**Object Oriented Programming Report**

**Traffic Violation System**

**Department name: Computer Science department**

**Course number:**

**Course name: Database**

**Instructor’s name:**

**Due date :19 April 2025**

**Project Title: Care Mate**

|  |  |  |
| --- | --- | --- |
| **Leader** | **ID** | **Member Name** |
| **Ö      Ö** | 2230005538 | Nora Alowaidh |
|  | 2230003199 | Rimas Aldosari |
|  | 2230002325 | Aryam Hakami |
|  | 2230009046 | Ola Yasir Omer |
|  | 2230007334 | Ibtehaj Alqahtani |
|  | 2230007081 | Ibtesam Alqahtani |

**Table of Contents**

[INTRODUCTION: 3](#_Toc196084990)

[DESIGN: 4](#_Toc196084991)

[IMPLEMENTATION: 23](#_Toc196084994)

[TESTING: 26](#_Toc196084996)

[CONCLUSION: 32](#_Toc196084997)

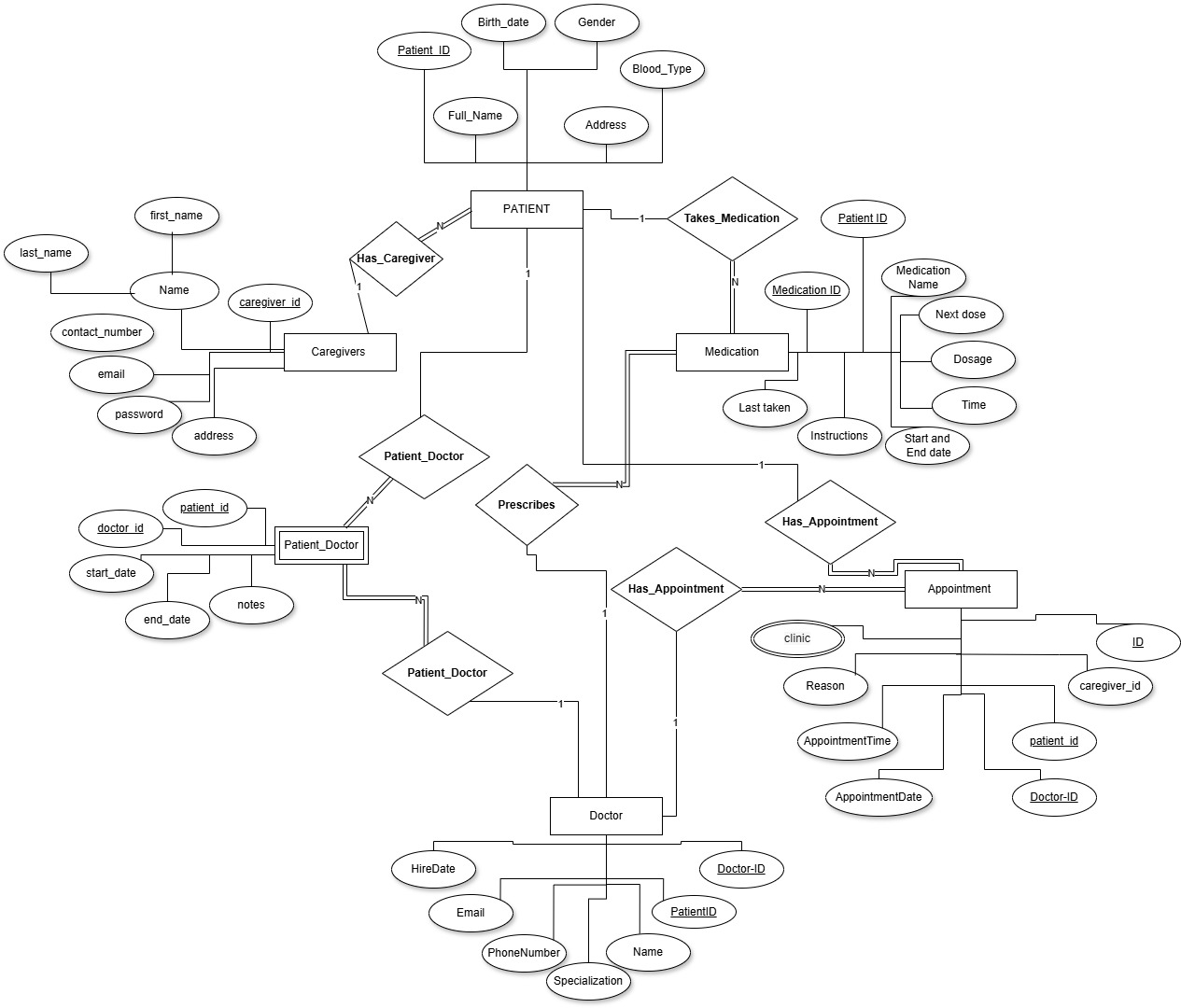
# INTRODUCTION:

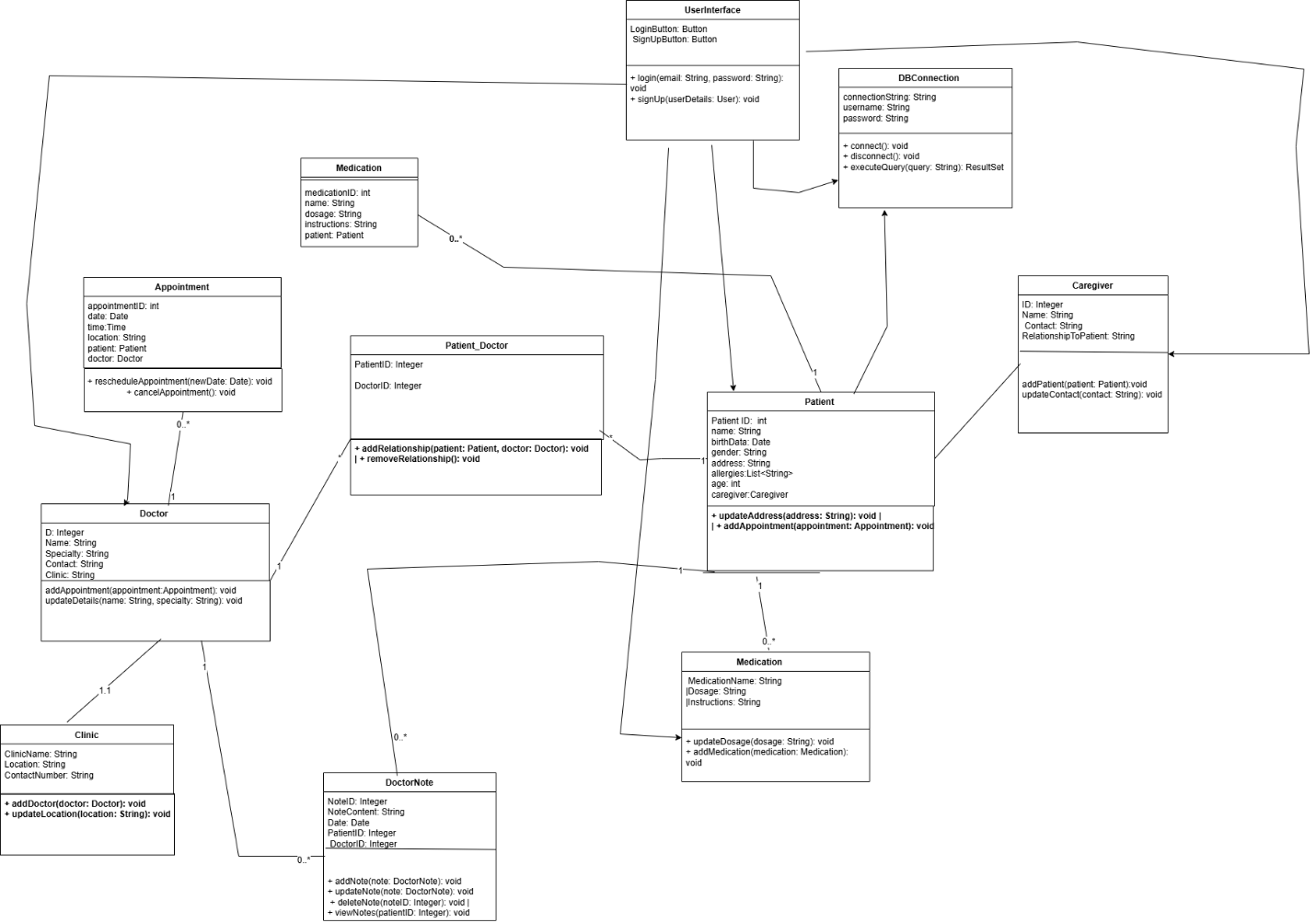
**Care Mate System** is a Java application that enables caregivers and physicians to follow and record the health condition of patients in an appropriate way. Caregivers are responsible for managing daily updates such as medication, appointments, and health notes while physicians can view patient records and document visits. It works towards accurate documentation and continual follow-up with the aim of enhancing the quality of care.  
  
