Pharmacy Management System Documentation

2nd March 2025
Midterm Sprint

Khoa Pham

User Documentation

Introduction

The Pharmacy Management System helps manage patients, doctors, medications, and prescriptions. It allows for adding, editing, deleting, and searching for records, as well as generating reports and restocking drugs. The system is built with Object-Oriented Programming (OOP) principles and can be accessed through a menu interface or by using specific functions.

Overview

The Pharmacy Management System is designed to help pharmacies manage medications, prescriptions, doctors, and patients efficiently. The system allows user to:

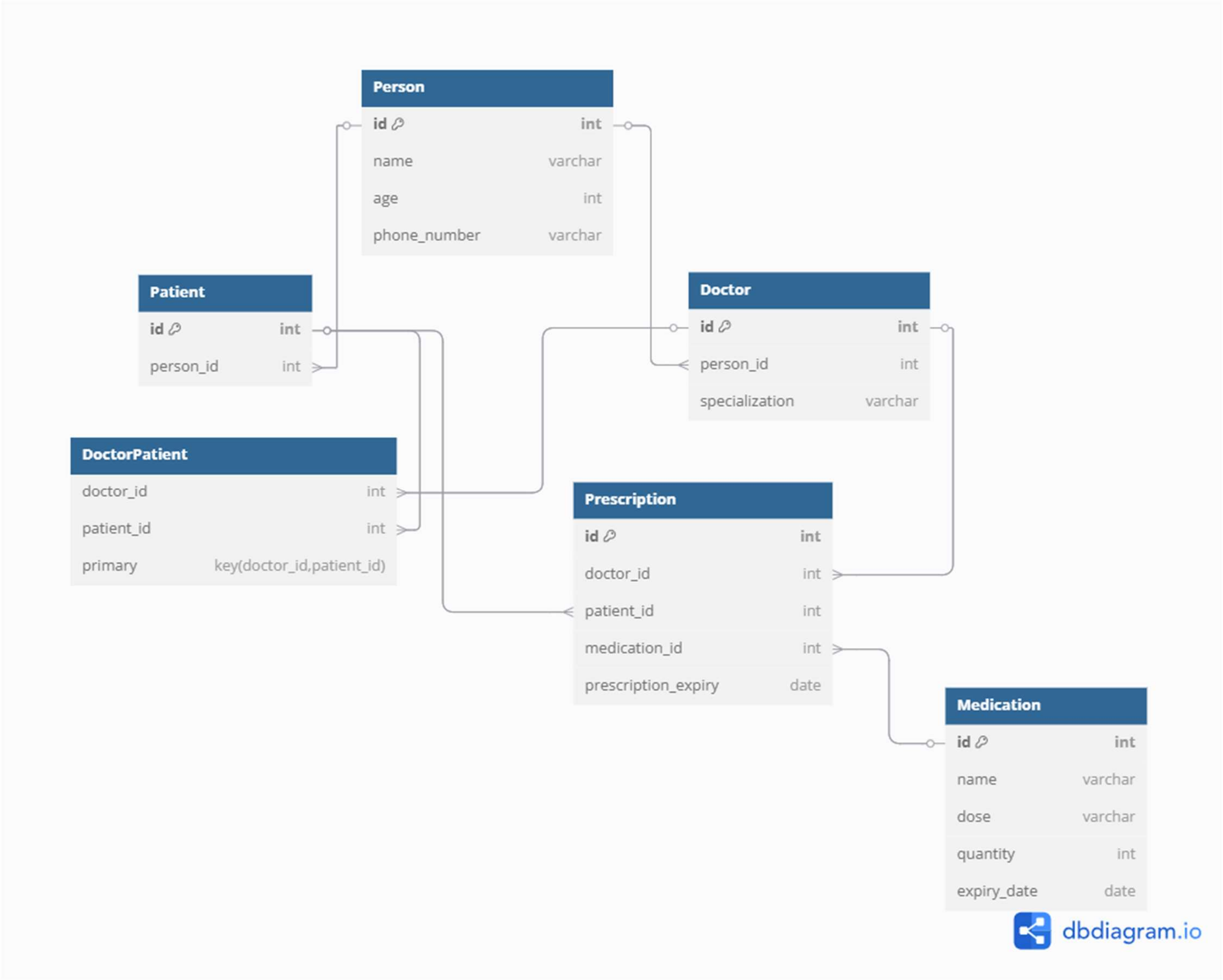
+Track patient prescriptions.
+Manage medications, including restocking and expiration checks.
+Assign patients to doctors.
+Search for doctors, medications, and patients.
+Generate reports for doctors, patients, medications, and prescriptions.

Class Descriptions

_Person (Super Class): The base class for Doctor and Patient, containing attributes: ID, Name, Age, and Phone Number.
_Patient: Represents a patient. Contains a list of medications and prescriptions.
_Doctor: Represents a doctor. Contains a specialization and a list of patients they manage.
_Medication: Represents a medication with attributes like ID, Name, Dose, Quantity, and Expiry Date.

_Prescription: Represents a prescription linking a doctor, patient, and medication. Includes an expiry date.
_ MedicationTrackingSystem: The main system handling searching, managing, and generating reports for medications, doctors, and patients.

