2025

Developer Documentation

JAVA

CAMERON BOYER & KASSAUNDRA FEQUET

Project Overview

The Medication Tracking System is a comprehensive Java-based console application designed for pharmacy management. It provides functionalities for managing patients, doctors, medications, and prescriptions through an interactive menu-driven interface.

Key Features:

- Patient, Doctor, and Medication management
- Prescription processing and tracking
- Comprehensive reporting system
- Expiry date monitoring
- Inventory restocking capabilities

Source Code Directory Structure

```
Sem3-Java-Sprint/
├── PharmacySystem/
│   ├── EmptyMenu.java
│   ├── MedicationTracking.java
│   ├── Person.java
│   ├── Patient.java
│   ├── Doctor.java
│   ├── Medication.java
│   └── Prescription.java
├── Documents/
│   └── User Documentation
│   └── Developer Documentation
```

Class Explanations

- **Menu.java:** Contains the main method and user interface logic, handles all user interactions and menu navigation.

- **MedicationTracking.java:** The controller class that manages all entities and operations in the system.

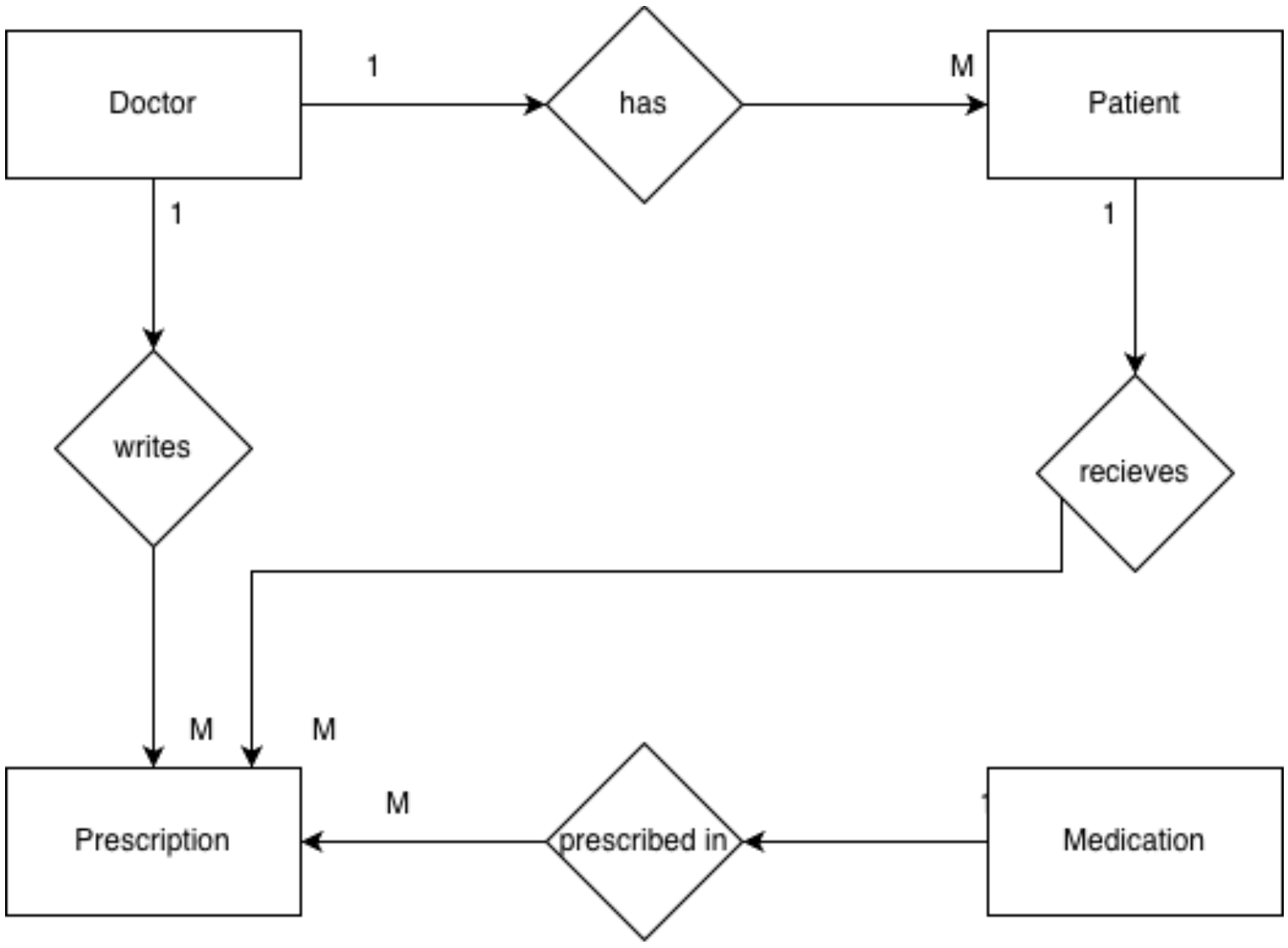
- **Person.java:** Abstract parent class defining common attributes for Patient and Doctor.

