Hospital Management System Design Documentation

by

Gormery K. Wanjiru, Aziz Azizi

Version: 1.0

Grimstad, NOV 2024

Contents

1 Use Case Diagram

1.1 Overview

Use case diagrams help to visualize the requirements and functionalities of the system. The diagram shows the interaction between the actors and the various use cases they are involved with.

1.2 Use Case Diagram

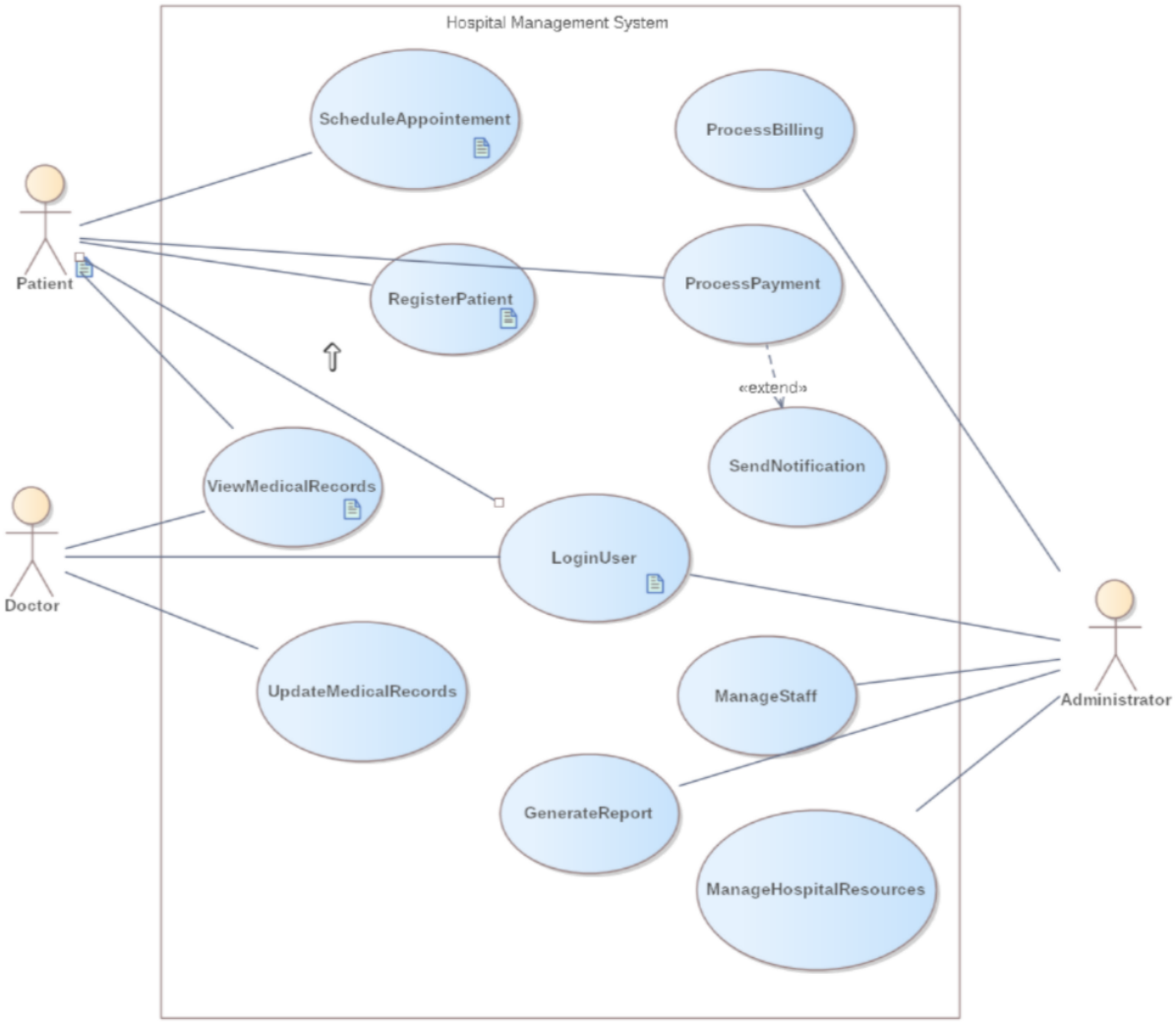

Figure 1: Use Case Diagram for Hospital Management System

1.3 Description of Actors and Use Cases

1.3.1 Actors

- **Patient**:
  - Register Patient
  - Login Patient
  - Schedule Appointment

  - View Medical Records
  - Make Payment
- **Doctor**:
  - Login Doctor
  - View Patient Records
  - Update Patient Records
  - Manage Appointments
- **Administrator**:
  - Login Administrator
  - Manage Staff
  - Manage Hospital Resources
  - Generate Reports
  - Process Billing
  - Send Notifications

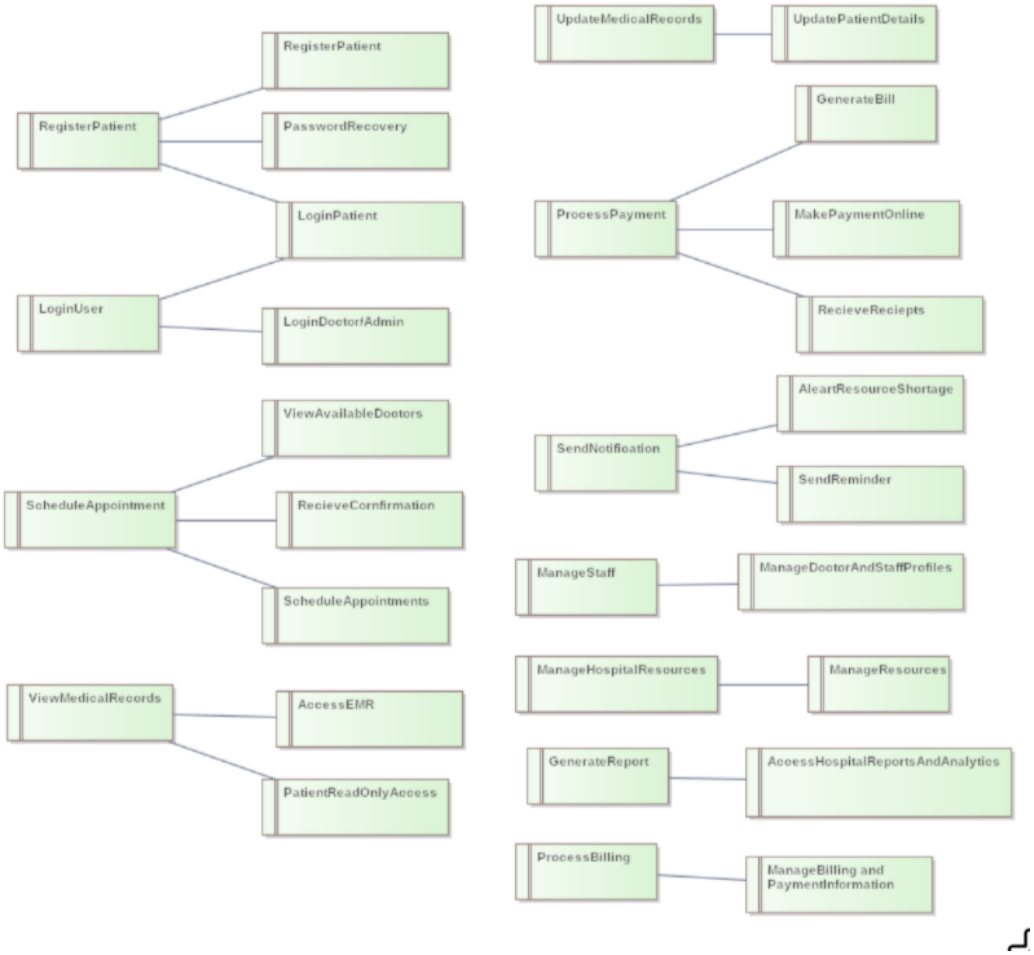

Figure 2: showing how all functional requirements are covered by the usecase

2 Use Case/Requirements Matrix

2.1 Overview

The use case/requirements matrix maps each use case to the functional requirements in the system requirement specification (SRS). This mapping shows how each functional requirement is fulfilled by one or more use cases.

2.2 Use Case/Requirements Matrix

Use Case	**Functional Requirements**
Register Patient	FR1 (Register patient), FR2 (Login patient), FR3 (Password recovery)
Login User	FR2 (Login patient), FR4 (Login doctor/admin)
Schedule Appointment	FR4 (View available doctors), FR5 (Schedule appointments), FR6 (Receive confirmation)
View Medical Records	FR7 (Access EMR), FR9 (Patient read-only access)
Update Medical Records	FR8 (Update patient details)
Process Payment	FR10 (Generate bill), FR11 (Make payment online), FR12 (Receive receipt)
Send Notification	FR16 (Send reminders), FR17 (Alert resource shortage)
Manage Staff	FR13 (Manage doctor and staff profiles)
Manage Hospital Resources	FR14 (Manage resources)
Generate Report	FR14 (Access hospital reports and analytics)
Process Billing	FR15 (Manage billing and payment information)

3 UI Mock-Ups for Each Use Case

3.1 Overview

simple UI mock-ups for each use case are presented to illustrate the user interactions.

3.2 Register Patient

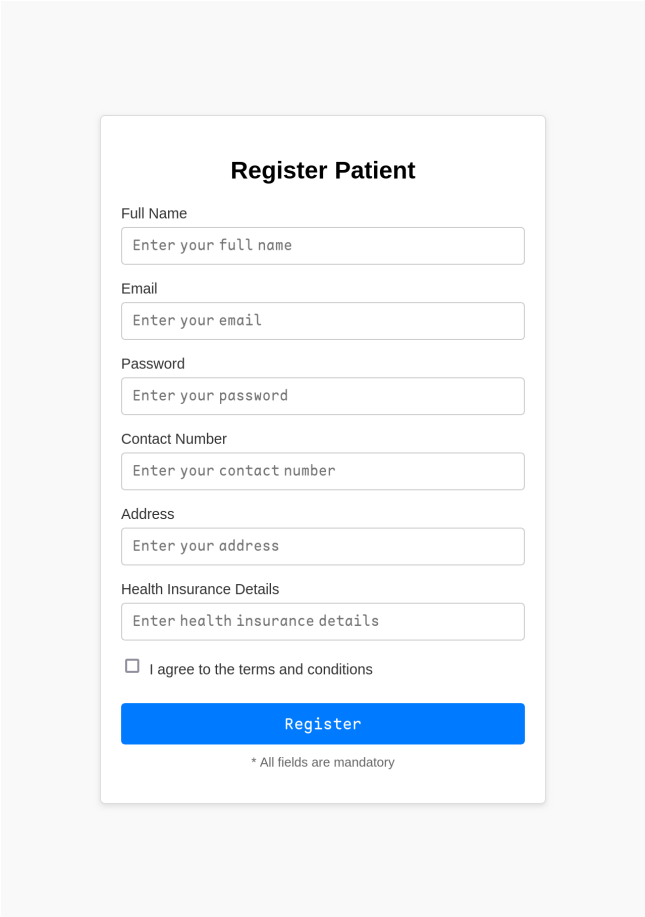

Figure 3: Mock-Up for Register Patient UI

- **Form Fields**: Full name, email, password, contact number, address, health insurance details.
- **Buttons**: Register.
- **Other Elements**: Terms and conditions checkbox.

3.3 Login User

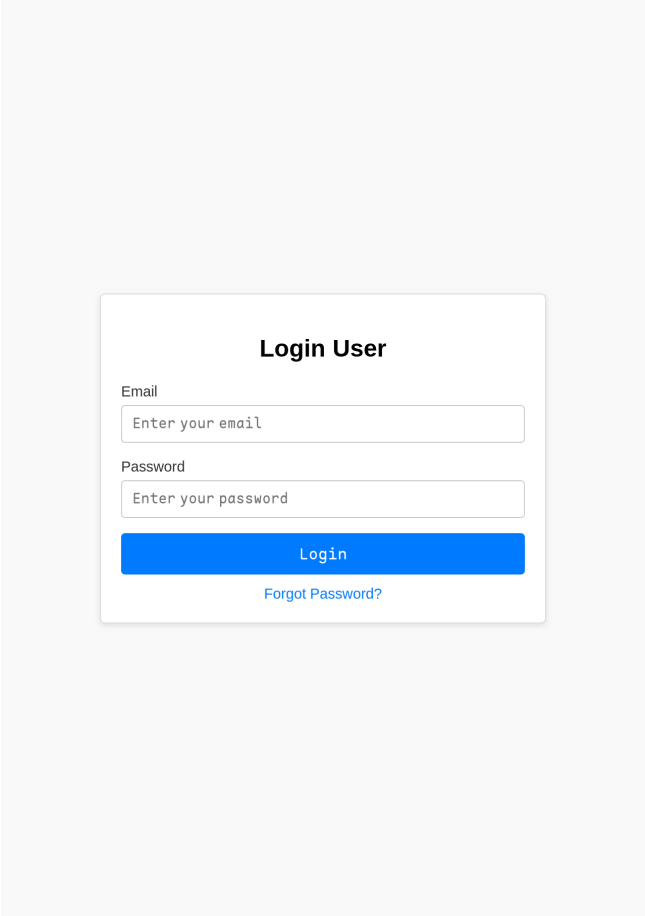

Figure 4: Mock-Up for Login User UI

- **Form Fields**: Email, password.
- **Buttons**: Login, Forgot Password.

