

Gestion de Clinique

Examen Final - Mini-Projet Java

Réalisé par :

GUENDOZ Ayman

AZIRAR Assia

Encadré Par :

Pr. HAMIDA Soufiane

Table des matières

1. Introduction	4
Présentation du projet	4
Objectifs du projet	4
2. Analyse et conception.....	4
Identification des besoins fonctionnels.....	4
Modèle conceptuel de données.....	4
Structure de la base de données.....	4
Table patients :.....	4
Table rendezvous :	5
Diagramme use case	5
Figure : diagramme use case	5
Diagramme de classe	6
Figure : diagramme de classe.....	6
Diagramme de séquence	7
Figure : diagramme de séquence (scénario de authentification)	7
3. Architecture technique	7
Pattern MVC utilisé	7
Organisation du code source.....	8
Technologies utilisées.....	8
4. Fonctionnalités implémentées.....	8
Gestion des patients.....	9
Profil utilisateur	9
5. Interface utilisateur	9
Création de compte :.....	9
Figure 1 : Création de compte.....	9
Écran de connexion	10
Figure 2 : Écran de connexion	10
Tableau de bord	10
Figure 3:Tableau de bord	10
Gestion des rendez-vous.....	11
Figure 4 : Gestion des rendez-vous	11
Gestion des patients.....	12
Figure 5 : Gestion des patients.....	12
Profil utilisateur	12
Figure 6 : Profil utilisateur	12
6. Défis rencontrés et solutions	13
Défis techniques	13
Solutions implémentées.....	13
7. Conclusion	13

Bilan du projet.....	13
Compétences acquises	13
Perspectives d'amélioration	14
8. Annexes	14
Guide d'installation et d'utilisation.....	14
Prérequis	14
Installation	14
Utilisation	15

1. Introduction

Présentation du projet

Ce rapport présente une application de gestion de clinique développée en Java dans le cadre de l'examen final du module de programmation Java. Le système permet de gérer les patients et leurs rendez-vous au sein d'une clinique médicale, en offrant une interface conviviale et des fonctionnalités complètes de gestion des données.

Objectifs du projet

- Créer une application complète avec interface graphique JavaFX
- Implémenter les opérations CRUD (Create, Read, Update, Delete)
- Établir une connexion à une base de données MySQL
- Suivre une architecture MVC (Modèle-Vue-Contrôleur)
- Sécuriser l'accès à l'application via un système d'authentification

2. Analyse et conception

Identification des besoins fonctionnels

L'application répond aux besoins suivants :

Authentification des utilisateurs
Gestion des patients (ajout, modification, suppression, recherche)
Gestion des rendez-vous (planification, consultation, modification, annulation)
Consultation des informations personnelles et changement de mot de passe

Modèle conceptuel de données

Notre modèle de données est composé de trois entités principales :

- Utilisateurs : représente les membres du personnel médical avec accès au système
- Patients : stocke les informations relatives aux patients de la clinique
- Rendez-vous : associe un patient à une date et heure spécifique avec description

Structure de la base de données

La base de données "clinique_db" contient les tables suivantes :

Table patients :

id (int, clé primaire, auto-incrémenté)

nom (varchar) prenom (varchar)

date_naissance (date)

telephone (varchar)

email (varchar)

Table rendezvous :

id (int, clé primaire, auto-incrémenté)

patient_id (int, clé étrangère)

date_rdv (date)

heure_rdv (time)

description (text)

Table utilisateurs :

id (int, clé primaire, auto-incrémenté)

nom (varchar)

email (varchar, unique)

mot_de_passe (varchar, hashé)

La table "rendezvous" est liée à la table "patients" par une contrainte de clé étrangère, avec suppression en cascade pour maintenir l'intégrité des données.

