



Module : projet pluridisciplinaire

Rapport de projet :

Application pour gestion d'un cabinet
médical

Réalisé par : RACHDI Chems eddine
Immatriculation : 181833002604
Section : B
Groupe : B3

System d'exploitation : Windows

Langage utilisé : Java + SQL.

Ide : Eclipse.

SGBD : SQLite.

Introduction :

La principale mission du projet fut de réaliser la conception d'une application de gestion d'un cabinet médical qui offre au gérant du cabinet un outil pour gérer les informations de son établissement, aux médecins de consulter leurs rendez-vous et de donner des ordonnances, et aux réceptionnistes de réserver des rendez-vous plus facilement et de gérer les patients. Mais encore, elle permet aux pharmaciens de gérer les médicaments (ajouté).

Structure du rapport :

Ce rapport est composé de 3 chapitres :

1. Objectifs du projet et une description du cahier de charge.
2. Modélisation du système.
3. Réalisation de l'application.

Chapitre 1 :

Cahier de charge :

L'application a 3 types d'utilisateurs :

1. Réceptionniste.
2. Docteur.
3. Pharmacien.

A. Réceptionniste :

- Gestion des patients (ajout, modification, suppression).
- Gestion des rendez-vous (création, annulation).
- Création de nouveaux utilisateurs.
- Accès à la liste des médecins.

B. Docteur :

- Gestion des médecins (ajout, modification, suppression).
- Accès aux rendez-vous (chaque médecin n'a d'accès qu'à ses propres rendez-vous).
- Création d'ordonnances.
- Accès à la liste des médecins.

C. Pharmacien :

- Accès aux ordonnances.
- Accès à l'inventaire à travers les ordonnances.
- Accès aux ventes (liste de vente des médicaments).

Chapitre 2 : Modélisation du système

Authentification :

Lorsque l'utilisateur veut accéder à l'application il doit d'abord saisir son nom d'utilisateur, son mot de passe et son type d'utilisateur (exemple : username: admin, password: admin, usertype: receptionnist), si ils sont corrects le system affichera la page d'accueil, sinon le system affichera un message d'erreur.

Gestion de patients :

- **L'ajout d'un patient :**

Après la d'un dossier d'un nouveau patient, la réceptionniste va saisir les données de ce dernier (nom, téléphone, adresse) et valider l'opération.

- **La suppression d'un patient :**

Après la sélection du patient concerné par cette opération, la réceptionniste peut le supprimer.

- **Modification d'un patient :**

En cas de saison de données erronées la réceptionniste va sélectionner les données à modifier et valider les modifications.

la création de médicaments se fait de la même manière

Nous avons modélisé comme classes les éléments suivants :

-Channel (rendez-vous)

-Doctor

-Inventory

-Item (médicaments).