3.4 Schedule Appointment

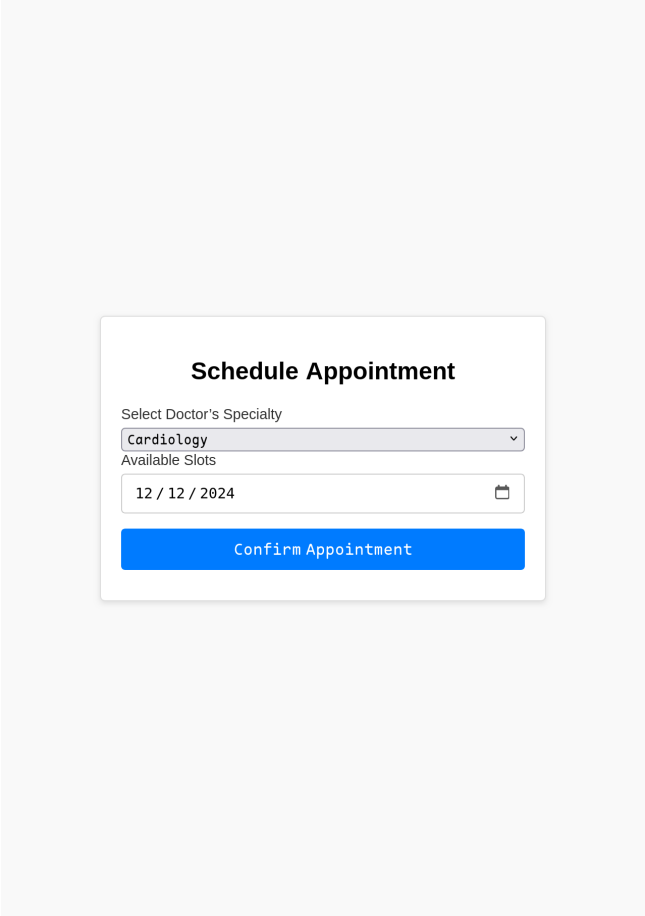

Figure 5: Mock-Up for Schedule Appointment UI

- **Dropdown**: Select doctor's specialty.
- **Calendar View**: Available slots to choose from.
- **Buttons**: Confirm Appointment.

3.5 View Medical Records

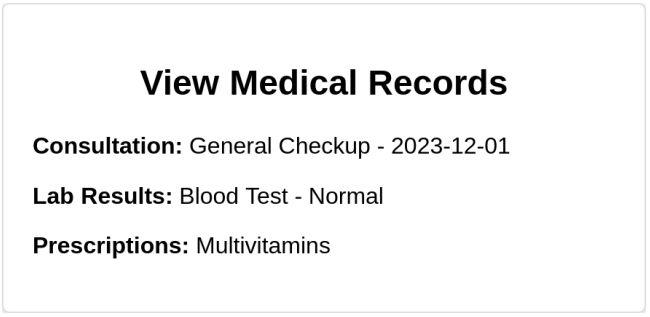

Figure 6: Mock-Up for View Medical Records UI

- **List View**: Displays past consultations, lab results, prescriptions.
- **Read-Only Fields**: Display patient records.

3.6 Update Medical Records

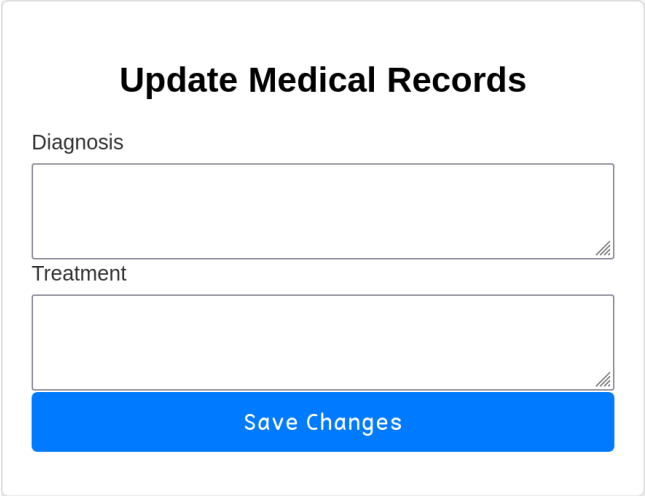

Figure 7: Mock-Up for Update Medical Records UI

- **Editable Fields**: Patient information like diagnosis, treatment.
- **Buttons**: Save Changes.

3.7 Process Payment

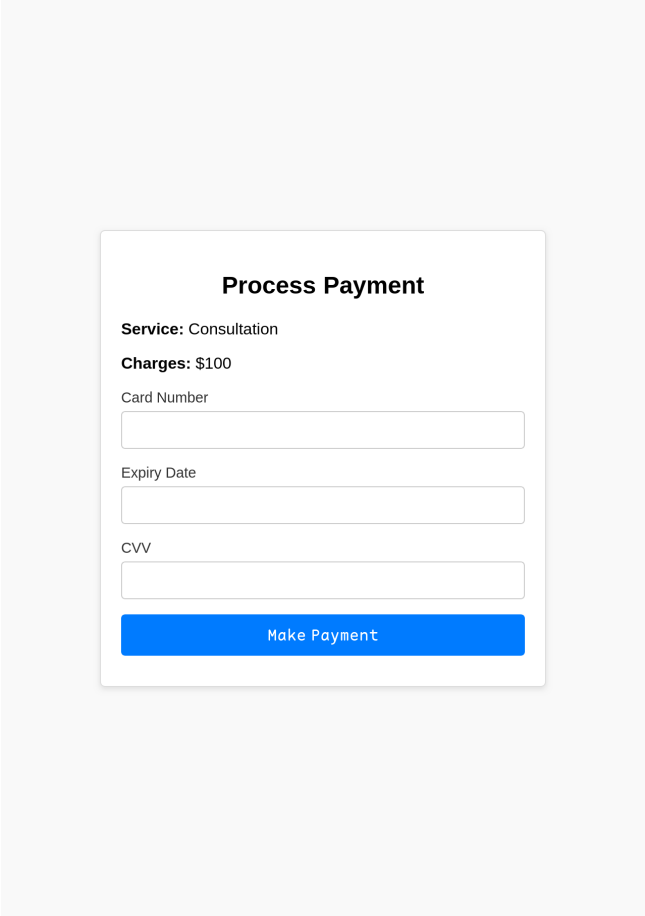

Figure 8: Mock-Up for Process Payment UI

- **Bill Summary subsection**: Details of charges.
- **Payment Information Fields**: Card number, expiry, CVV.
- **Buttons**: Make Payment.

3.8 Manage Staff

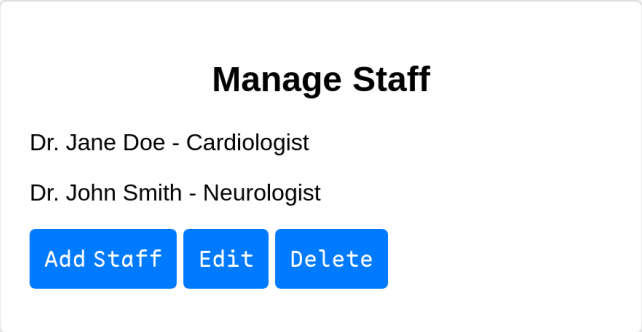

Figure 9: Mock-Up for Manage Staff UI

- **List View**: Displays staff members.
- **Buttons**: Add Staff, Edit, Delete.

3.9 Manage Hospital Resources

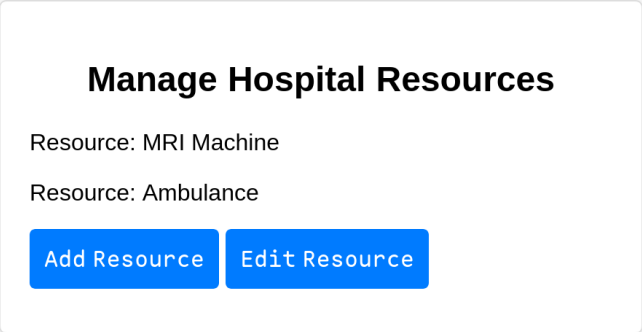

Figure 10: Mock-Up for Manage Hospital Resources UI

- **Dashboard**: List of resources.
- **Buttons**: Add Resource, Edit Resource.

3.10 Generate Report

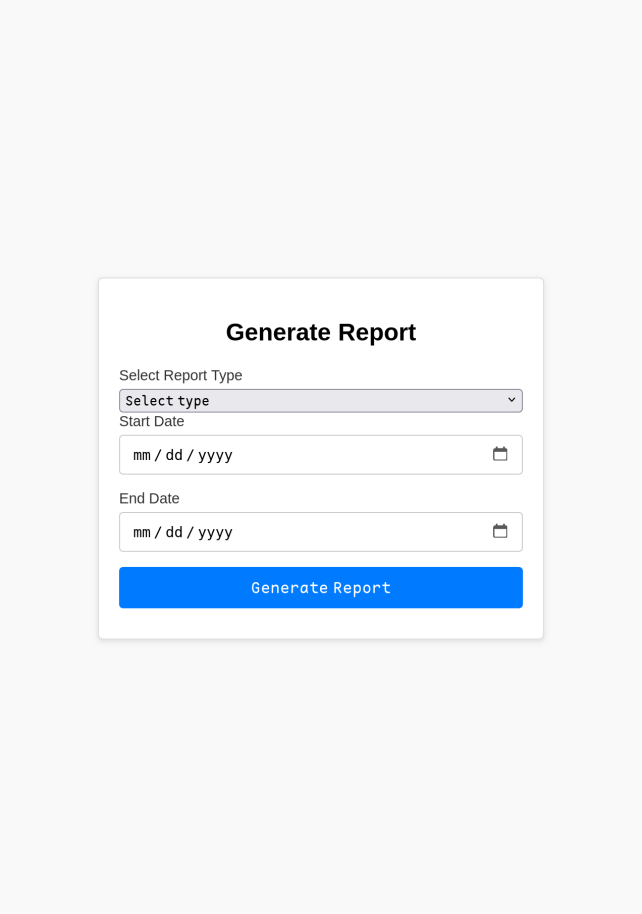

Figure 11: Mock-Up for Generate Report UI

- **Select Report Type**: Dropdown.
- **Date Range**: Start and end date.
- **Buttons**: Generate Report.

3.11 Send Notification

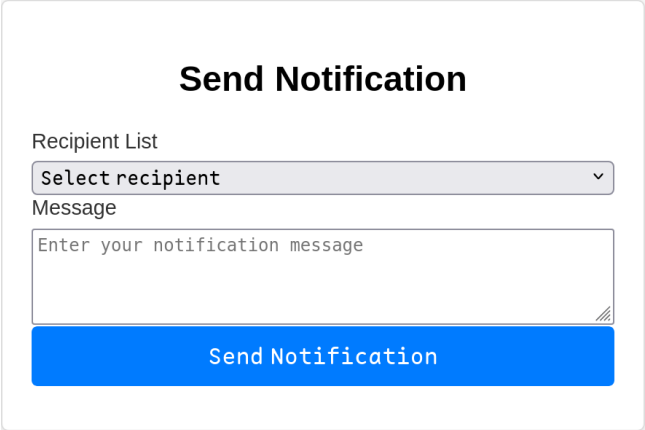

Figure 12: Mock-Up for Send Notification UI

- **Recipient List**: Patients, doctors.
- **Message Input Box**: Text area for entering notification content.
- **Buttons**: Send Notification.

4 Interaction Diagrams for Each Use Case

4.1 Overview

The process flow of the system. The diagrams include high-level - 0, key action - 1, and detailed system process - 2 levels for each use case.

4.2 Register Patient

4.2.1 Level 0 Interaction Diagram (High-Level Overview)

- **Actors Involved**: Patient
- **Action**: Patient registers for an account.
- **System Interaction**: System saves patient information.

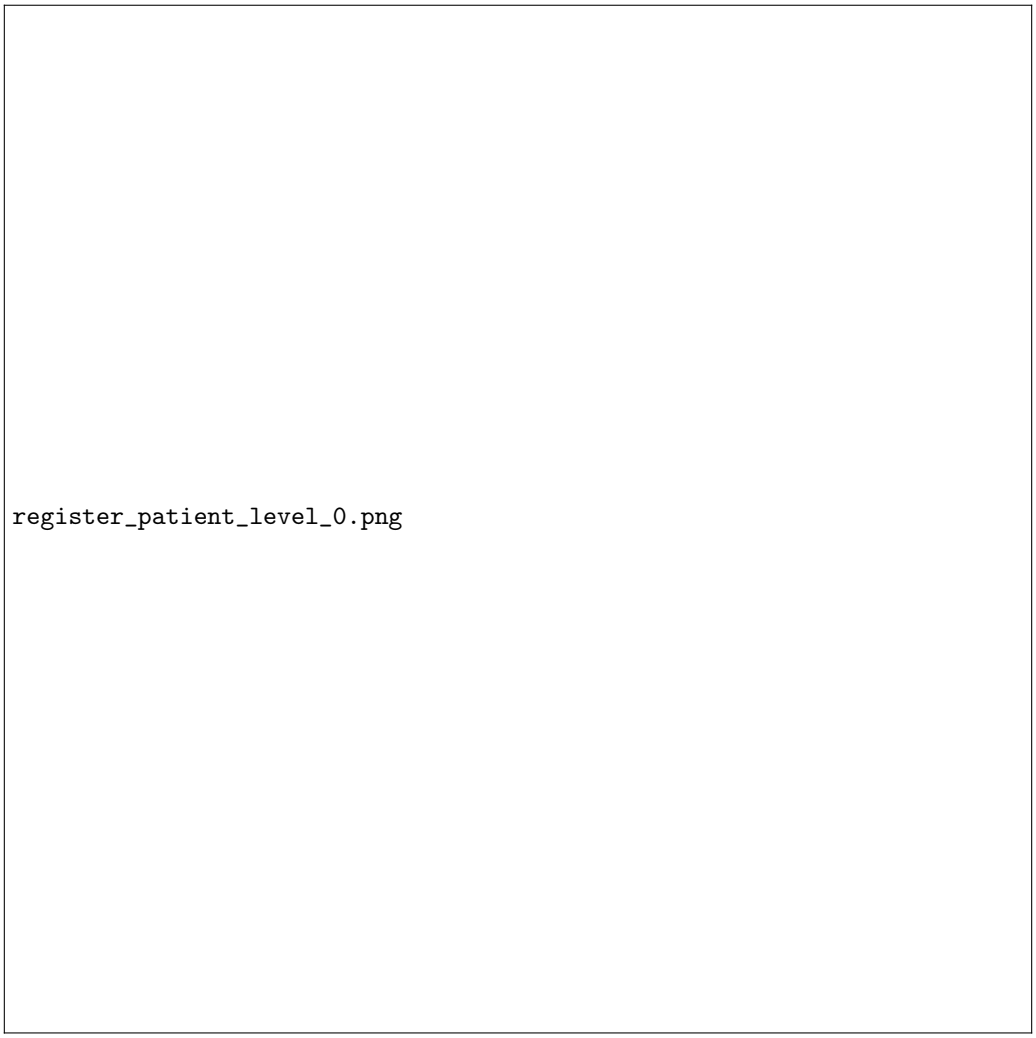

Figure 13: Level 0 Interaction Diagram for Register Patient

4.2.2 Level 1 Interaction Diagram (Key Actions)

- **Step 1**: Patient enters personal details, email, and password.
- **Step 2**: System validates the information.
- **Step 3**: System creates a new patient profile and stores it.
- **Step 4**: System confirms registration by displaying a success message.