The system is developed with Java and MySQL, following Object-Oriented Programming concepts such as inheritance, encapsulation, and polymorphism to provide modular, maintainable, and extensible design.  
  
**The Care Mate system aims at the following:**

1. To create a centralized object-oriented system to store patient information, daily care information, medicines, appointments, and doctor reports.
2. To apply OOP principles in system design to support clear class hierarchy and object interaction.
3. To facilitate communication between care givers and physicians with structured and timely exchange of information.
4. To supply user-friendly interfaces optimized for every role (care giver and physician), allowing secure and effective access to patient information.

# Design:

1.ERD (Entity-Relationship Diagram):





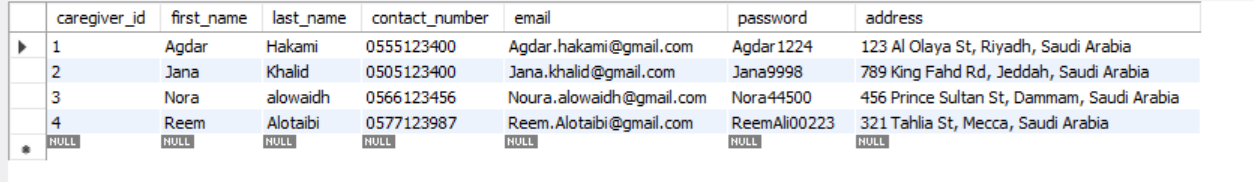
UML:

2.Database Schema:A diagram of a company

AI-generated content may be incorrect.

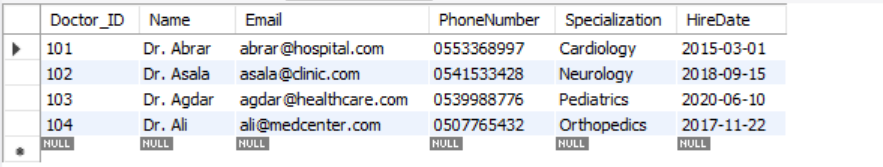
**3.Database Design (Tables):**

**Caregiver**



The **Caregiver** table stores information about people who assist patients in daily activities, such as family members or hired helpers. Each caregiver is linked to a patient and may provide emotional, physical, or medical support. This table includes personal contact details and the relationship to the patient.

**Doctor**



The **Doctor** table holds records of medical professionals registered in the system.  
Each doctor has a specialty and is associated with a specific clinic.  
Contact details and identification are stored here for appointment and reference purposes.

**Clinic**

A screenshot of a computer

AI-generated content may be incorrect., Picture

The **Clinic** table contains data about healthcare facilities within the system.  
It includes the clinic’s name, location, and contact number.  
Doctors are linked to clinics, allowing patients to know where to find them.

**Patient**

A screenshot of a computer

AI-generated content may be incorrect., Picture

The **Patient** table stores personal and health-related information about each patient.  
This includes name, birth date, gender, address, and caregiver reference.  
It also includes computed data like age and may store multiple allergies per patient.

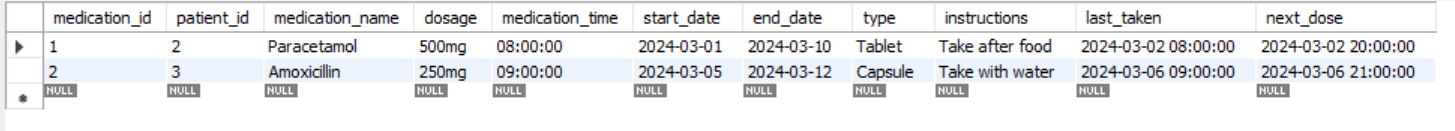
**Patient\_Doctor**

A screenshot of a computer

AI-generated content may be incorrect., Picture

The **Patient\_Doctor** table represents the many-to-many relationship between patients and doctors.  
Each record shows that a patient is treated by a certain doctor.  
It uses a composite key combining patient ID and doctor ID.

**Medication**



The **Medication** table keeps track of all medicines prescribed to patients.  
Each entry includes the name, dosage, and how the patient should take it.  
It helps doctors and caregivers manage treatments efficiently.

**Appointment**

A screenshot of a computer

AI-generated content may be incorrect., Picture

The **Appointment** table logs every visit scheduled between a patient and a doctor.  
It stores the date, time, and location of the appointment.  
It helps organize the clinic's schedule and provides historical visit data.

**View example**

A screenshot of a computer

AI-generated content may be incorrect., Picture

A **View** is a virtual table based on the result of an SQL query.  
It allows you to display specific data from one or more tables without modifying the original data.  
For example, a view can show only patient names and their assigned doctors for easy access.

**Update example**

A screenshot of a computer

AI-generated content may be incorrect., Picture

The **Update** statement in SQL is used to change existing data in a table.  
It allows the modification of one or more columns in selected rows based on a condition.  
For example, updating a patient’s address or changing the dosage of a medication.

**Delete example**

A screenshot of a computer

AI-generated content may be incorrect., Picture

The **Delete** statement is used to remove records from a table.  
You can delete specific rows by setting conditions in the WHERE clause.  
For example, deleting an old appointment or a caregiver no longer associated with any patient.

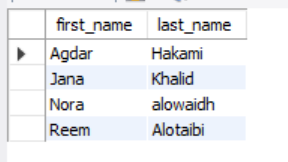
**Insert example**

A screenshot of a computer

AI-generated content may be incorrect., Picture

The **Insert** statement is used to add new records into a table.  
Each column must be provided with a value that matches its data type.  
For example, inserting a new patient into the system with all required details.

**Composite attribute**



A **Composite Attribute** is made up of multiple sub-parts, each representing a more basic attribute.  
For example, a full name could be divided into first name and last name.  
It helps in organizing data more clearly and efficiently.

**Multi-value column (Allergies))**

A screenshot of a computer

AI-generated content may be incorrect., Picture

A **Multi-value Column** stores multiple values in a single field.  
In this case, a patient may have multiple allergies, like "Peanuts, Dust, Penicillin".  
It’s usually better handled using a separate table to maintain normalization.

**Computed column (Age)**

A screenshot of a computer

AI-generated content may be incorrect., Picture

A **Computed Column** calculates its value based on other columns.  
For example, the age of a patient can be calculated automatically from their birthdate.  
This ensures up-to-date information without manual changes.

**A screenshot of a login page

AI-generated content may be incorrect.**4.GUI Design:

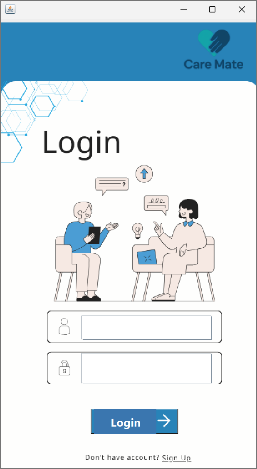
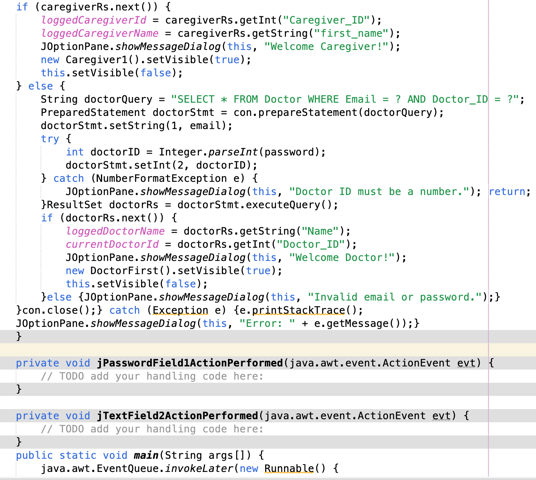
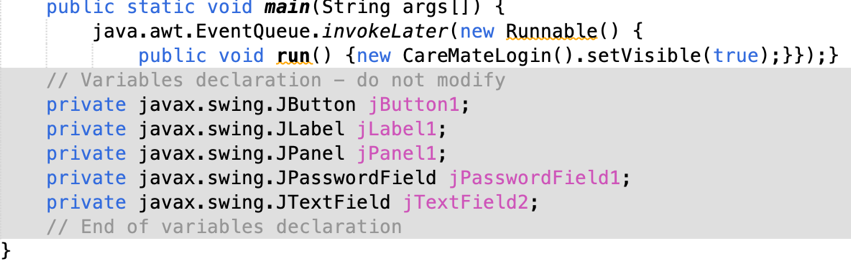
****