Diagramme use case

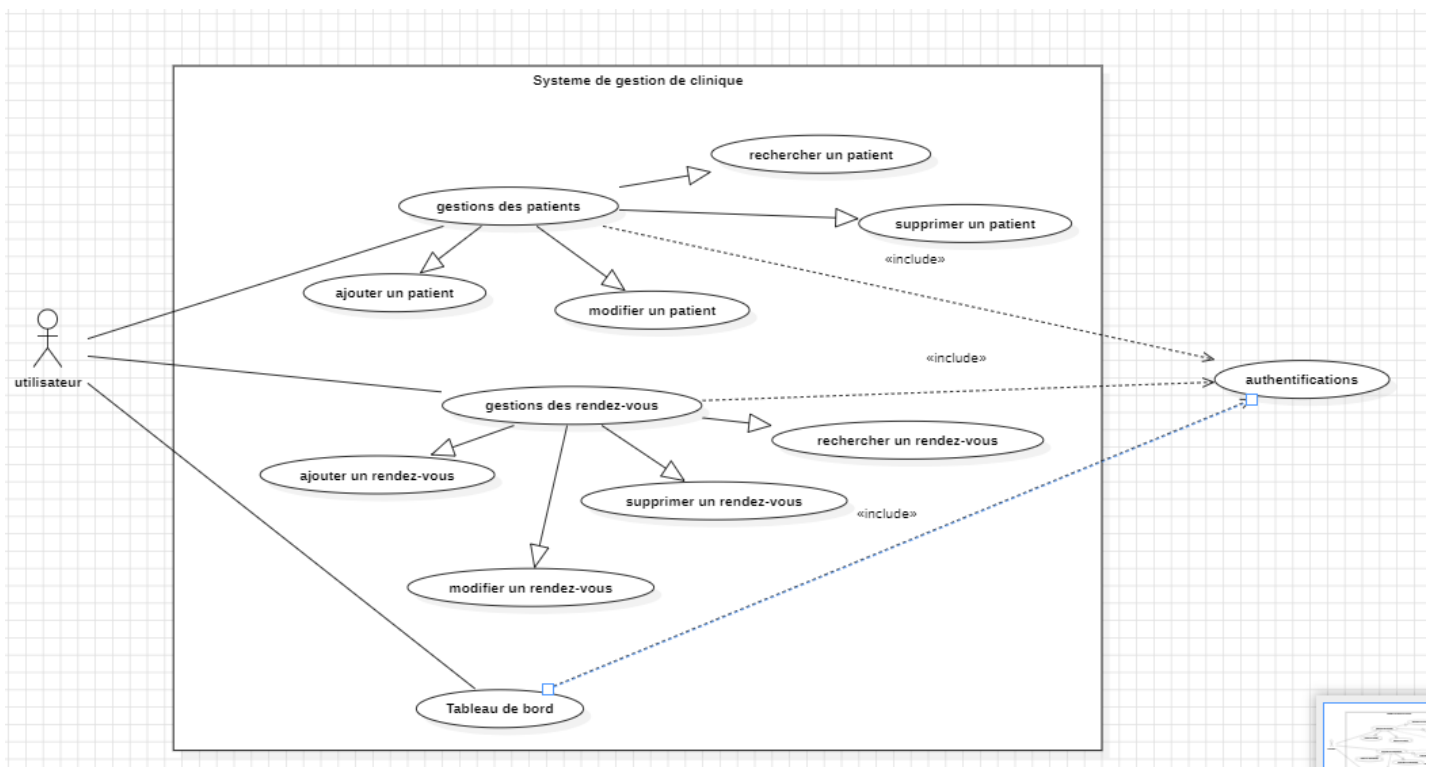


Figure : diagramme use case

Diagramme de classe

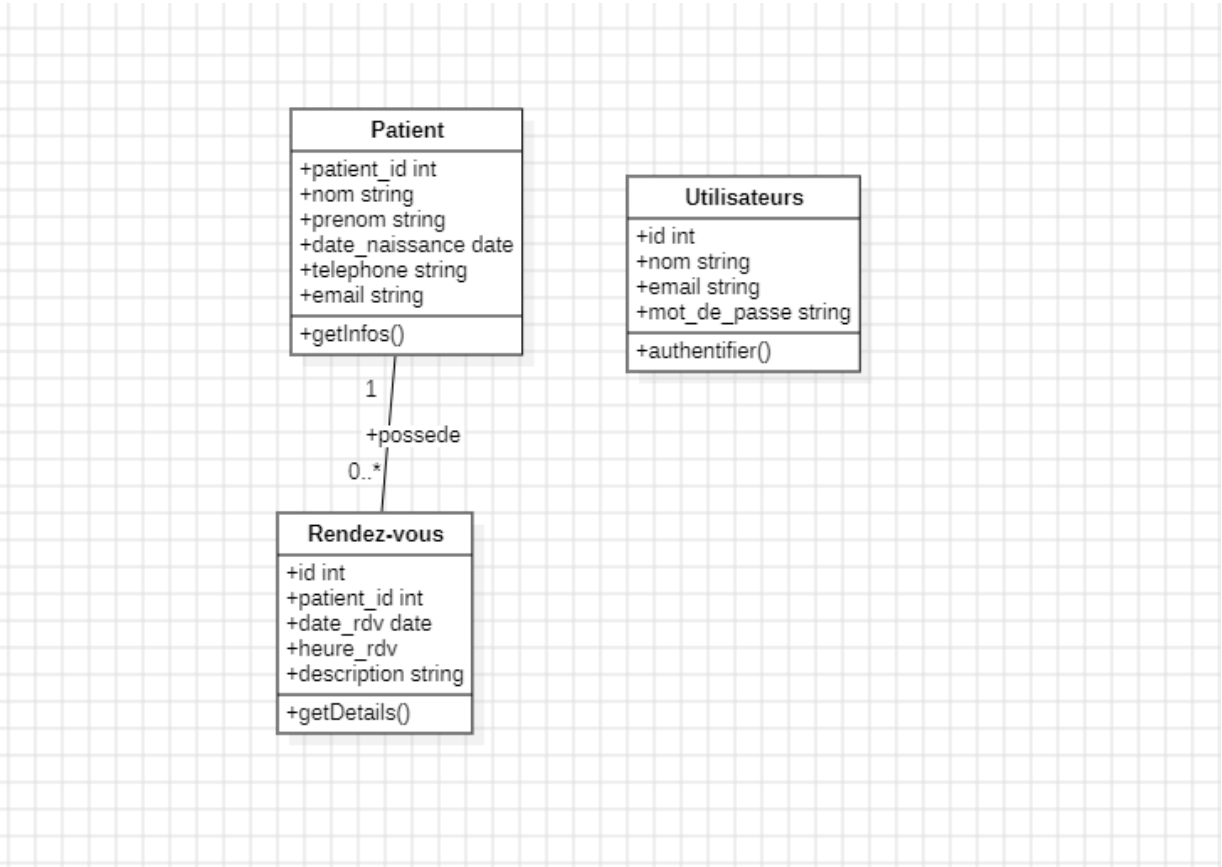


Figure : diagramme de classe

Diagramme de séquence

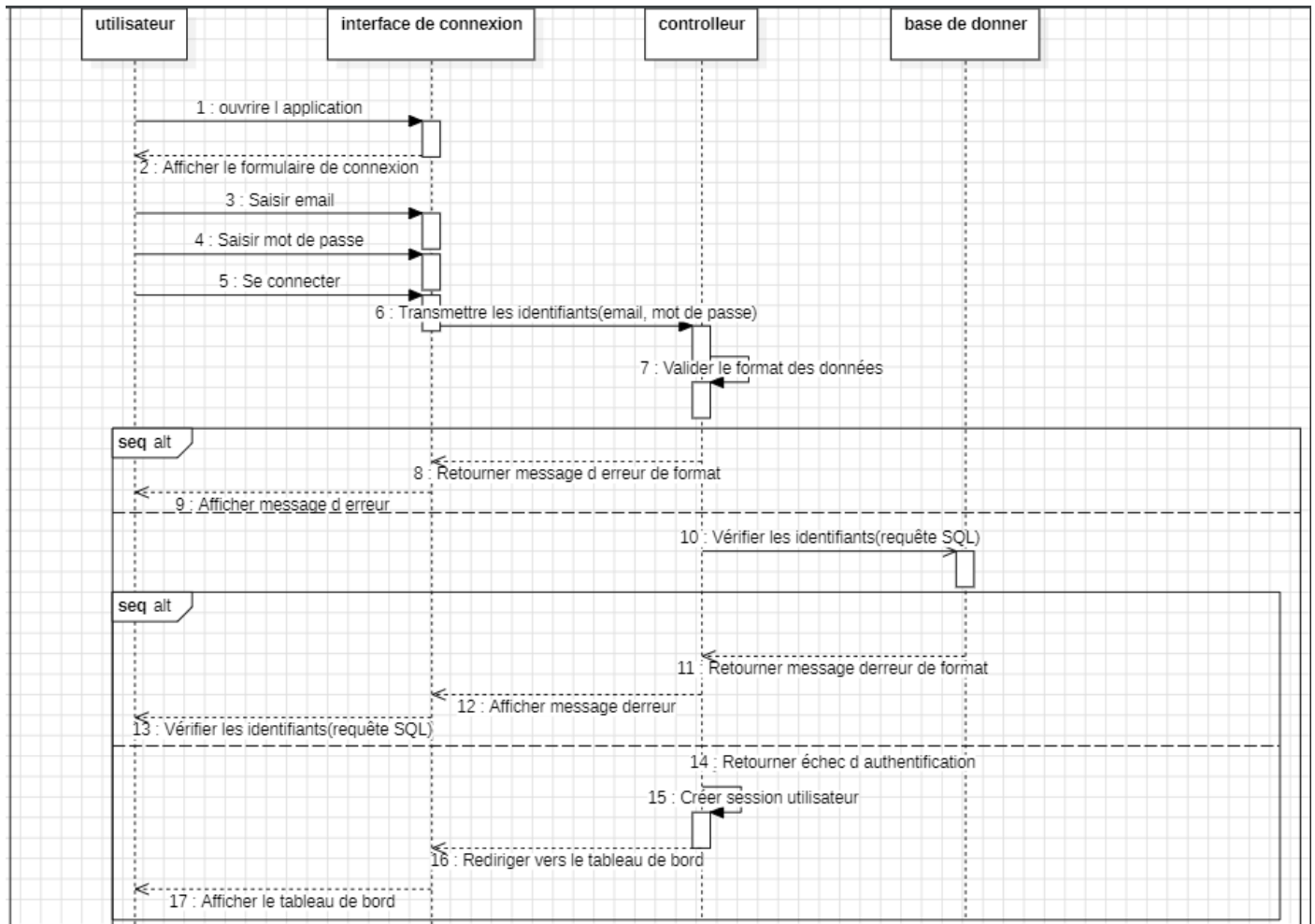


Figure : diagramme de séquence (scénario d'authentification)

3. Architecture technique

Pattern MVC utilisé

L'application suit le pattern Modèle-Vue-Contrôleur :

Modèle : Classes représentant les entités (Patient, RendezVous, Utilisateur) et les classes DAO pour l'accès aux données

Vue : Interfaces utilisateur définies en FXML

Contrôleur : Classes gérant les interactions entre la vue et le modèle

Organisation du code source

Le projet est structuré selon les recommandations du sujet :

src/

main/

java/

controllers /

com /

models /

dao /

daoImpl /

utils /

views /

Main /

Resources /

Module-info.java

Technologies utilisées

JavaFX : Pour l'interface

MySQL : Pour la base de données

JDBC : Pour la connexion à la base de données

CSS : Pour le style de l'interface utilisateur

SHA-256 : Pour le hachage des mots de passe

4. Fonctionnalités implémentées

Système d'authentification

Connexion avec email et mot de passe

Création de compte utilisateur

Protection des mots de passe par hachage SHA-256

Déconnexion sécurisée

Gestion des patients

L'application permet une gestion complète des patients avec les opérations CRUD :

Create : Ajout de nouveaux patients avec leurs informations personnelles

Read : Affichage de la liste des patients et recherche par nom/prénom

Update : Modification des informations des patients existants

Delete : Suppression des patients (avec suppression en cascade de leurs rendez-vous)

Profil utilisateur

L'application permet à l'utilisateur connecté de :

Consulter ses informations personnelles

Modifier son mot de passe

5. Interface utilisateur

Création de compte :