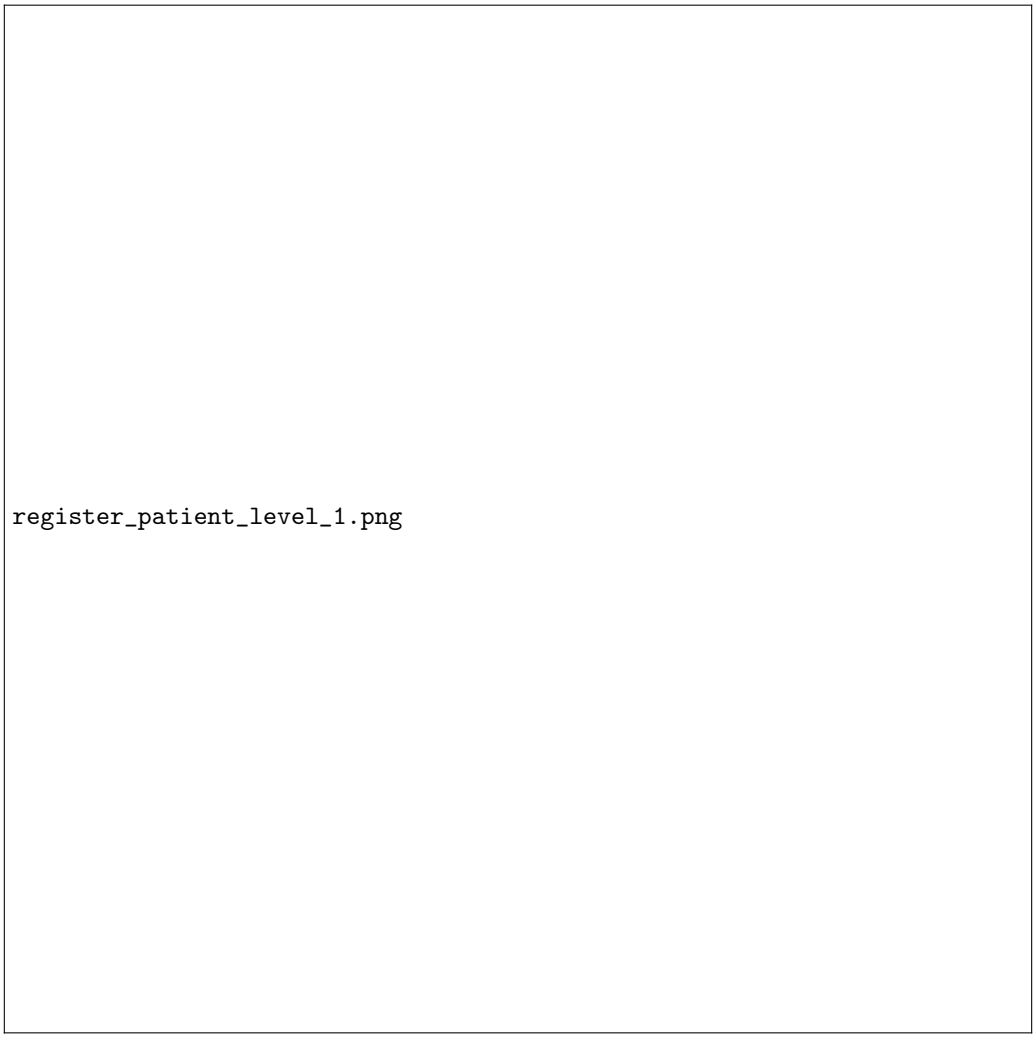

Figure 14: Level 1 Interaction Diagram for Register Patient

4.2.3 Level 2 Interaction Diagram (Detailed System Processes)

- **Step 1**: Patient enters personal information, email, password, and health insurance details.
- **Step 2**: System validates if the email is unique and checks for proper input format.
- **Step 3**: System interacts with the Database to store patient details.
- **Step 4**: System sends a registration confirmation email using the Notification Service.
- **Step 5**: System displays confirmation message to the patient.

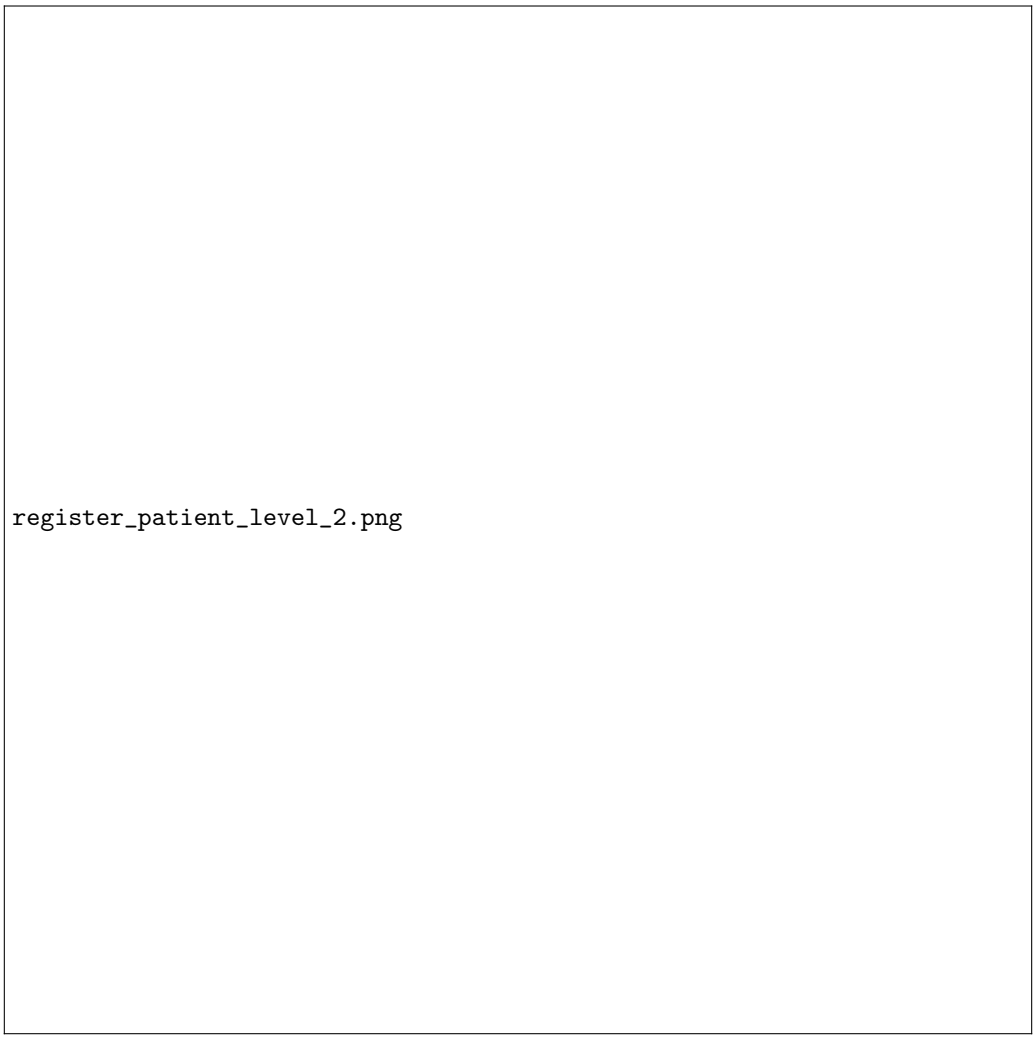

Figure 15: Level 2 Interaction Diagram for Register Patient

4.3 Login User

4.3.1 Level 0 Interaction Diagram (High-Level Overview)

- **Actors Involved**: Patient, Doctor, Administrator
- **Action**: Actor logs in to the system.
- **System Interaction**: System validates credentials and grants access.

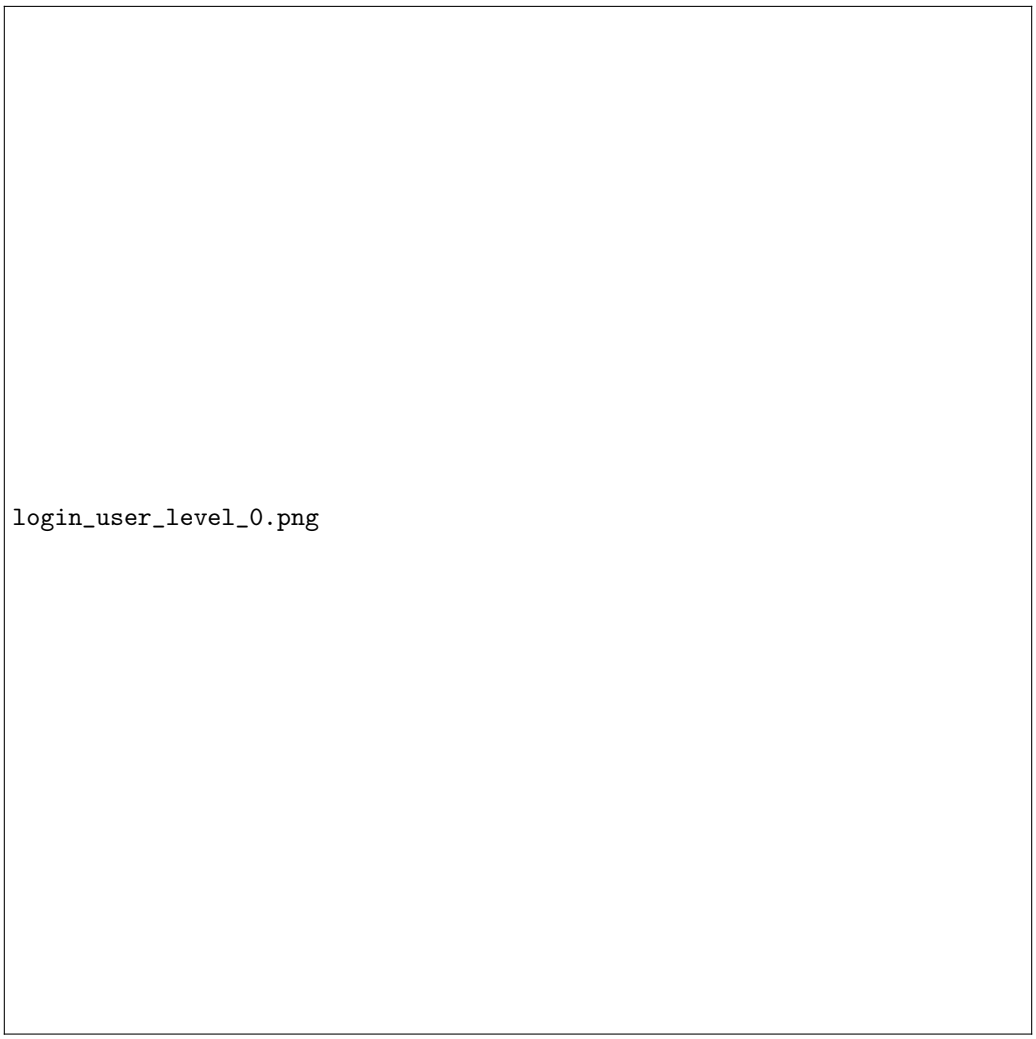

Figure 16: Level 0 Interaction Diagram for Login User

4.3.2 Level 1 Interaction Diagram (Key Actions)

- **Step 1**: Actor enters username and password.
- **Step 2**: System validates credentials.
- **Step 3**: System grants access if credentials are correct.
- **Step 4**: System displays appropriate dashboard based on the user role (Patient, Doctor, or Administrator).

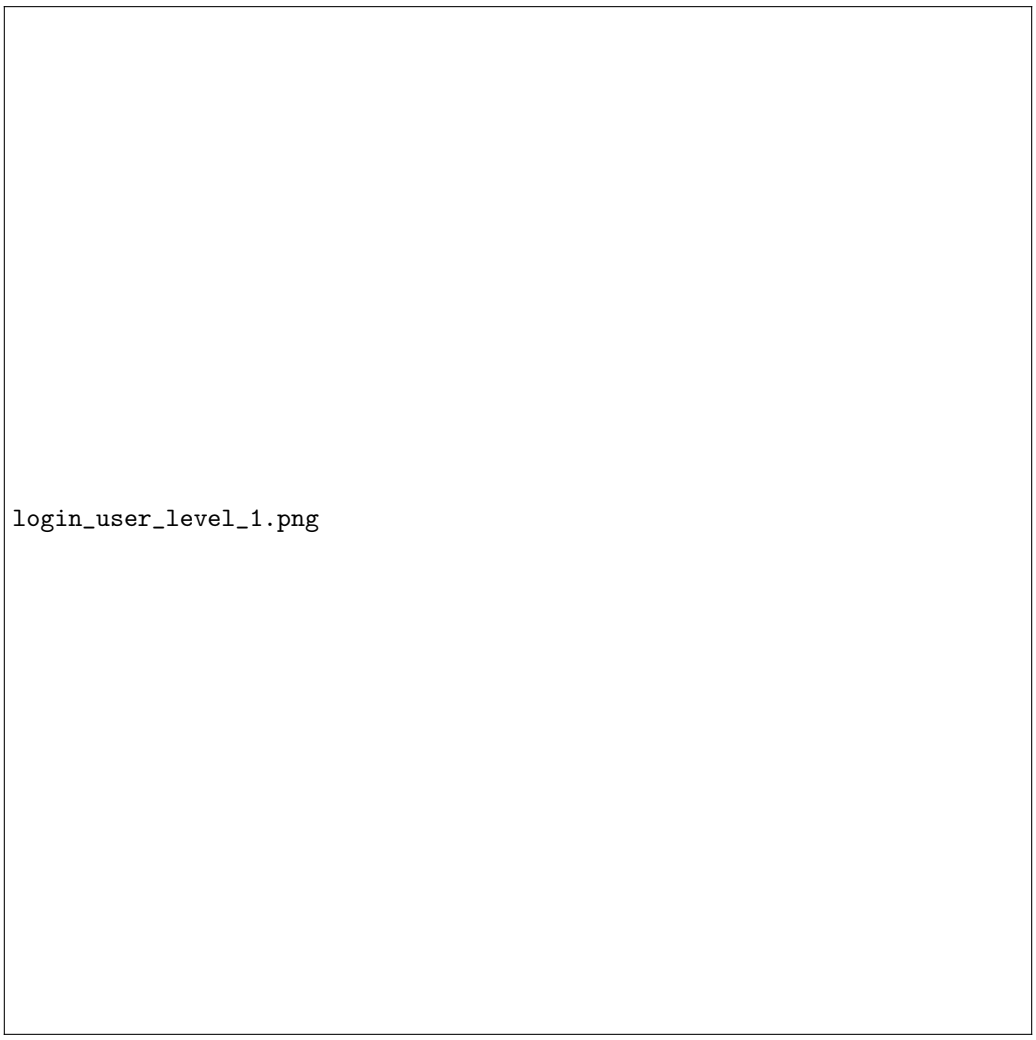

Figure 17: Level 1 Interaction Diagram for Login User

4.3.3 Level 2 Interaction Diagram (Detailed System Processes)

- **Step 1**: Actor inputs username and password.
- **Step 2**: System checks credentials against the User Database.
- **Step 3**: If credentials match:
  - System creates a session for the user.
  - System interacts with the Role Management Module to determine the actor's role.
  - System loads the appropriate dashboard (patient, doctor, or admin).
- **Step 4**: If credentials do not match, System displays an error message and allows the actor to try again.