**A screenshot of a computer program

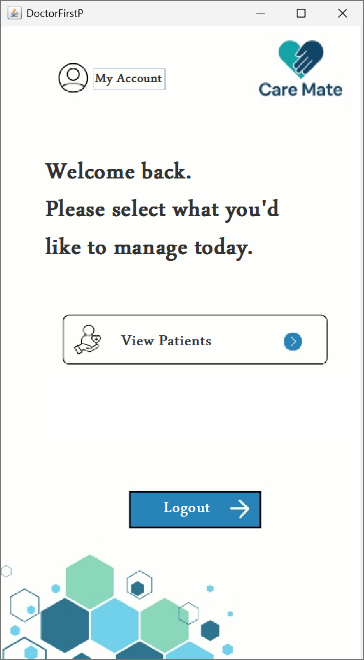
AI-generated content may be incorrect.A screenshot of a computer program

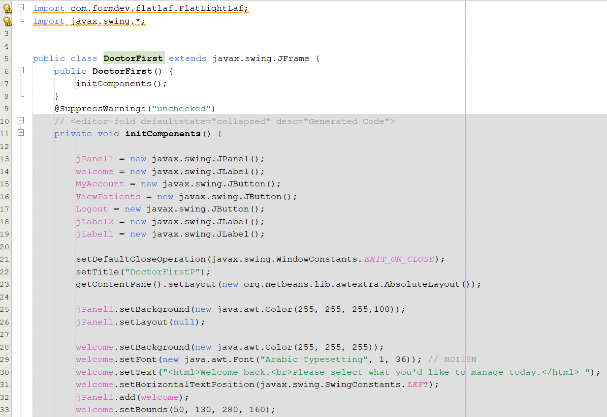
AI-generated content may be incorrect.Welcome Interface: CIDE LINK:** [**WelcometoCareMate.txt**](https://udksa-my.sharepoint.com/:t:/g/personal/2230005538_iau_edu_sa/EWjDSZtSp31CoHBmPIte7mMBnLHyEEd8JvZCigpyx9Mniw?e=euCwBM)  
Displays a welcome screen with two buttons: "Login" to access an existing account, and "Sign Up" to create a new account. Each button opens the corresponding window and closes the welcome screen.

**Sign Up Interface:** **CIDE LINK:** [**CareMateSignUp.txt**](https://udksa-my.sharepoint.com/:t:/g/personal/2230005538_iau_edu_sa/EQHdGtRfEypNm3GcqvrNvC0B5xFjgq7RF1G_6uFj0sp3IA?e=4SAV3f)  
A user registration form with fields for first name, last name, email, password, and date of birth. It uses Java Swing components and includes UI features like placeholders, focus effects, and a calendar picker. The interface is designed to collect user information for account creation.

****

**Login Interface:** **CIDE LINK:** [**CareMateLogin.txt**](https://udksa-my.sharepoint.com/:t:/g/personal/2230005538_iau_edu_sa/EQR2gN_4GH9Bpi4-fsJHmuwBwACrVVvDjeOloLdSEA8qGA?e=T2XLF7)  
Allows users to log in using email and password. It checks the database to identify if the user is a caregiver or doctor and redirects them to the correct page.

**A screenshot of a computer program

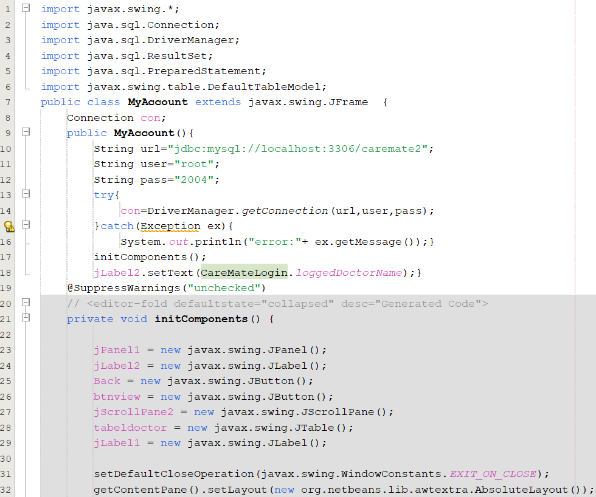
AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Doctor Home Interface:** **CIDE LINK:** [**DoctorFirst.txt**](https://udksa-my.sharepoint.com/:t:/g/personal/2230005538_iau_edu_sa/EZhxVLBKEkZOruoF8ep_TUsBC9dLNOMJ2V8Vi2JUYsGWNg?e=bd4Zzn)  
This screen appears after a doctor logs in. It provides quick access to the "My Account" page and the "View Patients" section. It also includes a logout button.

**A screenshot of a medical form

AI-generated content may be incorrect.**

**MyAccount Interface:** **CIDE LINK:** [**MyAccount.txt**](https://udksa-my.sharepoint.com/:t:/g/personal/2230005538_iau_edu_sa/EbiM4aBSZZhFj66rpJrxrrgBnz4A6Zx8tbe6JX53hNQQDA?e=oo2xoy)  
Shows the logged-in doctor's name and a table of all doctors with key details. Includes "Back" and "View" buttons to navigate and refresh data.

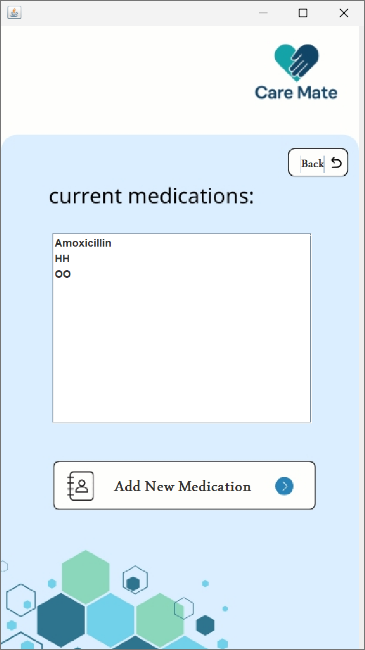
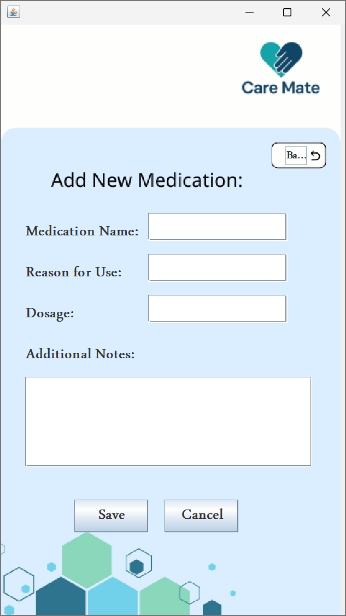
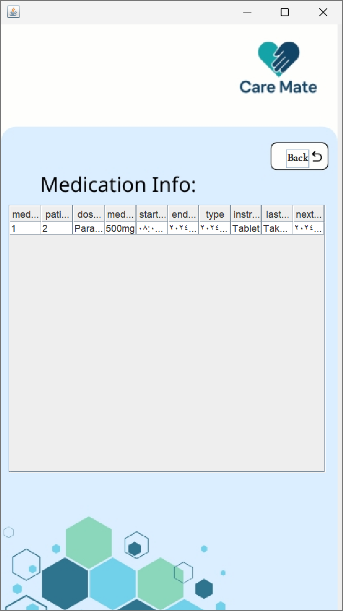
****

**Patients Interfaces:**  
These interfaces allow doctors to view patient data.

* **Patient Info** shows detailed information (name, gender, date of birth, etc.).
* **Patients List** displays a simple list of patient names.  
  Both screens include a **Back** button to return and a **View** button to load or refresh the data.

