui ont eu lieu sur les patients.

**Conclusion**

A travers ce rapport, nous avons pû présenter l’environement logiciel de notre application, les étapes de développement, les fonctionnalités ainsi que les outils et cibles du projet. Ensuite nous avous illustré notre rapport avec les images prises directement de l’application pour faciliter la compréhension du fonctionnement de cette application de gestion d’un cabinet médical.