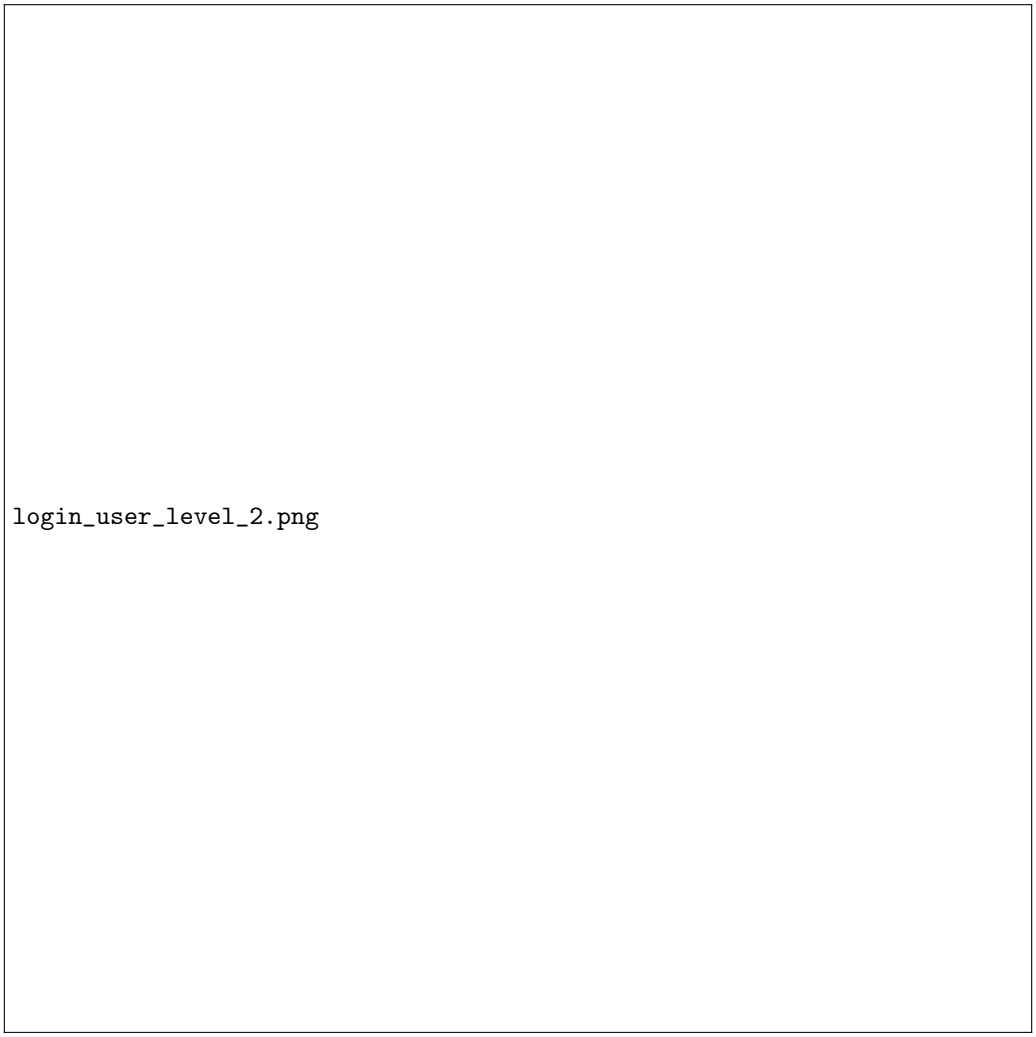

Figure 18: Level 2 Interaction Diagram for Login User

4.4 Schedule Appointment

4.4.1 Level 0 Interaction Diagram (High-Level Overview)

- **Actors Involved**: Patient
- **Action**: Patient schedules an appointment.
- **System Interaction**: System confirms the appointment.

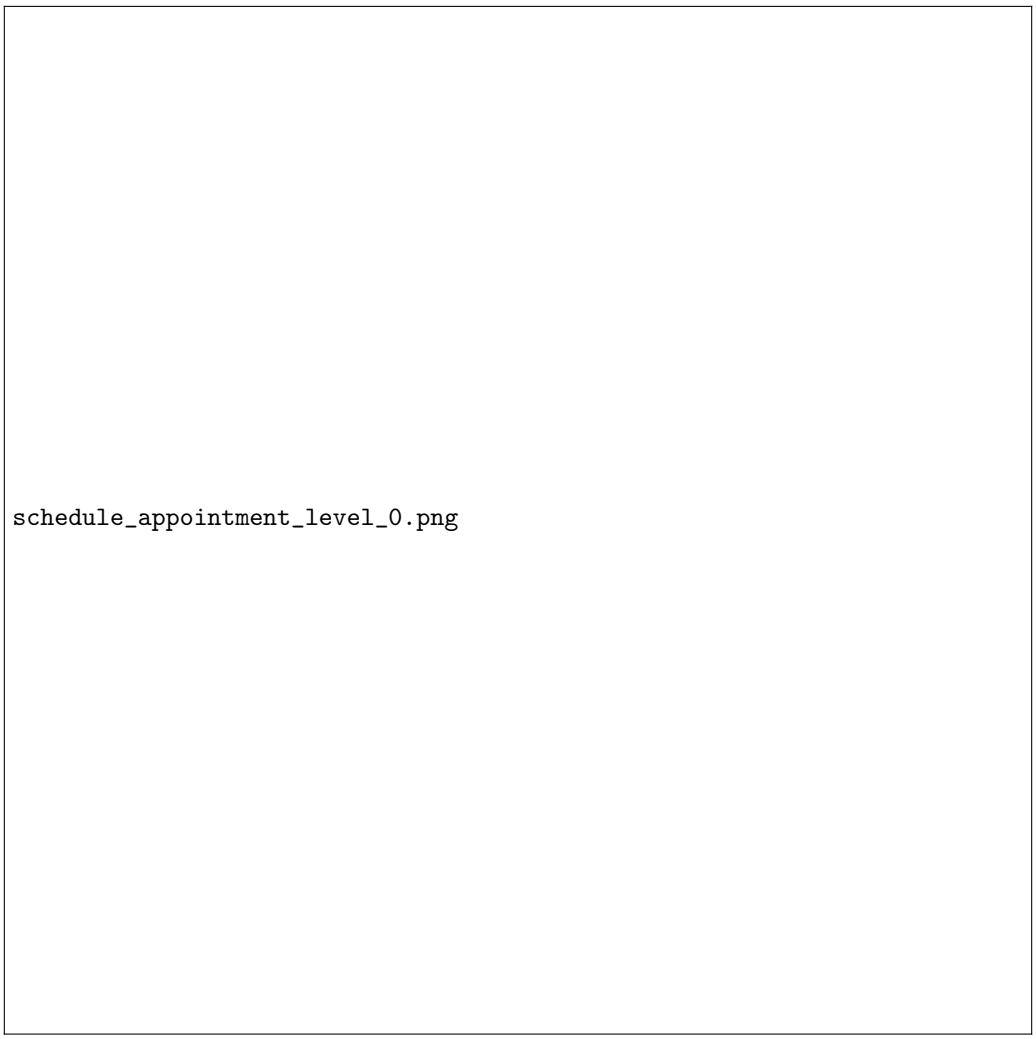

Figure 19: Level 0 Interaction Diagram for Schedule Appointment

4.4.2 Level 1 Interaction Diagram (Key Actions)

- **Step 1**: Patient views list of available doctors.
- **Step 2**: Patient selects doctor and appointment slot.
- **Step 3**: System confirms the appointment.
- **Step 4**: System sends appointment confirmation to the patient.

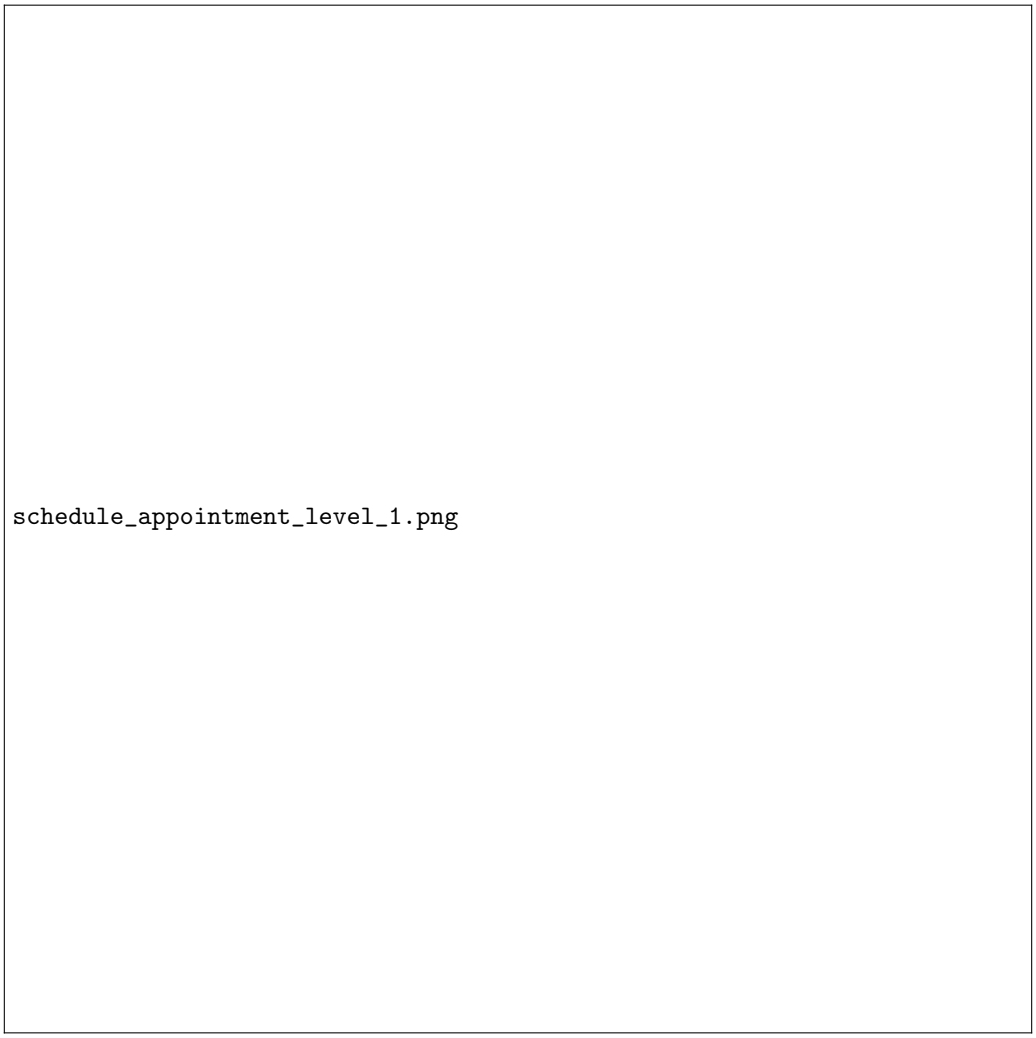

Figure 20: Level 1 Interaction Diagram for Schedule Appointment

4.4.3 Level 2 Interaction Diagram (Detailed System Processes)

- **Step 1**: Patient logs in and requests to schedule an appointment.
- **Step 2**: System retrieves doctor availability from the Appointment Management System.
- **Step 3**: Patient selects preferred doctor and time slot.
- **Step 4**: System interacts with the Database to book the appointment.
- **Step 5**: System sends appointment confirmation to the Notification Service.
- **Step 6**: Notification Service sends an email and SMS to the patient.