**Patient Info** CODE LINK: [PatientDetailsCareMate.txt](https://udksa-my.sharepoint.com/:t:/g/personal/2230005538_iau_edu_sa/EfAegaF3-DdDu3hT_iDgh-kBdzYujm9sGvgA-gJTg_FG_g?e=eaZpgG)

**Patients List** CODE LINK: [PatientsList.txt](https://udksa-my.sharepoint.com/:t:/g/personal/2230005538_iau_edu_sa/EeG_PMvw6nRFm85DTqtoJ00BEAtsvk2gI0M5AWO98qv6Ig?e=9ivy1e)



Medications Interface:

These screens manage patient medications.

Current Medications shows a list of a patient’s active medications.

CODE LINK: [ViewMedications.txt](https://udksa-my.sharepoint.com/:t:/g/personal/2230005538_iau_edu_sa/EZz4_IH5wRpLpSVtJS8MyHYBO7HpmgrWdId9YnxNvmiYyg?e=Z2arVx)

Add New Medication allows doctors to input new prescriptions with name, reason, dosage, and notes.

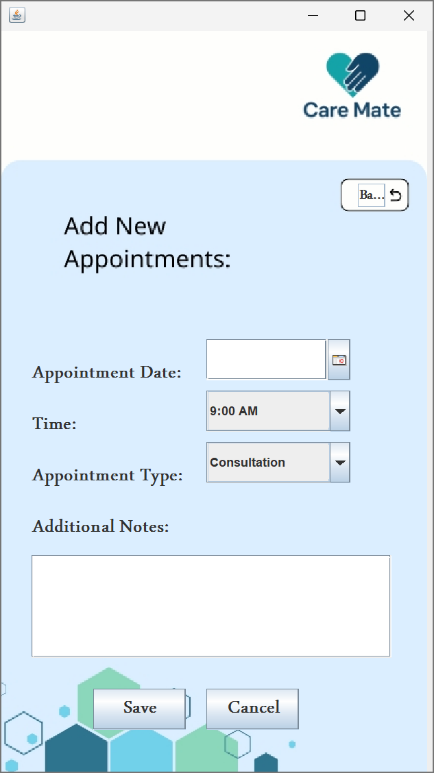
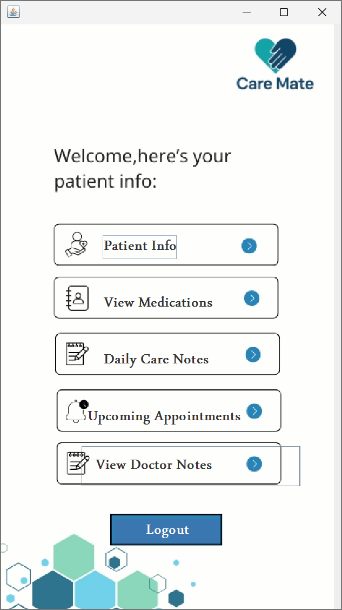
CODE LINK: [AddNewMedication.txt](https://udksa-my.sharepoint.com/:t:/g/personal/2230005538_iau_edu_sa/EUK16Hkw7O1ImM7no1v3DvQB0bZm92aXWoEg3WhB-238CA?e=YV7XSP)

Medication Info displays detailed medication data in table format.

CODE LINK: [MedicationInfo.txt](https://udksa-my.sharepoint.com/:t:/g/personal/2230005538_iau_edu_sa/EY4nUiVjUupLjfVrZ8sot_oBIU_lid05258tRx6y6vNtng?e=ImyTDh)

Each screen includes navigation buttons

A screenshot of a phone

AI-generated content may be incorrect.

### **Dashboard Interface: CODE LINK:** [Caregiver1.txt](https://udksa-my.sharepoint.com/:t:/g/personal/2230005538_iau_edu_sa/EVDfG5oMXNtEqsDJx1G0iWYBoV-UIm_Qw5A0wtL5yan3hA?e=drprqV)

This screen acts as the main navigation panel for accessing all patient-related information.

It includes the following options:

* **Patient Info** – View the patient's personal details.
* **View Medications** – Access and manage current prescriptions.
* **Daily Care Notes** – Check daily care logs or nursing notes.
* **Upcoming Appointments** – Review scheduled future appointments.
* **View Doctor Notes** – See notes written by the doctor.

The interface also includes a **Logout** button for secure exit from the system.

**Appointments Interface:**

**Add New Appointment** allows doctors to schedule appointments by selecting the date, time, type (e.g., Consultation), and writing any additional notes. CODE LINK: [AddNewAppointments.txt](https://udksa-my.sharepoint.com/:t:/g/personal/2230005538_iau_edu_sa/EUn4RTdKXtNIkTt8XlNZzIoBAUMPdvf9NRDtg6BHEPZ35Q?e=dZXtRZ)

**Upcoming Appointments** displays a list of all future appointments with their date, time, and purpose. CODE LINK: [UpcomingAppointments.txt](https://udksa-my.sharepoint.com/:t:/g/personal/2230005538_iau_edu_sa/ER-7lQ1NXa1Jm54hJtYBf1oBHmYXsZvsgL97mdOicNSHYA?e=wPGmRs)

**Patient Dashboard Interface:**  
This is the main menu for patients. It allows navigation to patient info, medications, care notes, appointments, and doctor notes. CODE LINK: [PatientsInfo.txt](https://udksa-my.sharepoint.com/:t:/g/personal/2230005538_iau_edu_sa/EZYXw2r91elLowBhfIpGyVkBq128ja9InqTT26KUIMgXAA?e=Jz4NV6)

**Patient Info Interface:**  
Displays patient details like name, birthdate, gender, blood type, and address in a table. Users can load data using the "View" button. CODE LINK: [CPatientInfo.txt](https://udksa-my.sharepoint.com/:t:/g/personal/2230005538_iau_edu_sa/ES7TC2qpzg1NklvWgDTScxwB0DByOddwhOSGND8wi1tWxw?e=d3nBKZ)

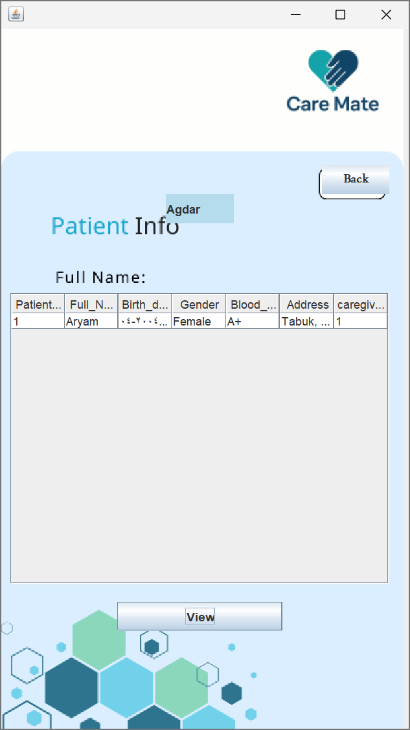
**View Medications Interface:**  
Shows a list of current medications for the patient. Simple layout with scrollable view and a back button. CODE LINK: [CFViewMedications.txt](https://udksa-my.sharepoint.com/:t:/g/personal/2230005538_iau_edu_sa/ERfOreBzd2hHqlwp4jsR22cBh4c87JV8slkuVnIWK4hJRQ?e=q8NOSg)

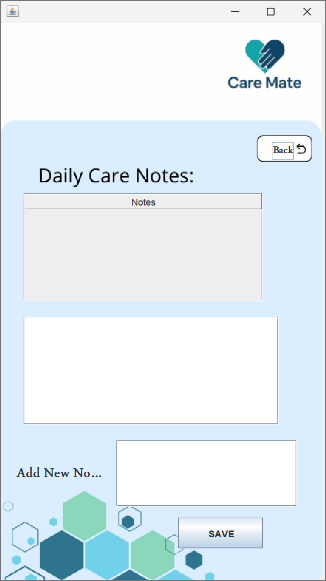
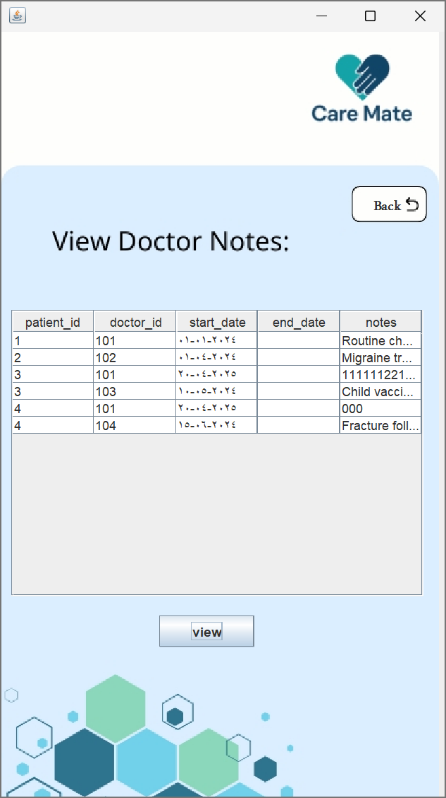
**Medication Info Interface:**  
Presents detailed medication data in a table, including dosage, dates, instructions, and schedule. Helps track prescriptions accurately. CODE LINK: [CMedicationInfo.txt](https://udksa-my.sharepoint.com/:t:/g/personal/2230005538_iau_edu_sa/EY7AvJsove9GhrUOTM4zCMsBDAzOFh-ItlVzJOErchQdQg?e=ktWi7G)

A screenshot of a medical application

AI-generated content may be incorrect.A screenshot of a medical application

AI-generated content may be incorrect.**A screenshot of a medical application

AI-generated content may be incorrect.**

A screenshot of a medical form

AI-generated content may be incorrect.