- **Doctor.java:** Represents medical professionals with specializations and patient assignments.
- **Patient.java:** Represents patient entities with medication history and prescription tracking.
- **Medication.java:** Manages pharmaceutical products with dosage, stock, and expiry information.
- **Prescriptions.java:** Links doctors, patients, and medications with prescription details.

Database Entities & Attributes

- **Doctor**
  - id (PK)
  - name
  - age
  - phoneNumber
  - specialization
- **Patient**
  - id (PK)
  - name
  - age
  - phoneNumber
- **Medication**
  - id (PK)
  - Name
  - Dosage
  - quantityInStock
  - expiryDate
- **Prescription**
  - id (PK)
  - doctor (FK)
  - patient (FK)
  - medication (FK)
  - dateIssued
  - prescriptionExpiry

Entity Relationships



- **Medication to Prescription**
  - **Type:** One to Many
  - Each medication can appear in many prescriptions

- **Patient to Prescription**
  - **Type:** One to Many
  - Each patient can have many prescriptions.

- **Doctor to Prescription**
  - **Type:** One to Many
  - Each doctor can write many prescriptions.

- **Doctor to Patient**
  - **Type:** One to Many
  - One doctor can have many patients, but a patient can only have one doctor.
  - If changes are made so patients can have multiple doctors, the relationship would become many to many.

GitHub

1. Open a JavaIDE(IntelliJ IDEA, VS Code or Eclipse)
2. Clone the repository

   ```
   git clone https://github.com/KassFequet/Sem3-Java-Sprint.git
   cd Sem3-Java-Sprint
   ```

3. Navigate to project directory

   ```
   cd Sem3-Java-Sprint
   cd PharmacySystem
   ```

4. Compile and run

```
Java Menu
```

5. The console menu will display available options for interacting with the system.

Javadocs

All javadocs have been compiled into an HTML file in the repository.

To access open docs/index.html in your browser.

Build Process

Compilation:

- Navigate to project directory
- **Run:** javac *.java (compiles all Java files)
- **Run:** java Menu (executes application)
- **IDE Alternative:** Use Run button in VS Code/IntelliJ/Eclipse

Compiler Time Dependencies

Java Standard Library Only:

- java.util.Scanner (user input)
- java.util.List/ArrayList (data storage)
- java.time.LocalDate (date handling)
- Requirements: JDK 8+

Development Standards

Naming:

- PascalCase for classes, camelCase for methods/variables.

Code Organization:

- Single responsibility per class
- Private fields with public getters/setters
- Static helper methods for menu operations

Documentation:

- Javadoc for public methods, inline comments for complex logic.

Error Handling:

- Null checks, user-friendly messages, input validation.

UI Standards:

- Numbered menus, consistent formatting, clear success/error messages.