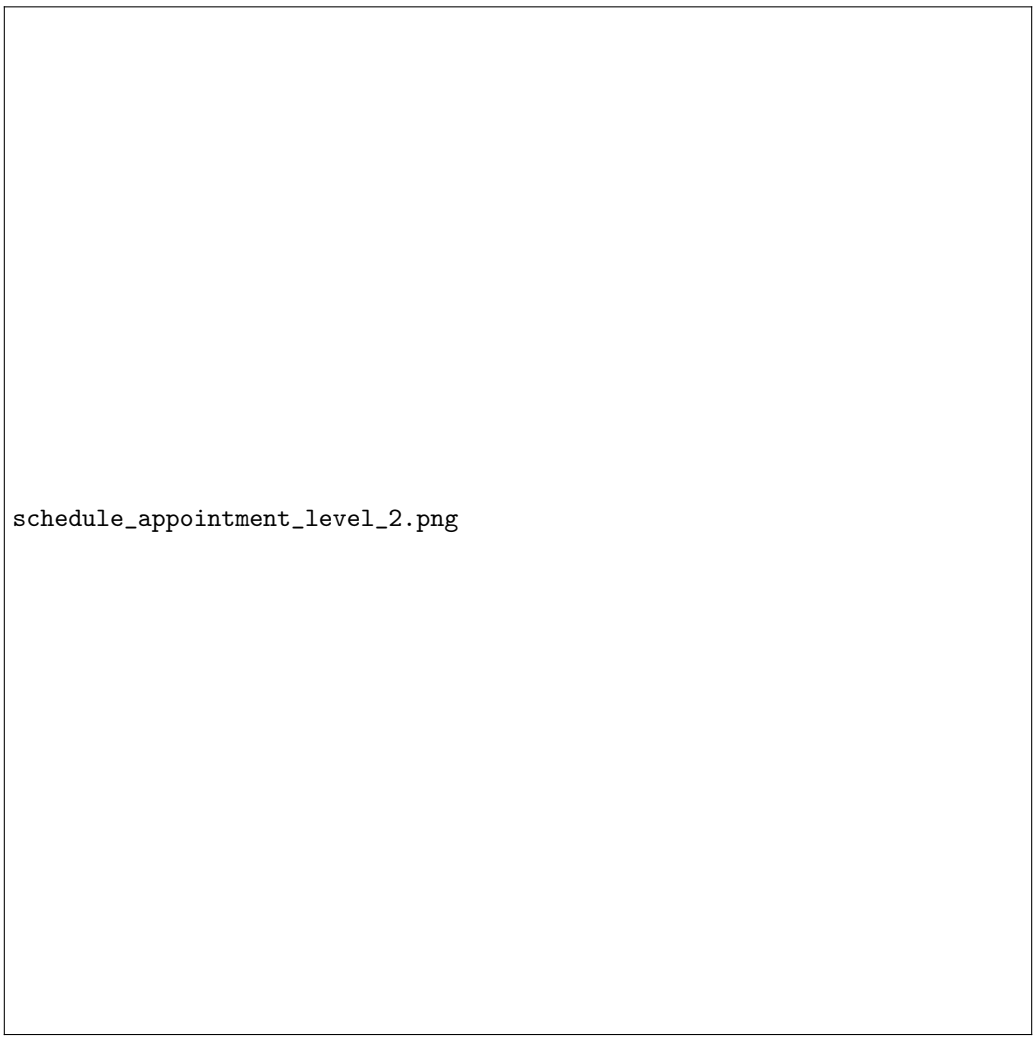

Figure 21: Level 2 Interaction Diagram for Schedule Appointment

4.5 View Medical Records

4.5.1 Level 0 Interaction Diagram (High-Level Overview)

- **Actors Involved**: Patient, Doctor
- **Action**: Patient or Doctor views medical records.
- **System Interaction**: System retrieves and displays medical records.

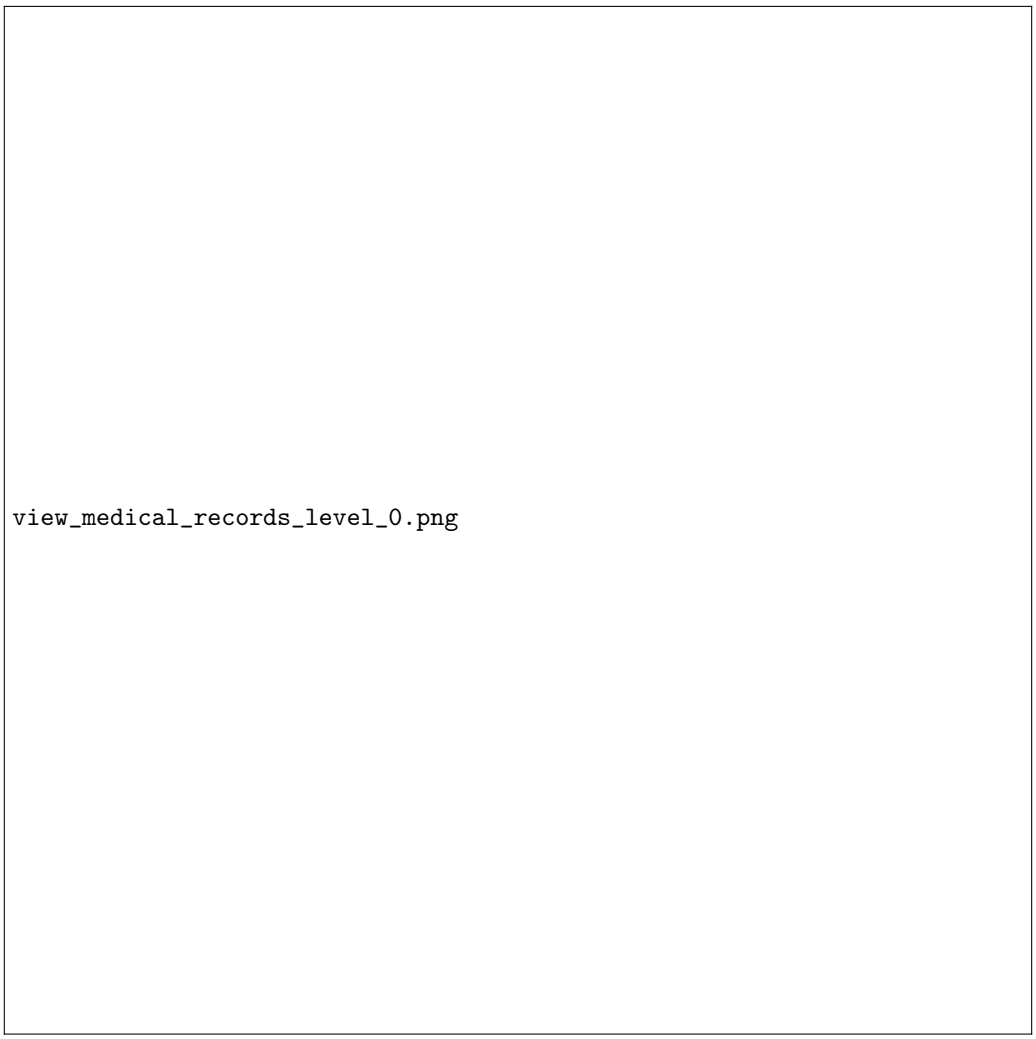

Figure 22: Level 0 Interaction Diagram for View Medical Records

4.5.2 Level 1 Interaction Diagram (Key Actions)

- **Step 1**: Actor requests to view medical records.
- **Step 2**: System retrieves records from the database.
- **Step 3**: System displays the medical records to the actor.

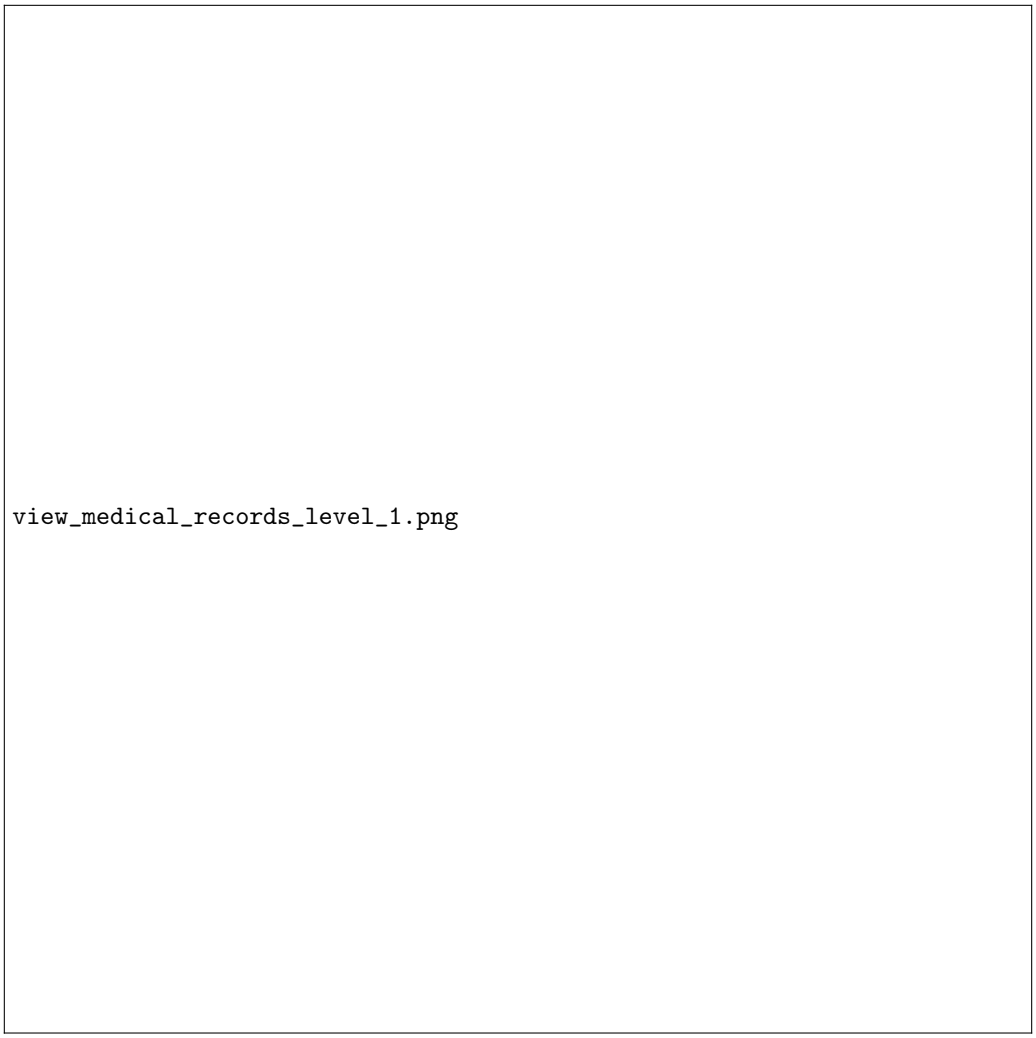

Figure 23: Level 1 Interaction Diagram for View Medical Records

4.5.3 Level 2 Interaction Diagram (Detailed System Processes)

- **Step 1**: Actor logs in and selects the "View Medical Records" option.
- **Step 2**: System retrieves MedicalRecord from the Database based on patientID.
- **Step 3**: System verifies access rights:
  - If Patient, only read access is granted.
  - If Doctor, full access is granted.
- **Step 4**: System displays the records to the actor with appropriate access rights.

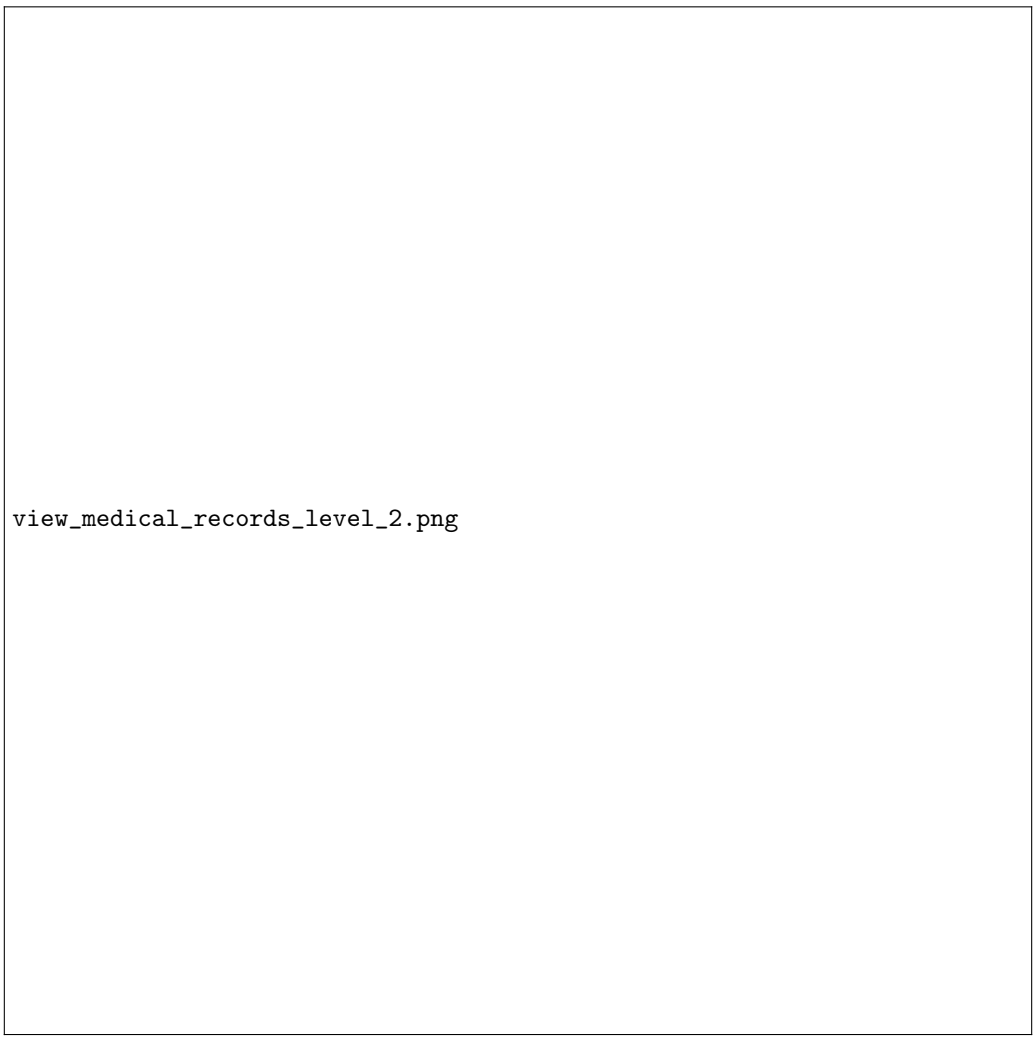

Figure 24: Level 2 Interaction Diagram for View Medical Records

4.6 Update Medical Records

4.6.1 Level 0 Interaction Diagram (High-Level Overview)

- **Actors Involved**: Doctor
- **Action**: Doctor updates patient medical records.
- **System Interaction**: System saves updated medical records.

Figure 25: Level 0 Interaction Diagram for Update Medical Records

4.6.2 Level 1 Interaction Diagram (Key Actions)

- **Step 1**: Doctor views patient medical records.
- **Step 2**: Doctor makes updates to patient diagnosis or prescription.
- **Step 3**: System saves the updated medical records.