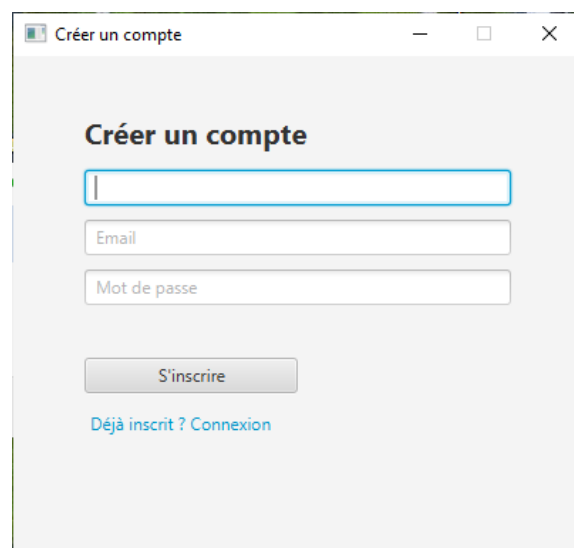
The image shows a web browser window titled "Créer un compte". Inside the window, there is a form with the heading "Créer un compte". The form contains three input fields: a text field for the name, an "Email" field, and a "Mot de passe" (Password) field. Below these fields is a button labeled "S'inscrire" (Sign up). At the bottom of the form, there is a link that says "Déjà inscrit ? Connexion" (Already registered? Login).

Figure 1 : Création de compte

L'écran de création de compte comprend :

Champs pour le nom, email et mot de passe

Bouton d'inscription

Lien vers la page de connexion

Écran de connexion

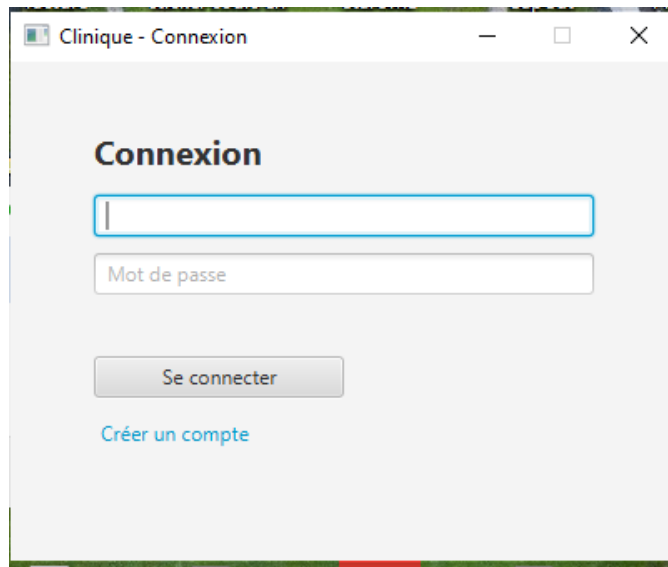


Figure 2 : Écran de connexion

L'interface de connexion présente un design épuré avec :

Champs pour l'email et le mot de passe

Bouton "Se connecter"

Lien vers la création de compte

Tableau de bord

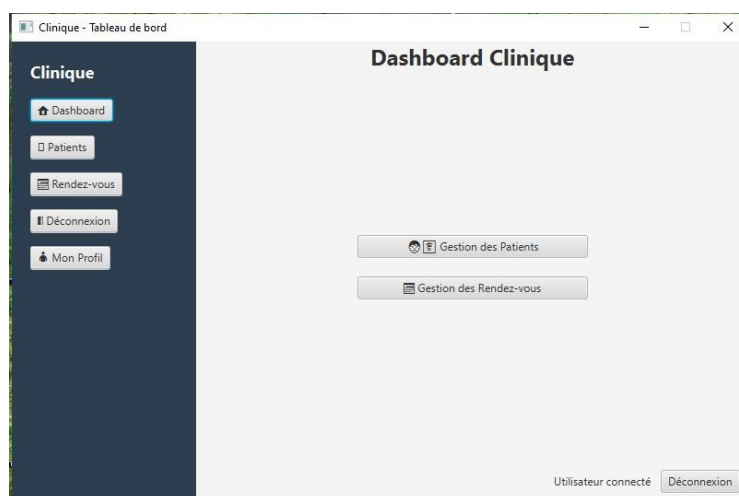


Figure 3 : Tableau de bord

Le tableau de bord sert de point central à l'application avec :