**Doctor Notes Interface: CODE LINK:** [**AddDoctorNotes.txt**](https://udksa-my.sharepoint.com/:t:/g/personal/2230005538_iau_edu_sa/EVeQnyaYHxVCrbYAZZUvHZABVontNu22braXzFNJi_dQsA?e=3slRat)  
This screen allows doctors to add notes about a patient’s condition or treatment. It includes a large text area to display existing notes and a separate field for adding new ones, along with a save button.

**Daily Care Notes Interface: CODE LINK:** [**DailyCareNotes.txt**](https://udksa-my.sharepoint.com/:t:/g/personal/2230005538_iau_edu_sa/Ed-GWKVjohlGkKn2BedR3TYB5vddEML_v7SP5NQYrGe8_w?e=plhNLv)  
Used to record daily observations or care details for patients. It features a table view to show past notes and a field to input new care notes, with a save button for submission.

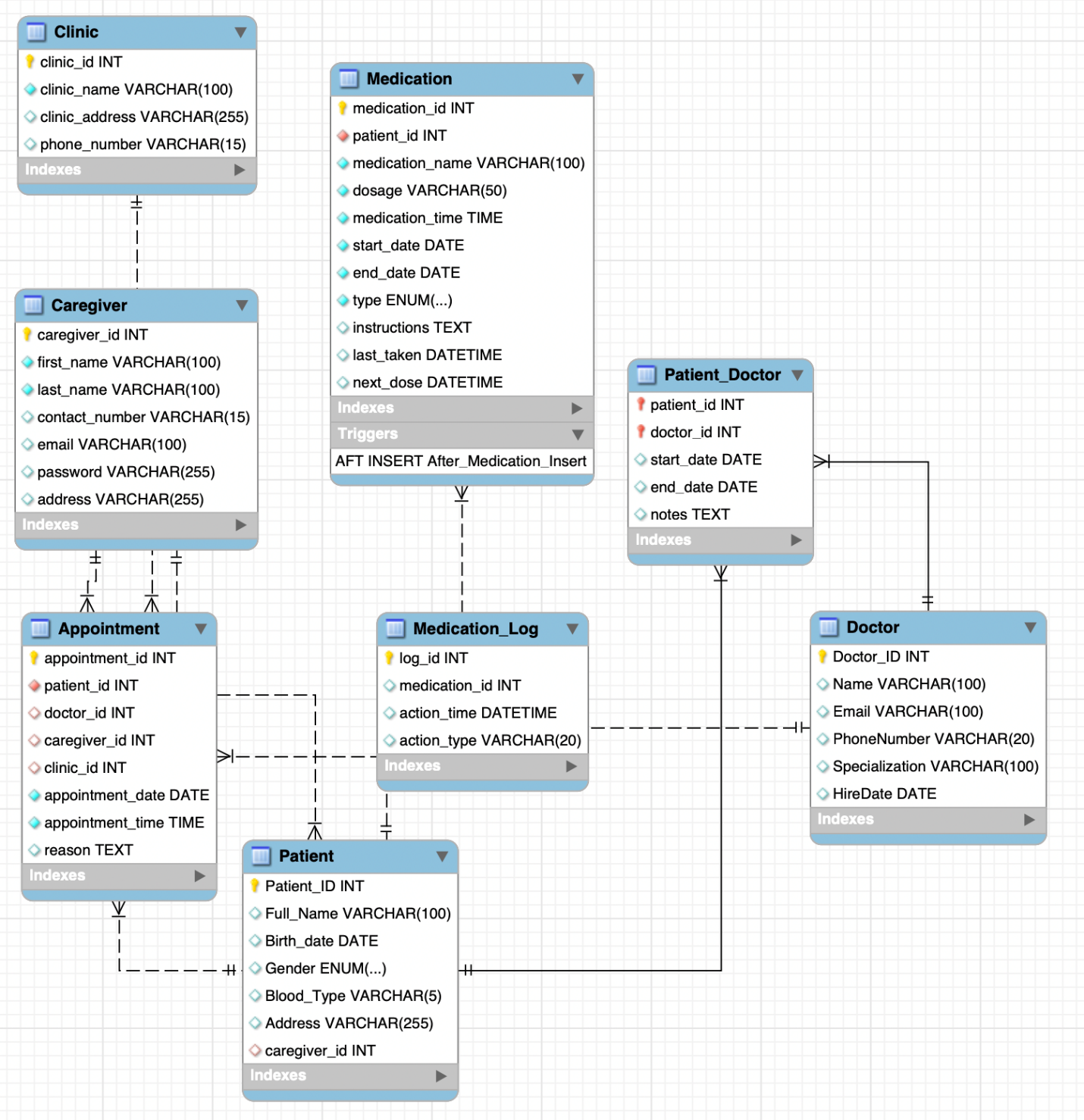
**View Doctor Notes Interface: CODE LINK:** [**ViewDoctorNotes.txt**](https://udksa-my.sharepoint.com/:t:/g/personal/2230005538_iau_edu_sa/EQF3CVLNIK1Ao-VJClOSGfABdP_iDfTb7tjUJ0kHYexVGA?e=ayzTJT)  
Displays all doctor notes related to patients in a table format, including doctor ID, patient ID, start and end dates, and note content. A "View" button refreshes or loads the data.

A screenshot of a phone

AI-generated content may be incorrect.

**Upcoming Appointments Interface: CODE LINK:** [**ViewAppointments.txt**](https://udksa-my.sharepoint.com/:t:/g/personal/2230005538_iau_edu_sa/EWmKDhgtu4BFjzCptHMg-nQBiPnsQBmWGatWhQKC7sVRng?e=fzVWaf)  
This screen displays a clear list of all upcoming appointments scheduled for the patient. It helps patients stay informed about their future visits, such as routine checkups, vaccinations, and follow-ups. The appointments are shown in a simple text area for easy readability. Users can scroll through the list to see what's next in their care plan. A "Back" button at the top allows smooth navigation to the previous menu. This interface ensures that patients never miss important appointments and are always aware of their medical schedule.

5.Class Hierarchy:

****

# Implementation:

#### **Technologies Used**

#### The system was developed using **Java** as the primary programming language, with **Java Swing** for building the graphical user interface. **MySQL** was used as the database to store patient, appointment, medication, and note data. The connection between the application and the database was handled using **JDBC (Java Database Connectivity)**.

#### **Database Connection**

#### The connection to the MySQL database was implemented using the following line:



This allows the system to perform real-time database operations such as retrieving and storing patient records, appointments, and doctor notes.

### **Key Functionalities**

* **User Login & Access**: Users can log in or sign up and are redirected to role-specific dashboards (doctor or caregiver).
* **Appointment Management**: Users can view and schedule appointments by selecting date, time, type, and adding notes.
* **Medication Tracking**: Caregivers can view and add medications with details like name, dosage, reason, and notes.
* **Doctor Notes**: Doctors can write and save notes for each patient, stored securely for future access.
* **Patient Information**: Patient profiles can be viewed and updated, including name, birthdate, gender, blood type, and address.
* **Daily Care Logging**: Caregivers can document daily observations and care activities for patients.
* **Role-Based Features**: The interface adapts to the user’s role, showing only the tools relevant to their responsibilities.