Figure 26: Level 1 Interaction Diagram for Update Medical Records

4.6.3 Level 2 Interaction Diagram (Detailed System Processes)

- **Step 1**: Doctor logs in and selects a patient to update records.
- **Step 2**: System retrieves MedicalRecord from the Database based on patientID.
- **Step 3**: Doctor updates diagnosis or prescription information.
- **Step 4**: System validates the data and saves it to the Database.
- **Step 5**: System notifies the Patient via Notification Service that their records have been updated.

Figure 27: Level 2 Interaction Diagram for Update Medical Records

4.7 Process Payment

4.7.1 Level 0 Interaction Diagram (High-Level Overview)

- **Actors Involved**: Patient
- **Action**: Patient makes a payment.
- **System Interaction**: System processes the payment.

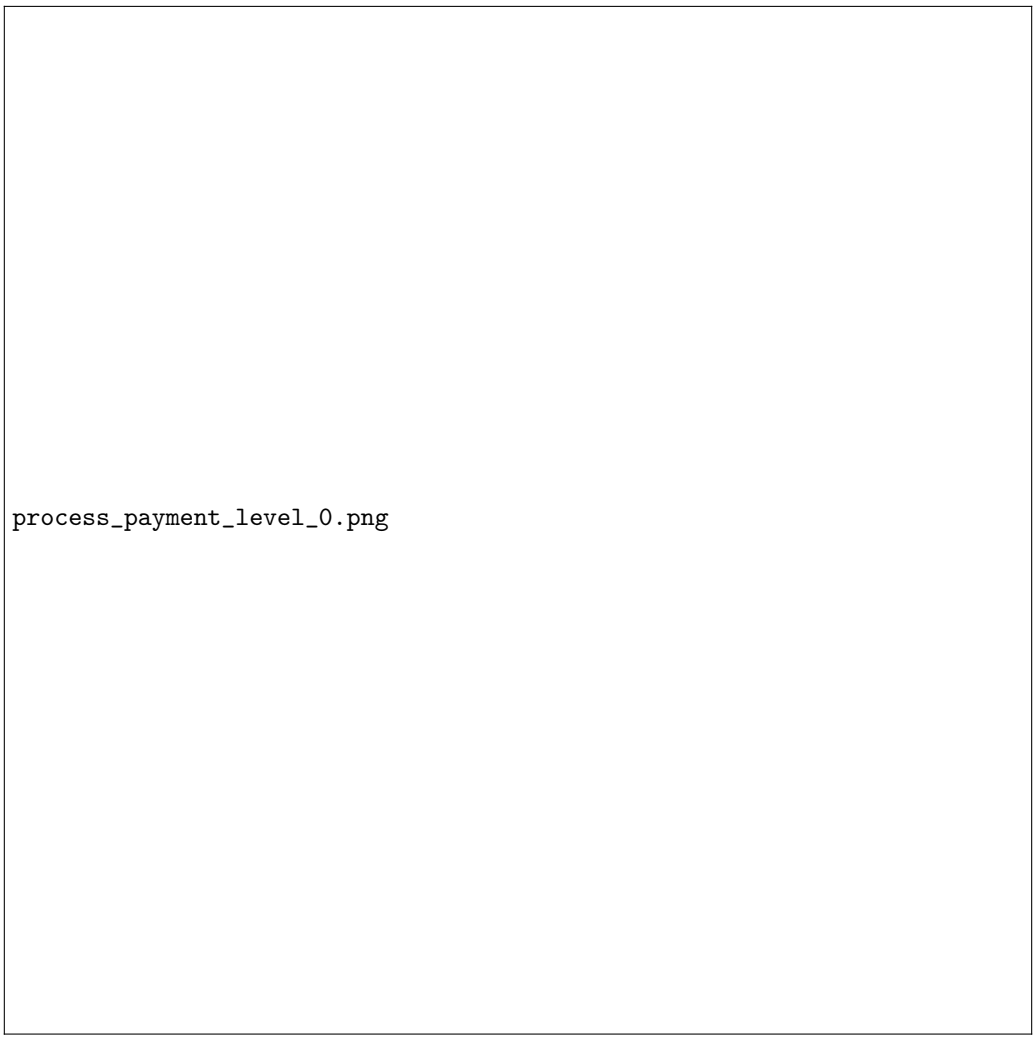

Figure 28: Level 0 Interaction Diagram for Process Payment

4.7.2 Level 1 Interaction Diagram (Key Actions)

- **Step 1**: Patient views billing details.
- **Step 2**: Patient selects payment method and submits payment.
- **Step 3**: System processes payment and generates a receipt.

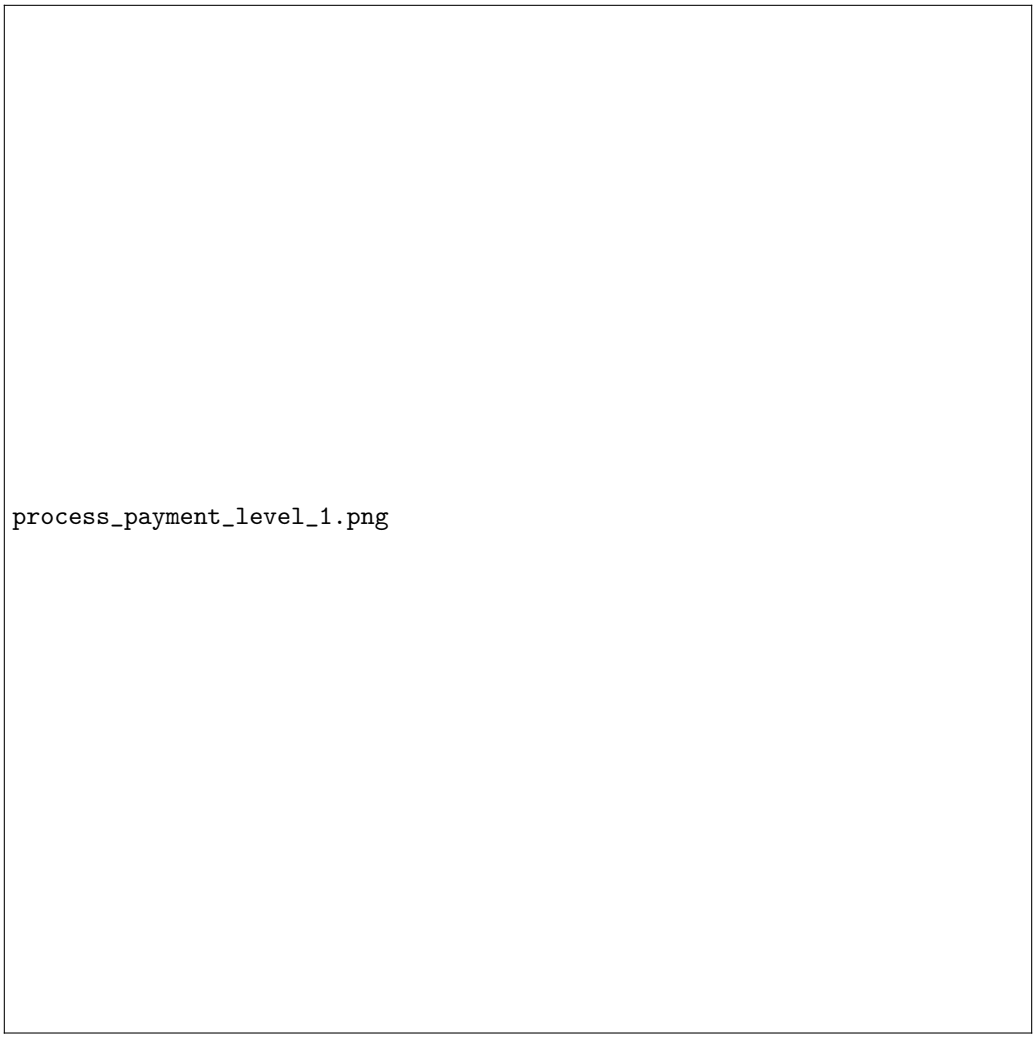

Figure 29: Level 1 Interaction Diagram for Process Payment

4.7.3 Level 2 Interaction Diagram (Detailed System Processes)

- **Step 1**: Patient logs in and selects "View Billing Details."
- **Step 2**: System retrieves outstanding bills from the Billing Module.
- **Step 3**: Patient enters payment information.
- **Step 4**: System interacts with Payment Gateway to process the payment.
- **Step 5**: If successful:
  - System updates the Billing Module with payment status.
  - System sends a receipt to Notification Service for email delivery to the patient.
- **Step 6**: If unsuccessful, System notifies the patient of payment failure.

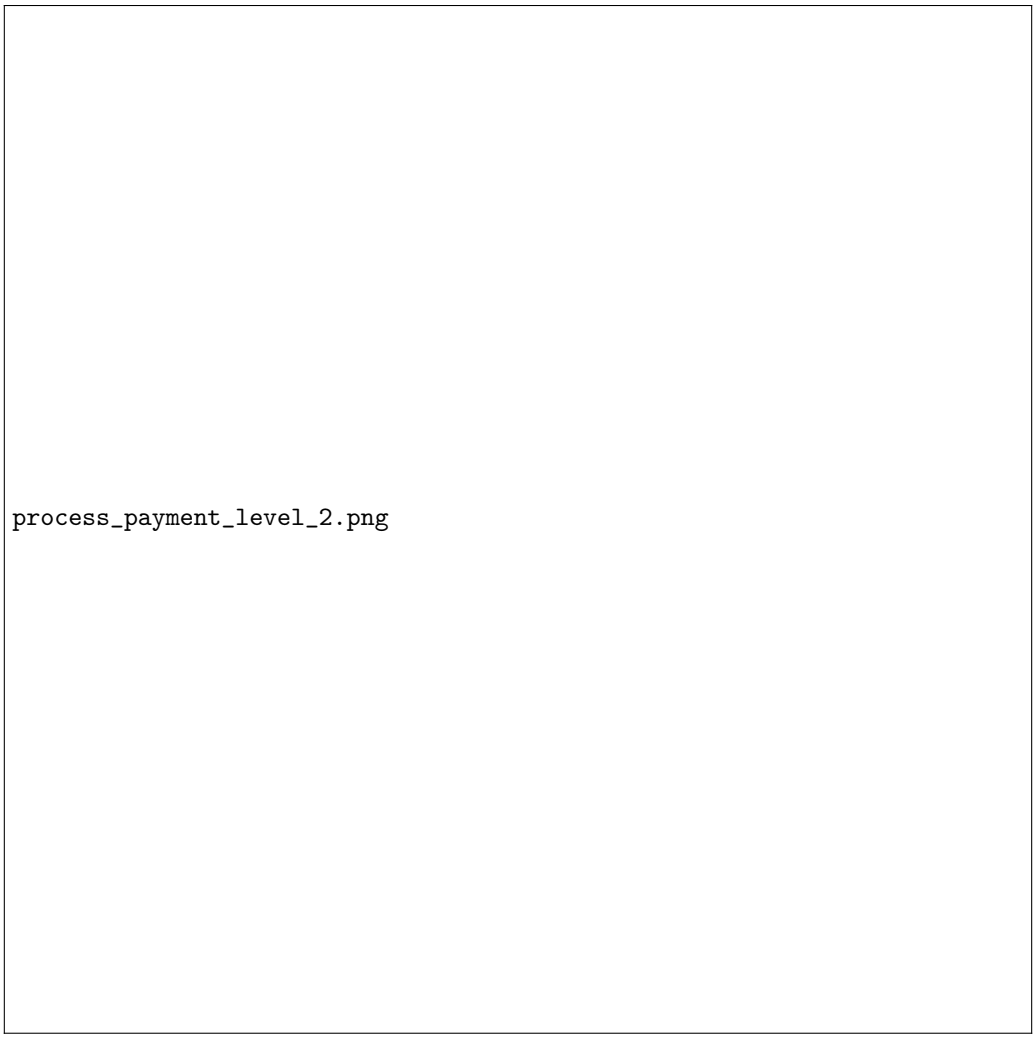

Figure 30: Level 2 Interaction Diagram for Process Payment

4.8 Manage Staff

4.8.1 Level 0 Interaction Diagram (High-Level Overview)

- **Actors Involved**: Administrator
- **Action**: Administrator manages staff details.
- **System Interaction**: System saves updated staff information.

