Midterm Sprint – Java

March 2, 2025

Sarah Murphy, Zachary Collier, Scarlett/Kyle Budgell

Pharmacy Management System

User Documentation

The Pharmacy Management System is a program designed to hold the information of doctors, patients, medications and prescriptions using Object-Oriented Programming principles in Java. Our system provides functionality for tracking prescriptions, managing medication inventory, as well as assigning patients to the doctors in the system, while also ensuring that all necessary data is accessible and well-organized.

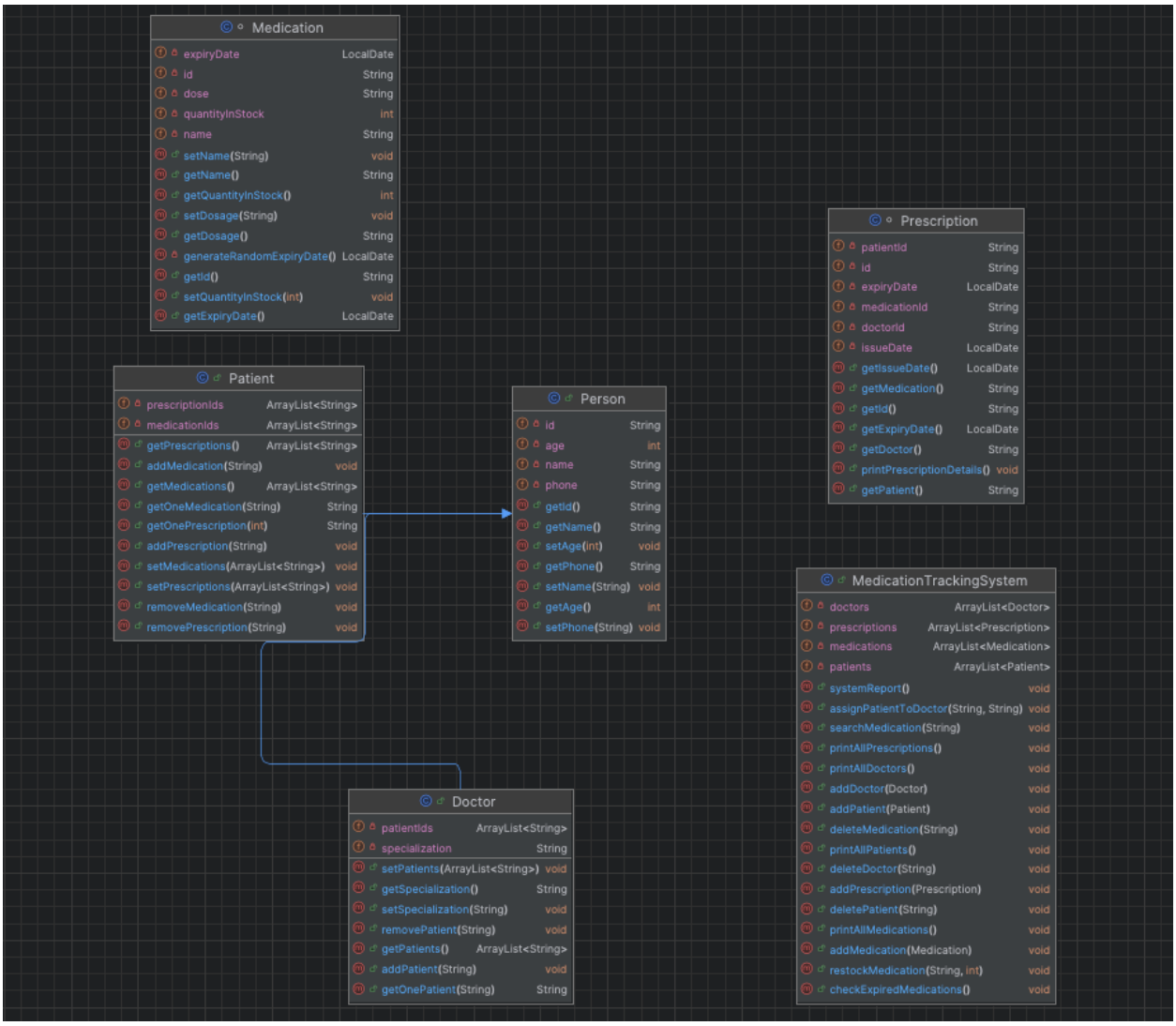
Classes & Their Responsibilities

This program contains 6 classes:

1. Person (Super Class)
   Attributes: Id, name, age, phone.
   Methods: Getters and setters for each attribute (ID, Name, Age, Phone).
   Usage: This class acts as a base class for the Patient and Doctor classes.
2. Patient (extension of Person class)
   Attributes: medicationId, prescriptionId
   Methods: Add, remove, get medications. Add, remove, get prescriptions.
3. Doctors (extension of Person class)
   Attributes: specialization, patientId
   Methods: add, remove, and get patients. Getter and setter for specialization.
4. Medication
   Attributes: Id, name, dose, quantityInStock, expiryDate.
   Methods: Getters and setters for Id, name, dose, quantityInStock, and expiryDate.
   Method to generate expiryDate.
5. Prescription
   Attributes: Id, doctorId, patientId, medicationId, issueDate, expiryDate.
   Methods: Method to link prescriptions to patients and doctors.
6. MedicationTrackingSystem

Attributes: doctors, patients, prescriptions, medications.
Methods: Add, delete, print doctors, patients, medications and prescriptions, manage assignments, generate reports.

Class Diagram



How to Start the Application

1. Clone repository from GitHub.
2. Open the project in your preferred IDE (VSCode, Intellij, etc).
3. Run the Menu class to start the program.
4. Follow the on-screen directions and in the terminal to interact with the application.

Development Documentation

Our source code directory structure is as follows: our repository is called java-midterm-sprint-w2025. Inside the repository is our main project, which contains a folder with all of the java files, as well as the README.md file. The main application is held in a folder called src. The src folder includes each class: Person.java, Patient.java, Doctor.java, Medication.java, Prescription.java, Menu.java and MedicationTrackingSystem.java.

Build Process

Navigate to the src folder > open terminal > run javac Menu.java > java Menu

Compiler Time Dependencies

No external dependencies are required to run this project.

Development Standards

We followed proper naming conventions for all of our classes, methods and variables. This project also contains the proper access modifiers (private/public) for each class. Our code contains appropriate Javadoc comments for each class and method.

Theoretical Database Design

- The tables in this theoretical database would be Patient, Doctor, Medications and Prescriptions. The database would not have a Person table, as this could add redundant data.
- Patients Table: PatientID (PK), Name, Age, Phone.
- Doctors Table: DoctorID (PK), Name, Specialization, Phone.
- Medications: MedicationID (PK), Name, Dose, Quantity, Expiry Date.
- Prescriptions: PrescriptionID (PK), PatientID (FK), DoctorID (FK), MedicationID (FK), IssueDate, Expiry Date.
- Relationships:
    1. Doctors to Patients: One-to-many, as one doctor can have multiple patients, but each patient only has one doctor.

2. Patients to Medications: Many-to-many, as one patient could have multiple medications, and each medication can be prescribed to multiple patients.
3. Patients to Prescriptions: One-to-many, as each prescription is meant for only one patient, but each patient can have more than one prescription.
4. Doctors to Prescriptions: One-to-many, as one doctor can write multiple prescriptions, but each persons' prescription is written by one doctor.
5. Medications to Prescriptions: One-to-many, as each prescription is for a specific medication, but each medication can be part of multiple prescriptions.

Getting Source Code from GitHub

Getting this source code from GitHub can be done in a few easy steps:

1. Clone the repository in your preferred IDE using the **git clone** command
   git clone https://github.com/ZCollier-dev/java-midterm-sprint-w2025

2. Navigate into the cloned repository to run the code using the **cd** command
   cd src

Deployment Documentation

1. Open your preferred IDE (VSCode, Intellij, etc).
2. Clone the repository from Github as described above.
3. Navigate to the src folder of the repository as described above.
4. Compile the project using the command: **javac Menu.java**.
5. Run the application using the command: **java Menu**.
6. Follow the menu options provided in the terminal to interact with the application.

Enjoy!