#### **User Interface Design**

The GUI was built using **Java Swing** with JFrame and JPanel components. Common UI elements include buttons, combo boxes, text fields, and scrollable lists. The layout uses absolute positioning, and background images were added to improve visual presentation.

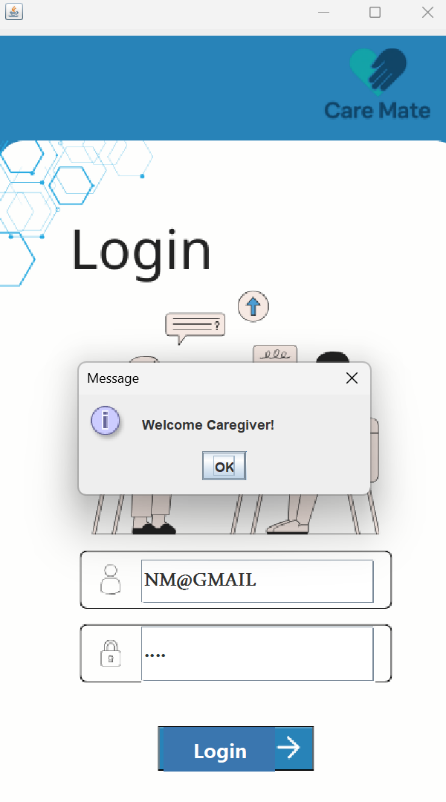
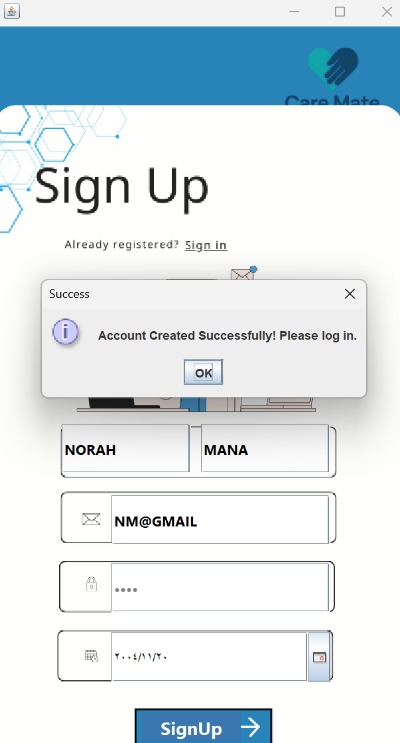
#### **Structure And Error Handling**

Each main feature is implemented in a dedicated screen (window), keeping the application modular and organized. Forms include input validation to ensure required fields are not left empty. Navigation between screens is handled smoothly, and error handling using try-catch blocks ensures stability in case of unexpected input or database errors.

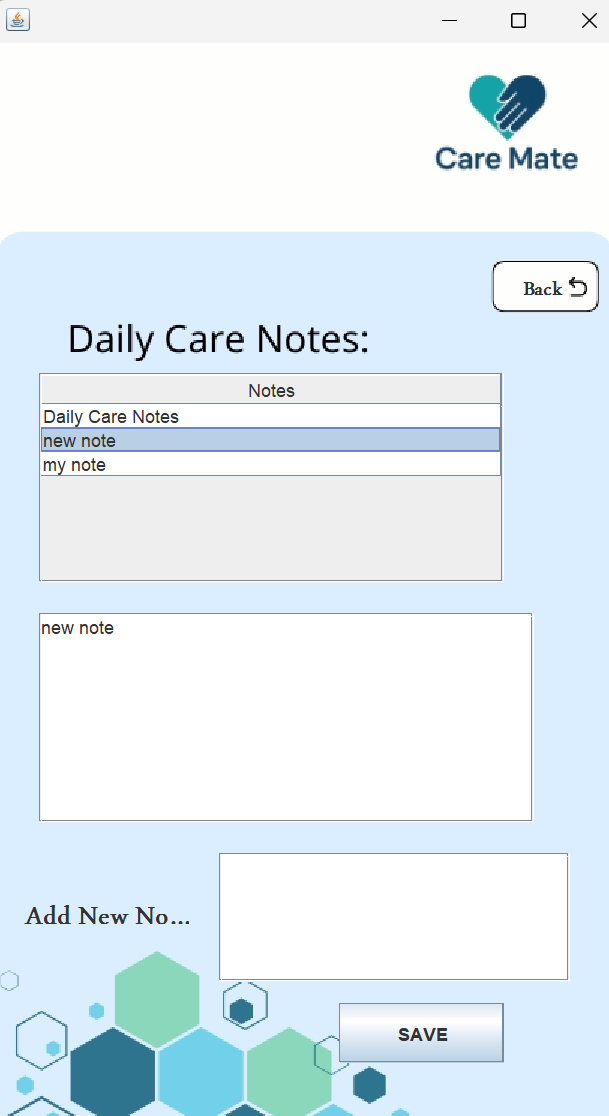
**System Behavior & Interaction:**

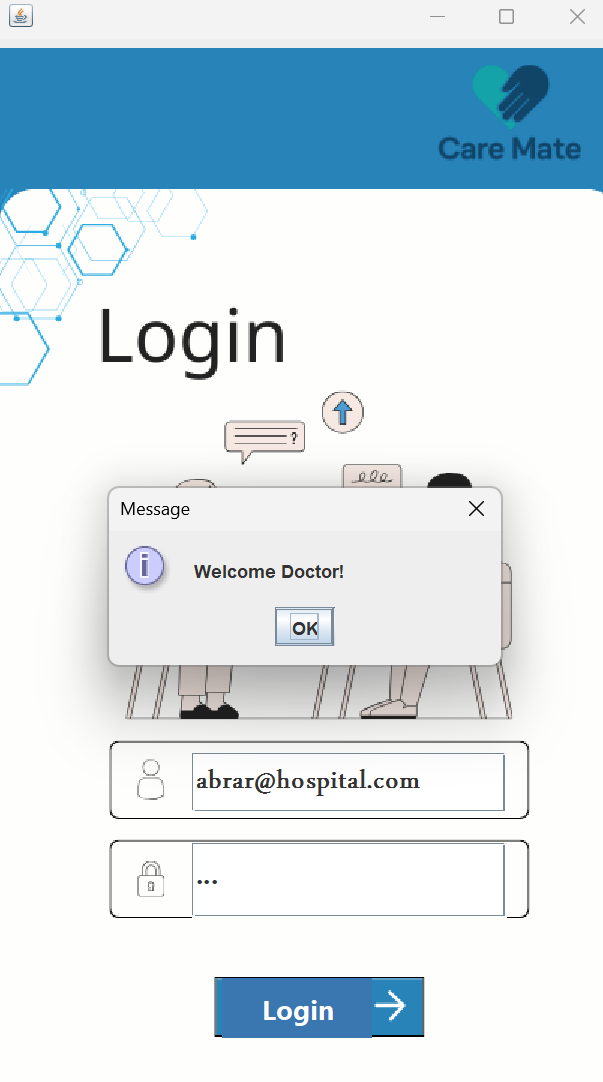
Each functionality is accessed through a dedicated user interface screen designed for clarity and ease of use. Basic error handling is implemented using try-catch blocks to prevent issues during data entry and database operations. Navigation between modules is managed through button actions, where clicking a button opens the target screen and closes the current one. This ensures smooth transitions and a user-friendly experience across the application.