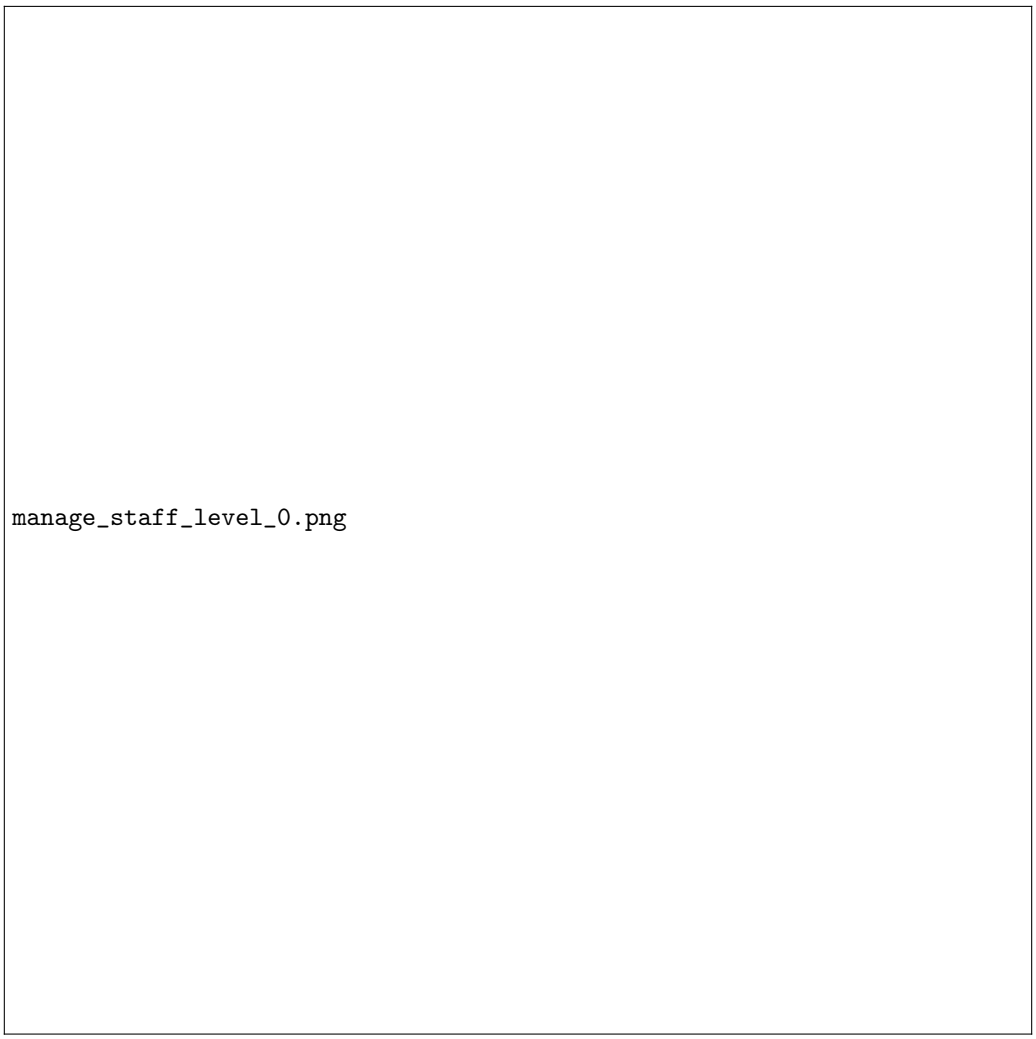

Figure 31: Level 0 Interaction Diagram for Manage Staff

4.8.2 Level 1 Interaction Diagram (Key Actions)

- **Step 1**: Administrator adds or updates staff information.
- **Step 2**: System saves the information to the database.
- **Step 3**: System confirms changes.

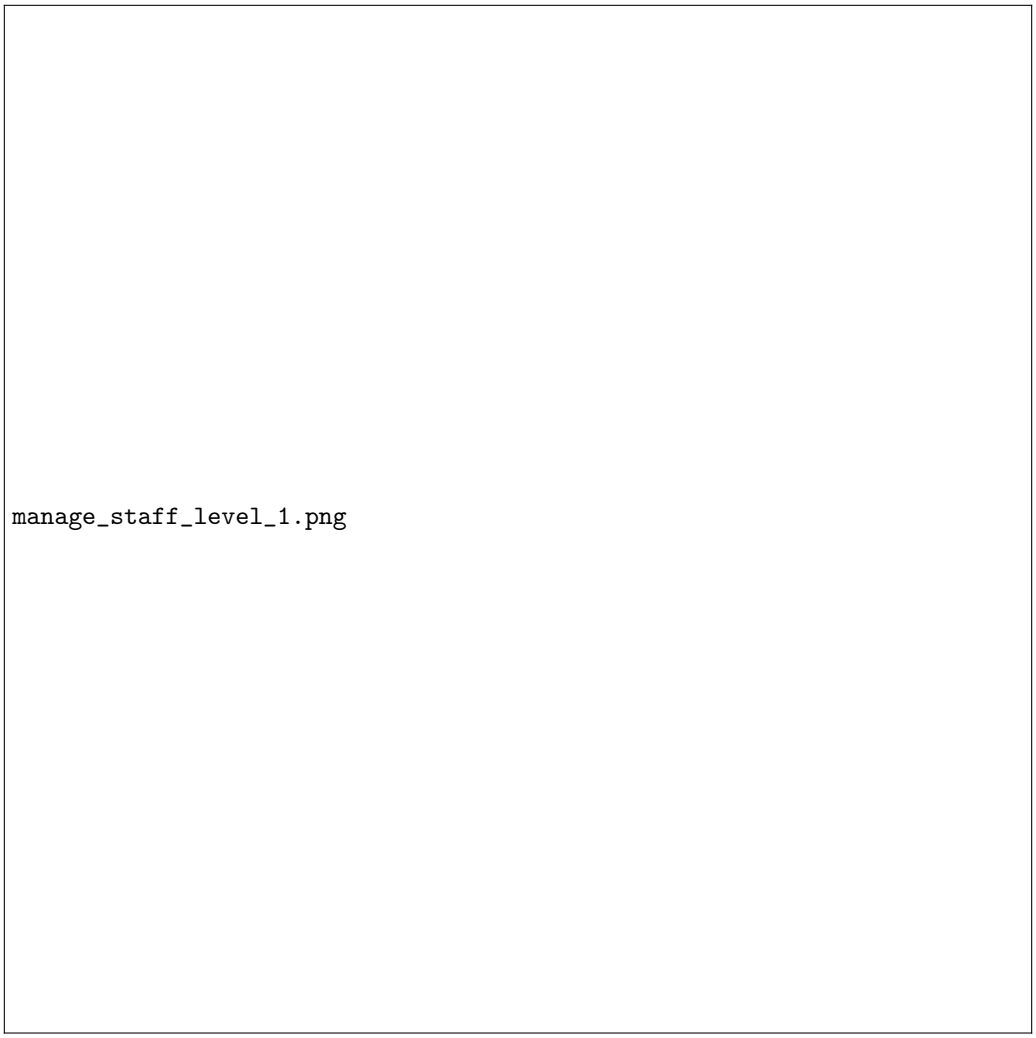

Figure 32: Level 1 Interaction Diagram for Manage Staff

4.8.3 Level 2 Interaction Diagram (Detailed System Processes)

- **Step 1**: Administrator logs in and selects the "Manage Staff" option.
- **Step 2**: Administrator adds or edits details for a staff member.
- **Step 3**: System validates the input data.
- **Step 4**: System updates Staff Database with the new or edited information.
- **Step 5**: System confirms success and logs the changes.

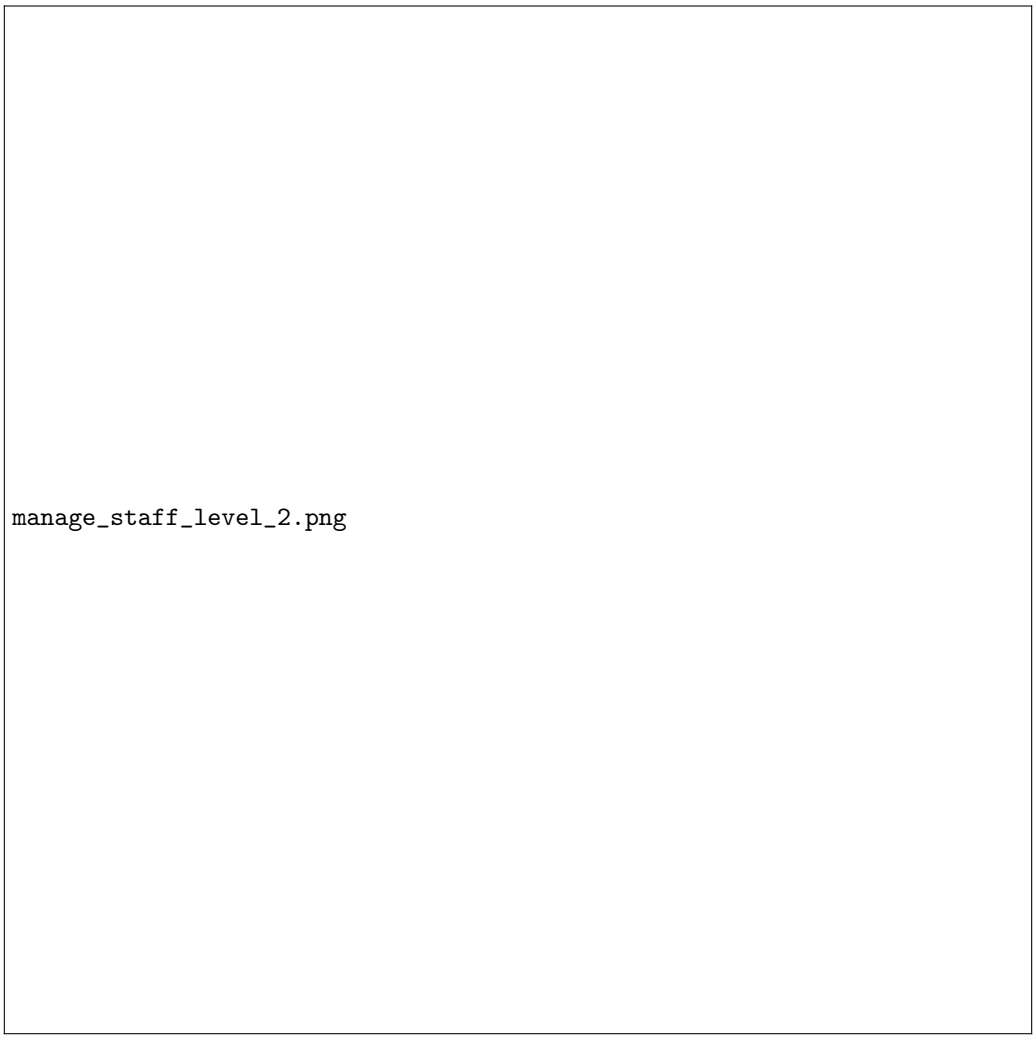

Figure 33: Level 2 Interaction Diagram for Manage Staff

4.9 Manage Hospital Resources

4.9.1 Level 0 Interaction Diagram (High-Level Overview)

- **Actors Involved**: Administrator
- **Action**: Administrator manages hospital resources.
- **System Interaction**: System updates resource inventory.

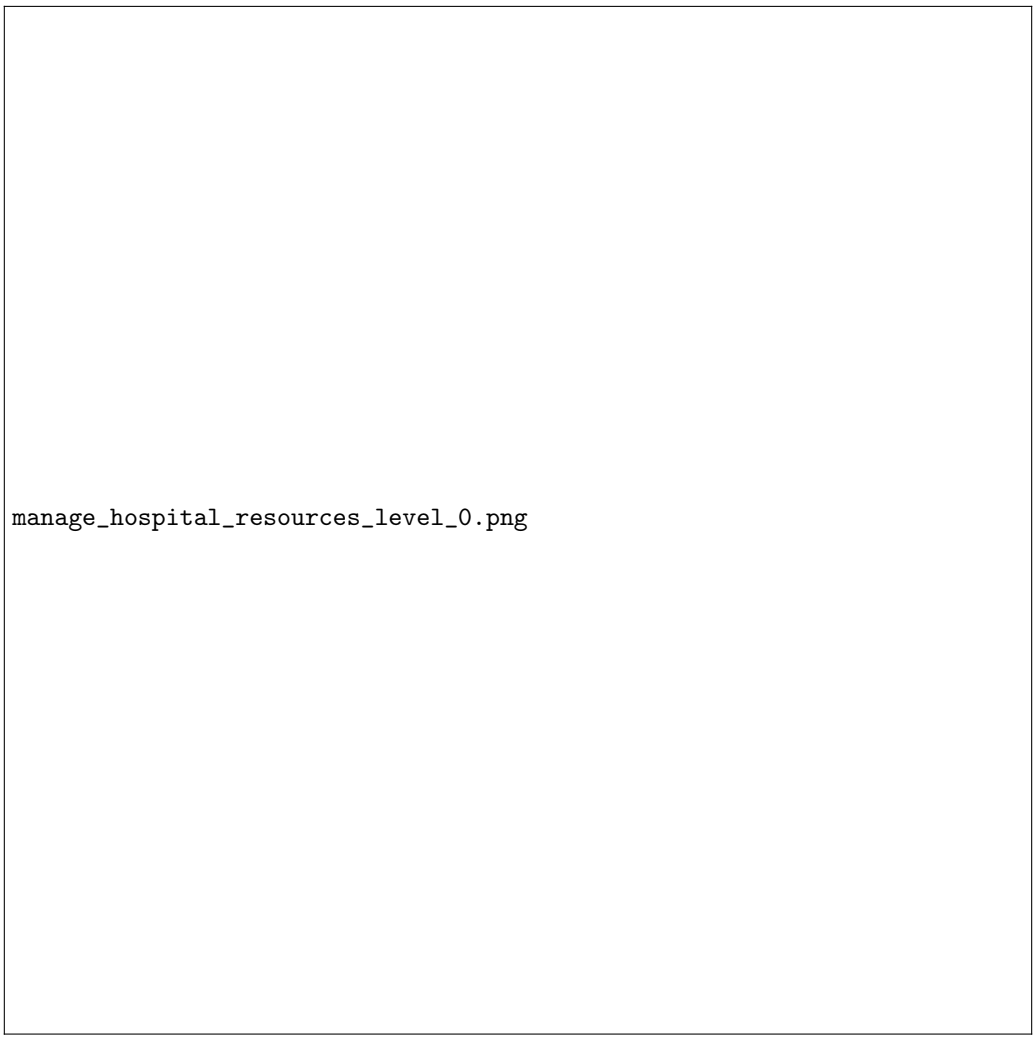

Figure 34: Level 0 Interaction Diagram for Manage Hospital Resources

4.9.2 Level 1 Interaction Diagram (Key Actions)

- **Step 1**: Administrator selects resource management.
- **Step 2**: Administrator adds or updates resource information.
- **Step 3**: System updates the resource database.