Class Diagram



How to use the system

1. Run The application from the Menu.java
2. Choose an option from the menu:

1/ Add a new patient
2/ Add a new doctor
3/ Add a new medication
4/ Generate a system report
5/ Check for expired medications
6/ Process a new prescription
7/ Print all prescriptions for a doctor
8/ Restock medications
9/ Print all prescriptions for a patient
10/ Exit

Conclusion

This system simplifies pharmacy management by streamlining tasks such as managing patients, prescriptions, and medication stocks. It's user-friendly and can be expanded for future needs.

Development Documentation

Introduction

The Development Documentation explains the technical aspects of the Pharmacy Management System, including its code structure, database design, and build process. It provides instructions for compiling, running, and extending the project.

Directory Structure

```
_Sprint2_PharmacyManagementSystem
    _src (contains all java files)
        +Person.java
        +Patient.java
        +Doctor.java
        +Medication.java
        +Prescription.java
        +MedicationTrackingSystem.java
        +Menu.java
    _docs (contains documentation files)
        +PharmacyManagementSystem.pdf
    _README.md
```

How to compile the Project

1/ Using Terminal (Mac/Linux) or Command Prompt(Windows):

```
    Javac -d bin src/*.java src/Menu.java
    Java -cp bin src.Menu
```

2/ Using an IDE (Vs Code, IntelliJ, Eclipse)

    +Open the project in your IDE
    +Run Menu.java

Required Dependencies

1/ Java Development Kit (JDK 11+)

2/ Git (for version control and Github submission)

Entity-Relationship (ER) Diagram

1/ Person: Represents a base class for both Patient and Doctor
*Attributes:
+id: Unique identifier for the person
+name: Name of the person
+age: Age of the person
+phone_number: Contact number of the person

2/ Patient: Represents a patient in the system.
*Attributes:
+id: Unique identifier for the patient.
+person_id: Foreign key linking to the Person table.

3/ Doctor: Represents a doctor in the system.
*Attributes:
+id: Unique identifier for the doctor.
+person_id: Foreign key linking to the Person table.
+specialization: The doctor’s specialization field (e.g., cardiologist, neurologist).

4/Medication: Represents the medication available in the pharmacy.
*Attributes:
+id: Unique identifier for the medication.
+name: Name of the medication.
+dose: Dose for the medication.
+quantity: Quantity in stock.
+expiry_date: Expiry date of the medication.

5/ Prescription: Represents a prescription issued by a doctor for a patient.
*Attributes:
+id: Unique identifier for the prescription.
+doctor_id: Foreign key to the Doctor table.
+patient_id: Foreign key to the Patient table.
+medication_id: Foreign key to the Medication table.

+prescription_expiry: Expiry date for the prescription.

6/ DoctorPatient: Represents a many-to-many relationship between Doctor and Patient.

*Attributes:

+doctor_id: Foreign key to the Doctor table.

+patient_id: Foreign key to the Patient table.

Entity-Relationship Diagram Explanation

1/ Person:

+Person is a super class for both Doctor and Patient.

+Person contains general attributes like id, name, age, and phone_number.

2/ Patient:

+Patient has a one-to-one relationship with Person (linked via person_id).

+A Patient can have many prescriptions and is linked to many doctors.

3/ Doctor:

+Doctor has a one-to-one relationship with Person (linked via person_id).

+A Doctor can have many patients (through DoctorPatient) and can issue many prescriptions.

4/ Medication:

+A Medication has a one-to-many relationship with Prescription (a medication can be prescribed in many prescriptions).

+It is linked by medication_id.

5/ Prescription:

+A Prescription is linked to a specific Doctor, Patient, and Medication.

+It contains a prescription_expiry date and can have many prescriptions per doctor-patient pair.

6/ DoctorPatient:

+This is a many-to-many relationship between Doctor and Patient.
+A doctor can manage multiple patients, and a patient can have multiple doctors.

Conclusion

This documentation guides developers on understanding, maintaining, and extending the system. It includes clear explanations and Javadoc comments to make contributing to the project easier.

Deployment Documentation

Introduction

This documentation provides instructions for installing, configuring, and running the Pharmacy Management System. It outlines the system requirements, installation steps, and how to run the application.

System Requirements

Java 11+: Required for compiling and running the system.
Git: For version control (to clone the repository).
Command-line interface (CLI) or IDE (e.g., IntelliJ, Eclipse, VS Code).

Installation Instructions

1/ Clone the repository

```
    +git clone https://github.com/KhoaPham2002/Sprint2_PharmacyManagementSystem
    +cd PharmacyManagementSystem
```

2/ Install Java: Download Java 17+ from Oracle.
3/ Compile the project

```
Javac -d bin src/*.java src/Menu.java
```

4/ Run the application

```
Java -cp bin src.Menu
```

Running the Application

Run the Menu.java file and interact with the system. This application allows you to add, remove and edit patients, doctors, medications and then generate reports. Moreover, you can check if the medicines are expired, create a prescription, generate reports for prescriptions and restock the drugs for the pharmacy system.

Conclusion

This system allows pharmacies to manage patients, doctors, medications, and prescriptions efficiently. Follow these steps to set it up and run it locally.