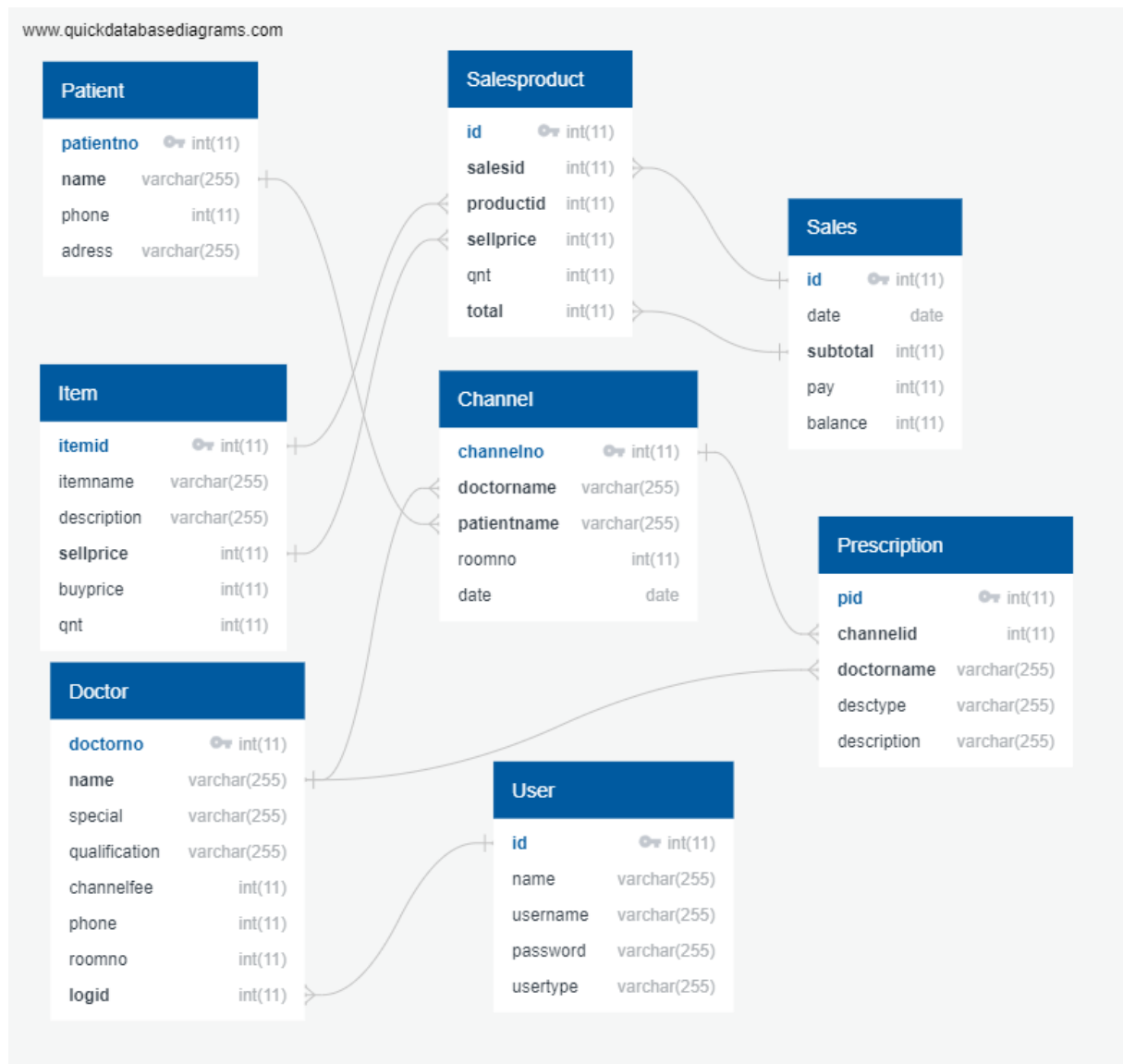
-Patient

-Prescription (ordonnance)

-Report (ventes)

-user

Modèle relationnel de la base de données :



Cette modélisation est une étape nécessaire et importante pour la création de la base de données et la création de cette application.

Chapitre 3 : Réalisation (fonctionnement)

Ce chapitre est consacré à l'implémentation de l'application qui s'appuie sur la modélisation précédente, pour l'implémentation j'ai utilisé le langage de programmation JAVA et l'environnement de gestion de base de données SQLite.

Description de l'application :

Page d'authentification (login) :

Cette page permet à l'utilisateur de s'identifier pour pouvoir accéder aux autres interfaces de l'application, si l'utilisateur ne s'est pas identifié correctement le system affichera un message d'erreur.

Identifiants pour accéder à l'application entant que réceptionniste :

Username : admin

Password : admin

Usertype : receptionist

Page d'accueil (main) :

Cette page permet d'accéder aux pages restantes de l'application selon l'utilisateur connecté, et en vous identifiant en tant qu'un des 3 différents types d'utilisateurs vous aurez accès à :

Réceptionniste :

- [Page Patient :](#)
La liste des patients et vous pourrez ajouter, modifier ou supprimer un patient.
- [Page Rendez-vous \(Créer rendez-vous\) :](#)
La liste des rendez-vous et vous pourrez en créer des nouveaux ou annuler les rendez-vous existants.
- [Page Utilisateur \(Créer utilisateur\) :](#)
La création d'un nouvel utilisateur.
- [Page Docteur \(Voir docteur\) :](#)
La liste des médecins.

Docteur :

- [Page Médecin :](#)
La liste des médecins et vous pourrez ajouter, modifier ou supprimer un médecin.
- [Page rendez-vous \(Voir rendez-vous\) :](#)
La liste des rendez-vous (chaque médecin n'a accès qu'à ses propres rendez-vous), Et en choisissant un rendez-vous, le médecin peut créer une ordonnance en appuyant sur le bouton « Prescription ».
- [Page Docteur \(Voir docteur\) :](#)
La liste des médecins.

Pharmacien :

- [Page ordonnances \(Voir prescription\) :](#)
La liste des ordonnances, en choisissant une ordonnance vous pourrez accéder à l'inventaire.

- **Page Inventaire :**

Après avoir rentré le code du médicament (exemple : 1 pour doliprane) et confirmé en ayant appuyé sur la touche « entrer », le nom du médicament sera automatiquement affiché et vous n'aurez qu'à rentrer la quantité à vendre (si la quantité saisie est supérieure à la quantité disponible l'application affichera un message d'erreur), après avoir ajouté la vente en appuyant sur le bouton « add » le prix total sera automatiquement affiché et vous devrez taper le montant payé par le patient puis appuyer sur le bouton « update » pour que la vente soit enregistrée et la balance sera automatiquement affichée.

- **Page médicaments (Créer médicament) :**

La liste des médicaments, vous pourrez ajouter, modifier ou supprimer un médicament.

- **Page ventes :**

La liste des médicaments vendus.

Et chaque utilisateur a un bouton « Logout » qui lui permet de se déconnecter et de revenir à la page d'authentification.

Conclusion :

La réalisation de ce projet m'a permis d'améliorer mes connaissances en programmation et leur mise en application, Aussi cela m'a permis de découvrir les bases de données et leur utilisation.