**Testing:**

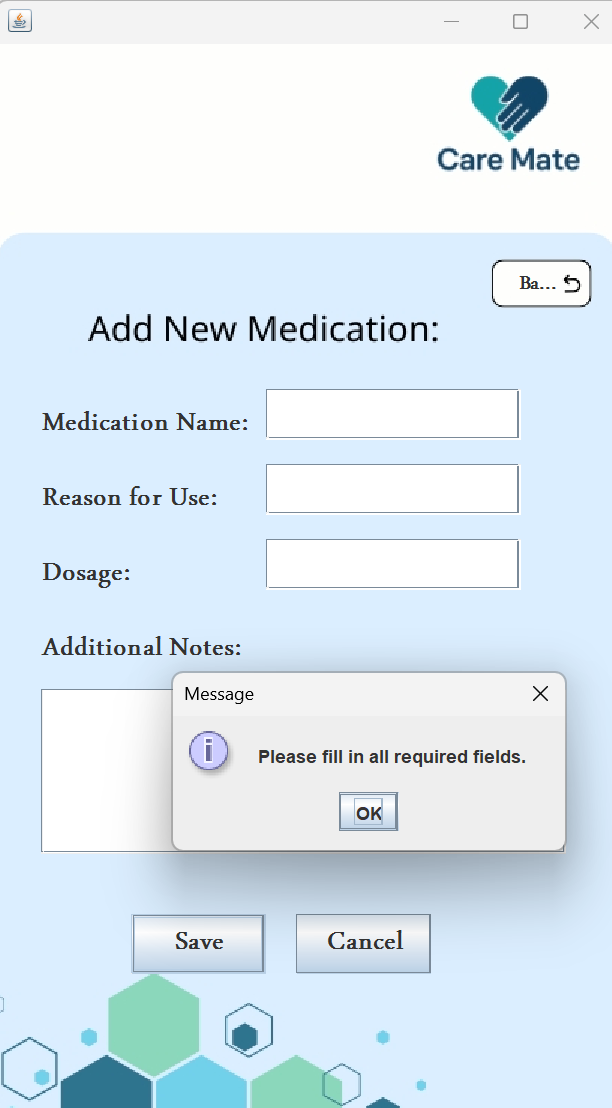
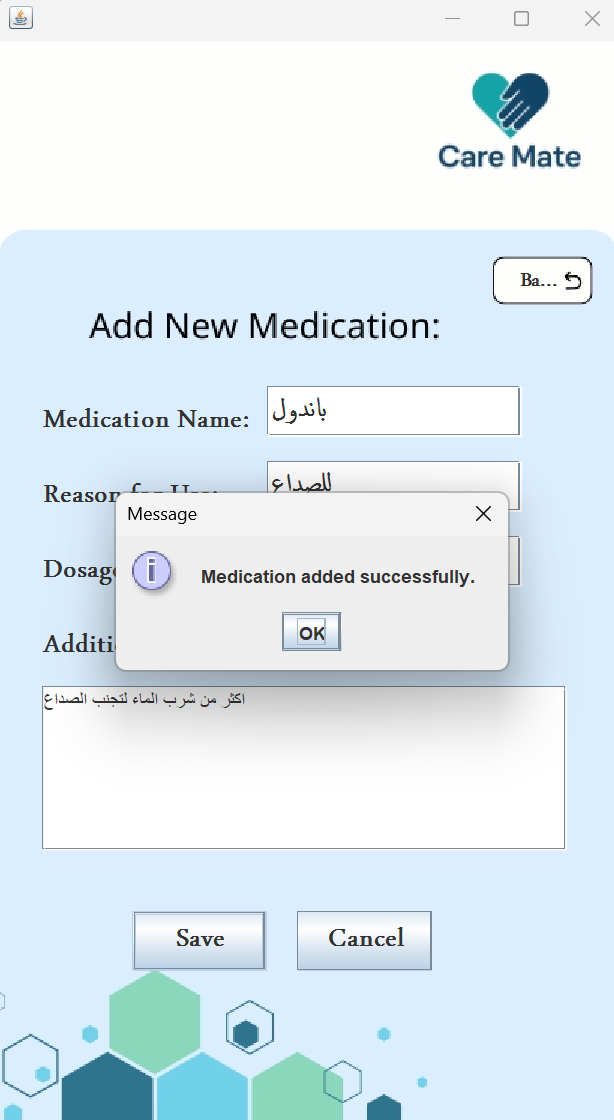


The testing interface ensures users receive clear feedback during sign-up and login. When valid information is entered, a success message confirms account creation or login. If incorrect credentials are entered, an error message appears saying “Invalid email or password.” This helps users understand whether their input is correct or needs to be fixed.

This interface allows caregivers to manage daily care notes for patients. It displays a list of saved notes in a table and provides two text areas one for viewing a selected note and another for writing a new one. After typing, users can click **SAVE** to store the note in the system. The design ensures easy navigation and note tracking, with a **Back** button to return to the previous screen.

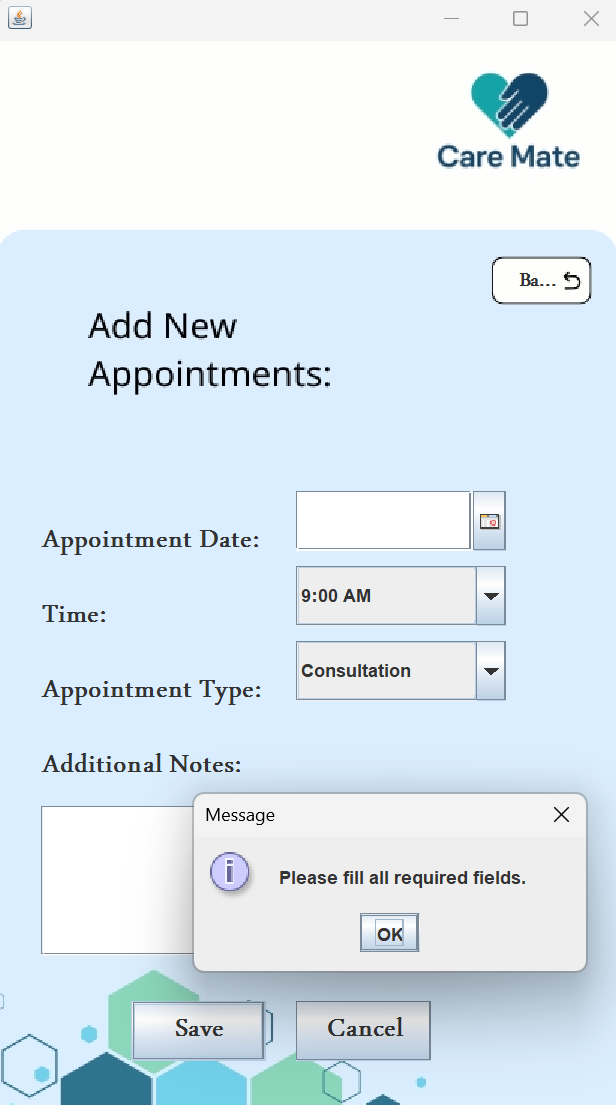
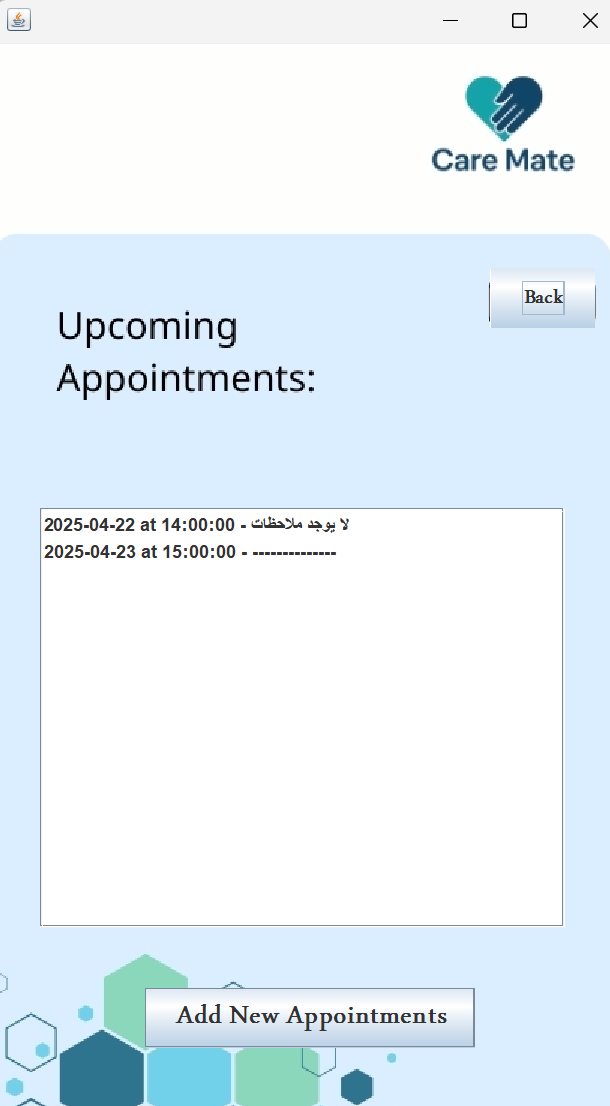
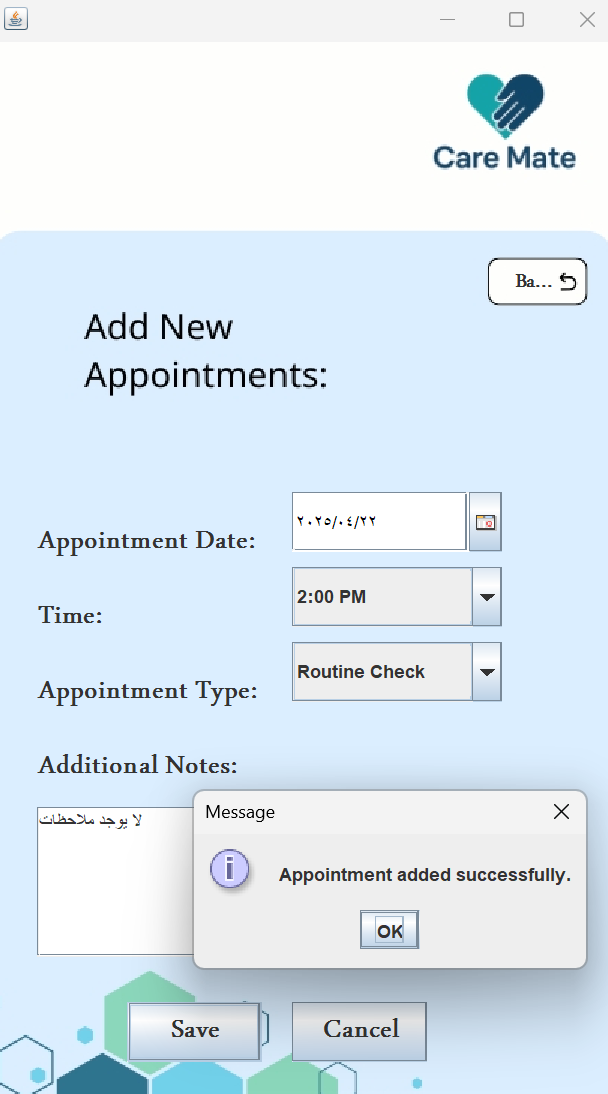