Menu latéral pour la navigation

Accès rapide aux principales fonctionnalités

Indication de l'utilisateur connecté

Bouton de déconnexion

Gestion des rendez-vous

Nom	Prénom	Date	Patient	Heure	
souhaila	omri	2025-05-25	souhaila omri	15:00	test
guendouz	ayman	2025-04-26	guendouz ayman	13:00	tetete
azirar	assia	2025-04-25	azirar assia	15:00	testt
hamidi	salma	2025-04-16	hamidi salma	14:00	zzz

Figure 4 : Gestion des rendez-vous

L'interface de gestion des rendez-vous permet :

Ajout d'un nouveau rendez-vous avec sélection de patient

Tableau récapitulatif des rendez-vous existants

Fonctions de recherche, modification et suppression

Visualisation claire avec informations essentielles (patient, date, heure)

Gestion des patients

ID	Nom	Prénom	Téléphone	Email
2	hamidi	salma	0123456789	salma@gmail.com
3	azirar	assia	01234567679	assia@gmail.com
4	guendouz	ayman	1234567890	testa@gmail.com
5	souhaila	omri	1234567890	souhaila@gmail.com

Figure 5 : Gestion des patients

L'interface de gestion des patients propose :

Formulaire pour l'ajout/modification

Recherche par nom ou prénom

Tableau des patients avec toutes leurs informations

Boutons d'actions (ajouter, modifier, supprimer)

Profil utilisateur

Nom : ayman
Email : ayman@gmail.com

Changer le mot de passe

Ancien mot de passe
Nouveau mot de passe
Confirmer le nouveau mot de passe

Mettre à jour

Figure 6 : Profil utilisateur

La page profil permet à l'utilisateur de :

Consulter ses informations personnelles

Changer son mot de passe avec confirmation

6. Défis rencontrés et solutions

Défis techniques

1. Gestion de la sécurité : Implémentation d'un système de hachage SHA-256 pour les mots de passe.
2. Relations entre entités : Gestion de la relation entre patients et rendez-vous avec mise à jour en cascade.
3. Validation des données : Mise en place de contrôles pour éviter les chevauchements de rendez-vous.

Solutions implémentées

1. Création d'une classe utilitaire pour le hachage sécurisé des mots de passe
2. Utilisation de contraintes de clé étrangère avec suppression en cascade
3. Implémentation de validateurs de formulaires côté client et de contraintes côté serveur

7. Conclusion

Bilan du projet

L'application développée répond à tous les critères demandés dans le cadre du mini-projet :

- Interface graphique complète avec JavaFX
- Connexion à une base de données MySQL
- Opérations CRUD sur toutes les entités
- Architecture MVC bien structurée
- Documentation du code

Compétences acquises

Ce projet a permis de consolider et d'approfondir plusieurs compétences :

- Conception et développement d'applications JavaFX
- Gestion des bases de données relationnelles
- Implémentation du pattern MVC

- Sécurisation d'une application (authentification, hachage)
- Développement d'interfaces utilisateur ergonomiques

Perspectives d'amélioration

Plusieurs améliorations pourraient être apportées au système :

- Ajout d'un système de gestion des médecins et spécialités
- Implémentation d'un système de notification pour les rendez-vous
- Développement d'un module de reporting et statistiques
- Exportation des données vers Excel ou PDF
- Amélioration de l'interface utilisateur (thèmes, responsivité)

8. Annexes

Script SQL de création de la base de données

[Voir Fichier base de donnée]

Guide d'installation et d'utilisation

Prérequis

Java JDK 11 ou supérieur

MySQL 5.7 ou supérieur

JavaFX SDK

Installation

1. Créer la base de données à l'aide du script SQL fourni
2. Configurer les paramètres de connexion à la base de données dans le fichier de
3. Compiler et exécuter l'application

Utilisation

1. Créer un compte utilisateur ou se connecter avec un compte existant
2. Naviguer entre les différentes fonctionnalités via le menu latéral
3. Gérer les patients et les rendez-vous selon les besoins
4. Se déconnecter à la fin de la session pour des raisons de sécurité