This section of the system shows the **Login Interface**, where user credentials are validated. If an incorrect email or password is entered, an error message appears: **"Invalid email or password."** However, if the credentials are correct, the system recognizes the user and displays a personalized welcome message based on their role, such as **"Welcome Doctor!"**. This enhances security and ensures a role-specific user experience.

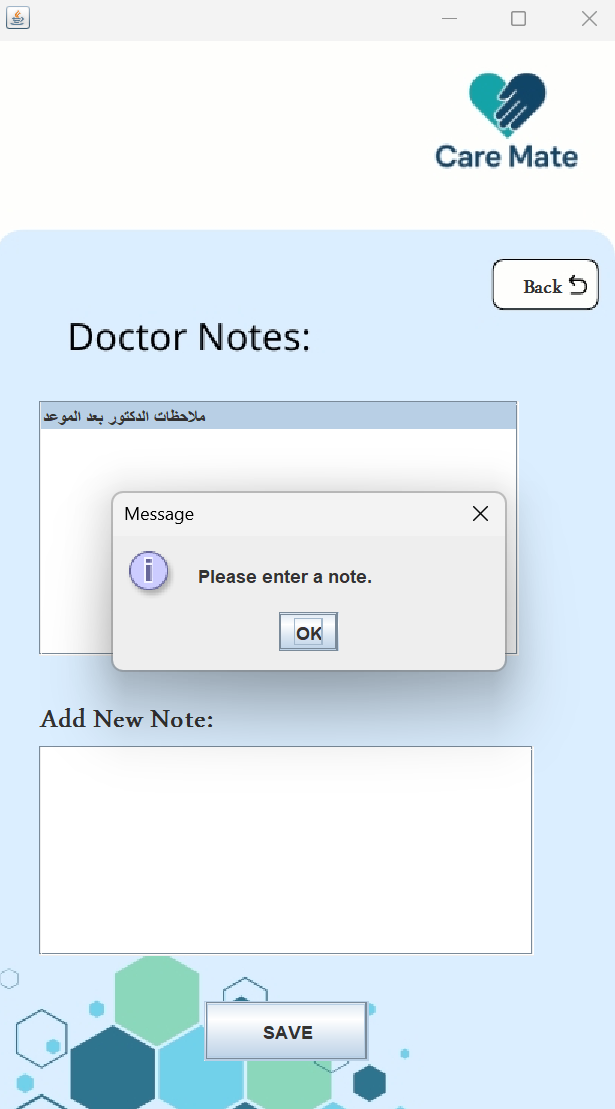
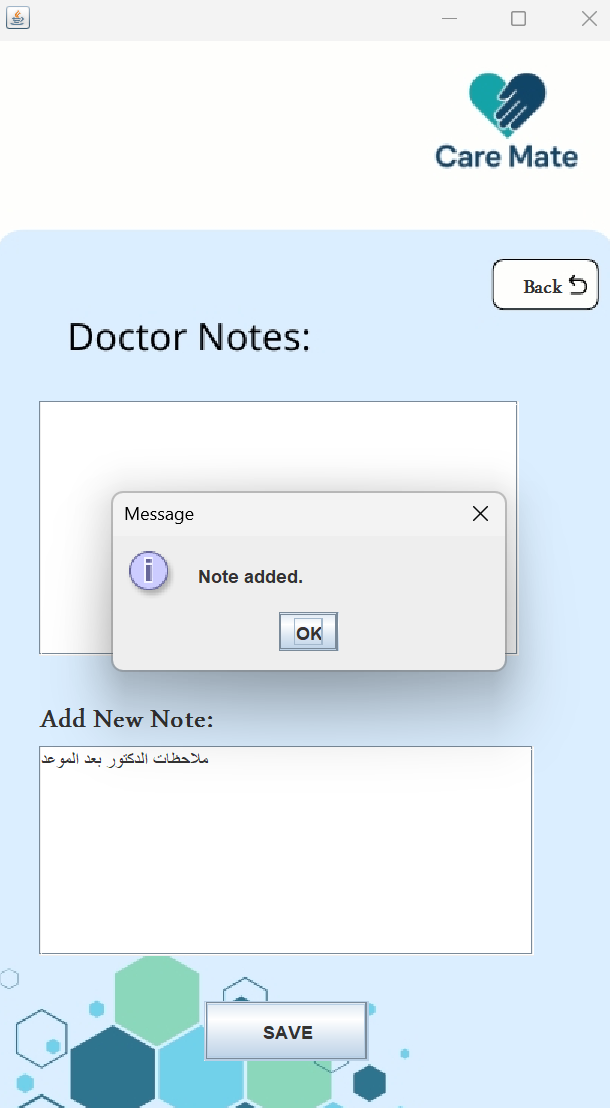


This section of the system shows the **Login Interface**, where user credentials are validated. If an incorrect email or password is entered, an error message appears: **"Invalid email or password."** However, if the credentials are correct, the system recognizes the user and displays a personalized welcome message based on their role, such as **"Welcome Doctor!"**. This enhances security and ensures a role-specific user experience.

This part of the system demonstrates the **Appointments Management** feature. When users try to save without completing the form, a validation alert appears: **"Please fill all required fields."** Once all data is properly entered, a confirmation popup states: **"Appointment added successfully."** After saving, the appointment appears in the **Upcoming Appointments** list, showing the date, time, and any additional notes. This ensures users can manage their schedules efficiently and receive immediate feedback on their actions.



This section highlights the **Doctor Notes** feature. If a user attempts to save an empty note, a message appears stating: **"Please enter a note."** Once a valid note is written, the system confirms with: **"Note added."** This functionality ensures that doctors can document important observations or instructions while preventing empty entries from being saved, maintaining data accuracy and completeness.



# Conclusion:

**The Care Mate System** is a significant step towards digitalizing and structuring healthcare management. It offers a combined solution for handling patients' information, caregivers, doctors, appointments, and medication prescribed. The system was developed using fundamental Object-Oriented Programming principles and integrated with a relational database for data integrity and scalability.  
  
One of the most robust features of the system is that it has separate interfaces for doctors, caregivers, and patients. These interfaces allow each category of user to see relevant information and perform desired operations in an appropriate and secure way, resulting in an improved overall user experience.  
  
This project makes a great foundation for the development of more complex health care applications in the future with much room to spin off into other features such as notification, intelligent reporting, and integration into general hospital-scale systems.

For full access to the source code implemented in the Care Mate System, please refer to the following link: <https://udksa-my.sharepoint.com/:f:/g/personal/2230005538_iau_edu_sa/EmNECbNuYxREhf2lWu-MdKgBZEGVlOhfgg_i5QO02q5u8g?e=byxa9C>