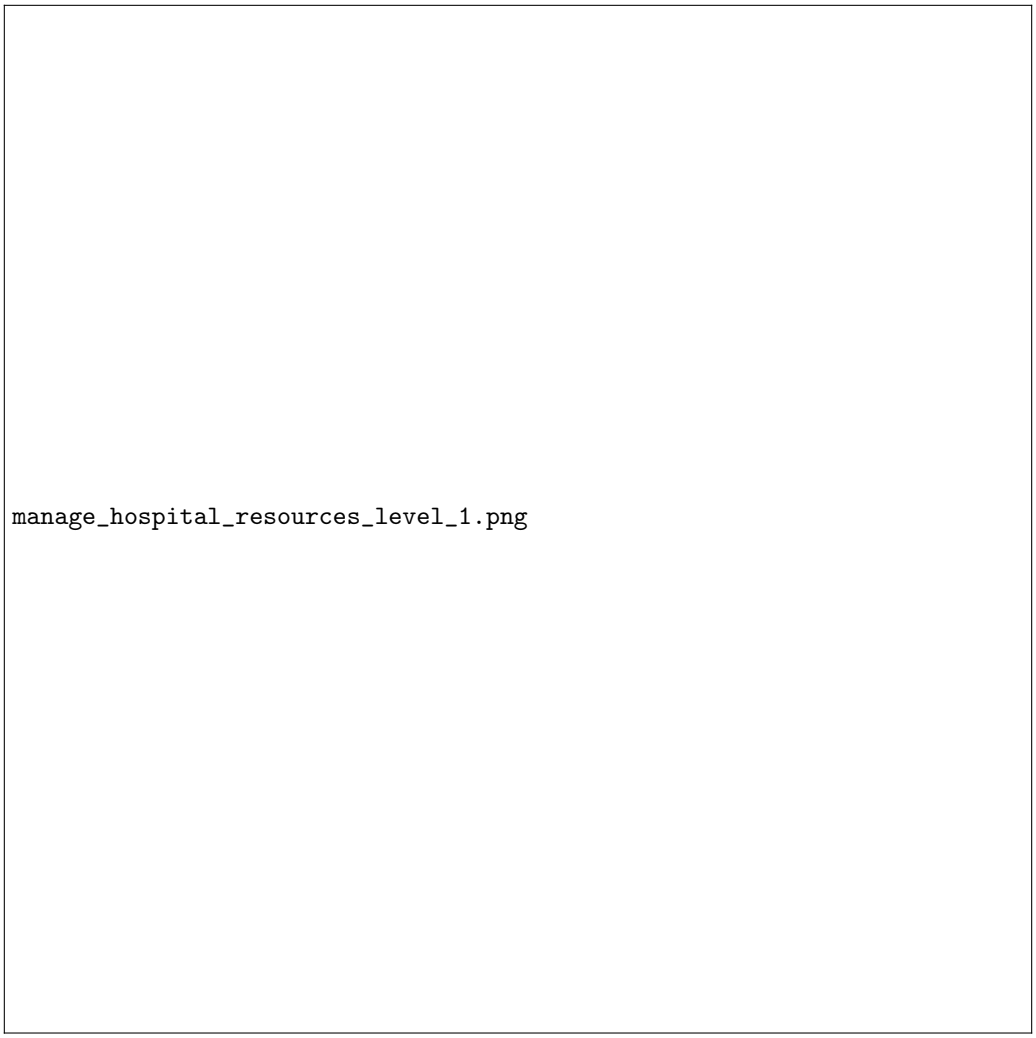

Figure 35: Level 1 Interaction Diagram for Manage Hospital Resources

4.9.3 Level 2 Interaction Diagram (Detailed System Processes)

- **Step 1**: Administrator logs in and selects "Manage Resources."
- **Step 2**: System retrieves current resources from the Resource Database.
- **Step 3**: Administrator adds a new resource or updates an existing one.
- **Step 4**: System validates and saves the changes to the Resource Database.
- **Step 5**: System confirms the update to the administrator.

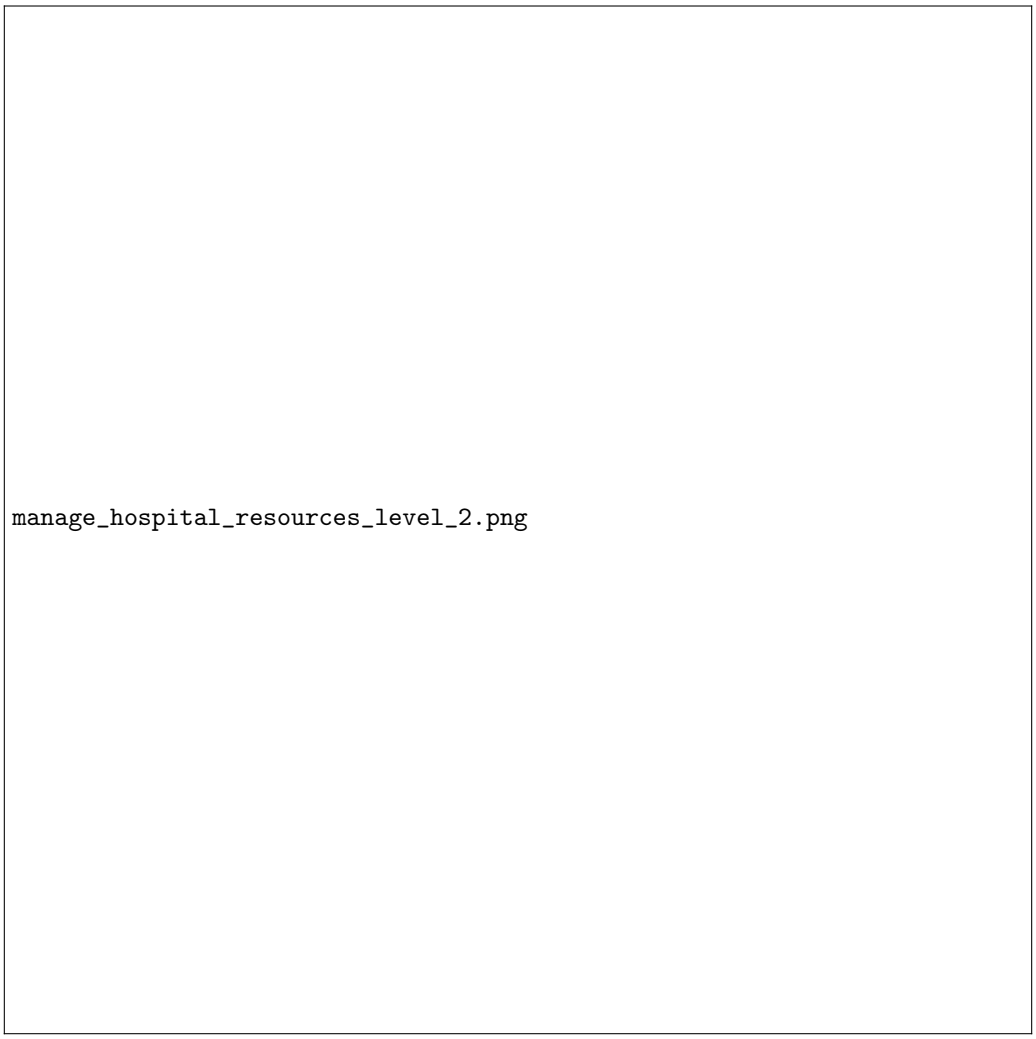

Figure 36: Level 2 Interaction Diagram for Manage Hospital Resources

4.10 Generate Report

4.10.1 Level 0 Interaction Diagram (High-Level Overview)

- **Actors Involved**: Administrator
- **Action**: Administrator generates a report.
- **System Interaction**: System produces the report.

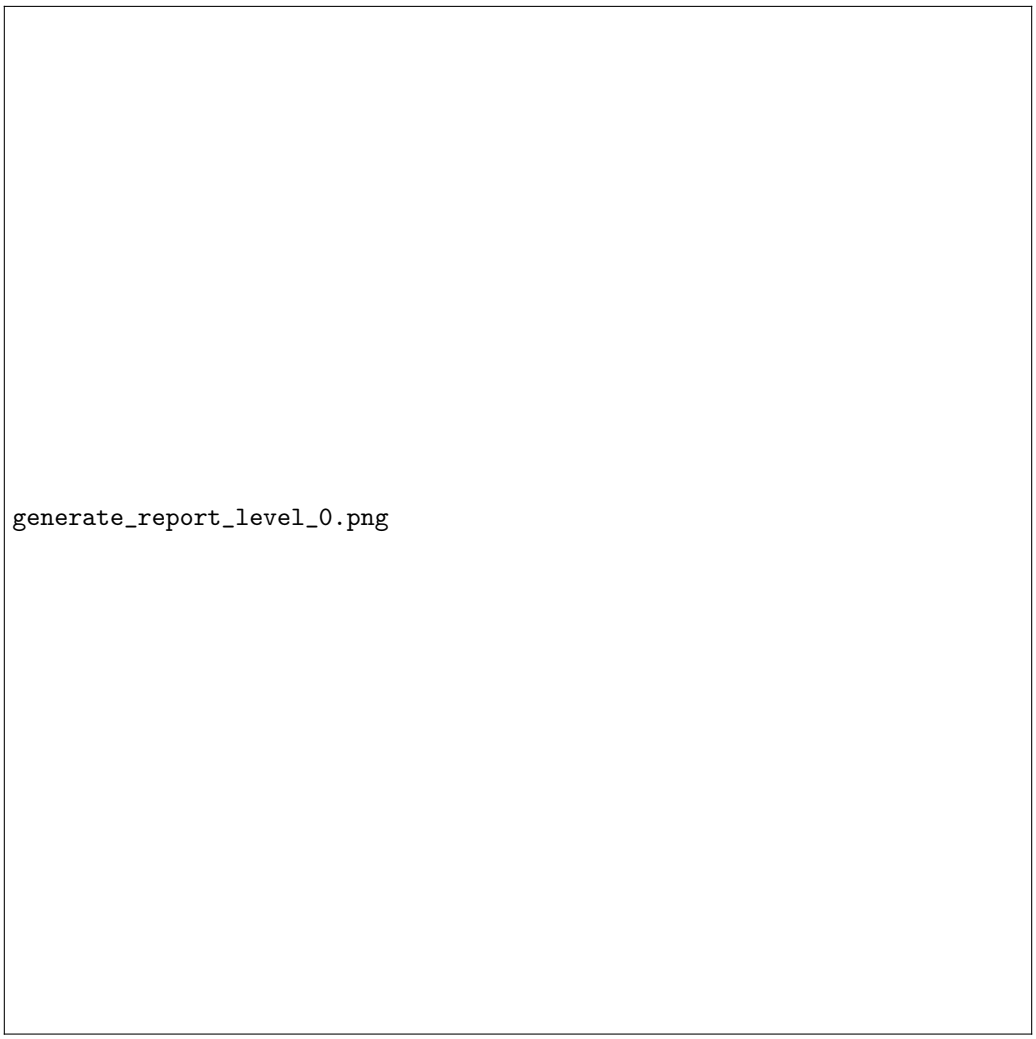

Figure 37: Level 0 Interaction Diagram for Generate Report

4.10.2 Level 1 Interaction Diagram (Key Actions)

- **Step 1**: Administrator selects report type and parameters.
- **Step 2**: System compiles report data.
- **Step 3**: System generates and displays the report.

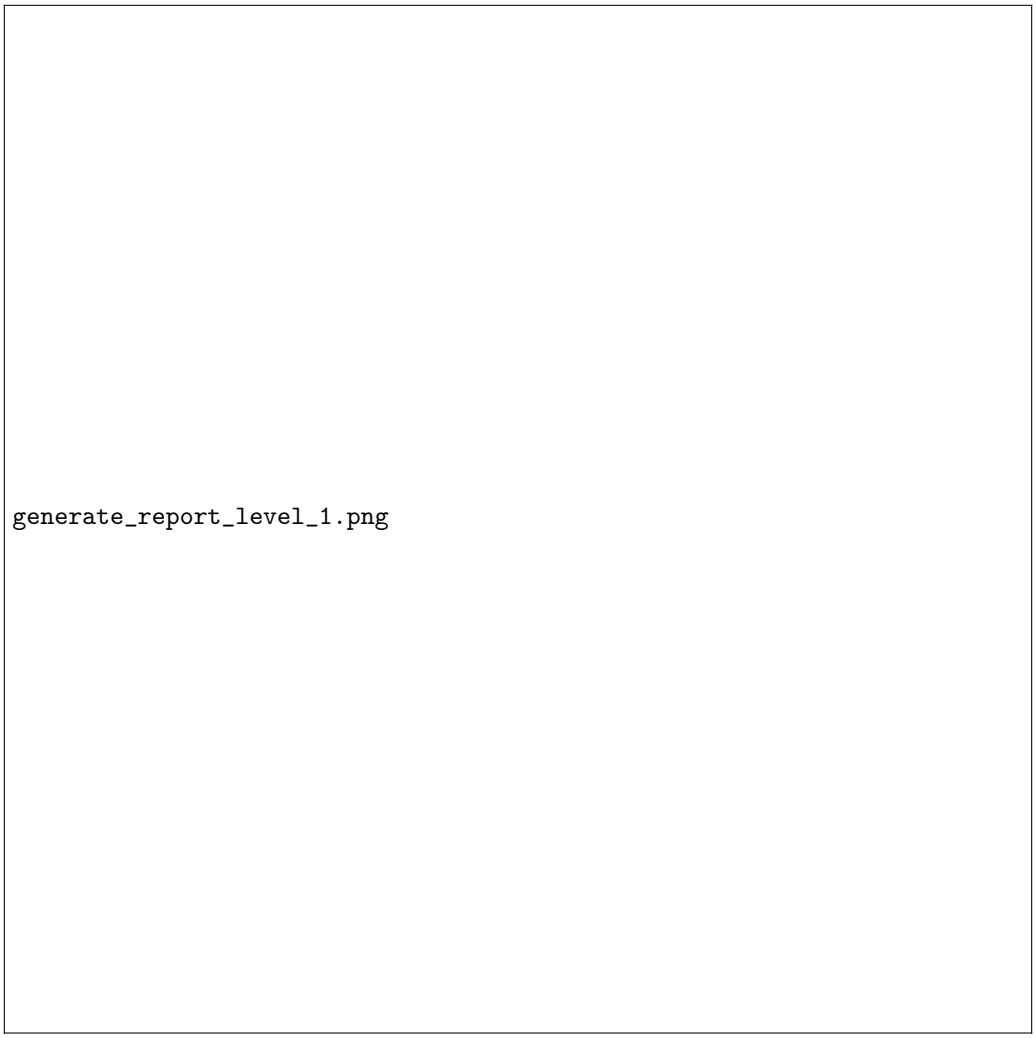

Figure 38: Level 1 Interaction Diagram for Generate Report

4.10.3 Level 2 Interaction Diagram (Detailed System Processes)

- **Step 1**: Administrator logs in and selects "Generate Report."
- **Step 2**: System prompts for report type and parameters (e.g., date range).
- **Step 3**: Administrator inputs parameters.
- **Step 4**: System retrieves data from the Database based on input parameters.
- **Step 5**: System compiles data and generates the report.
- **Step 6**: System displays or exports the report as requested.

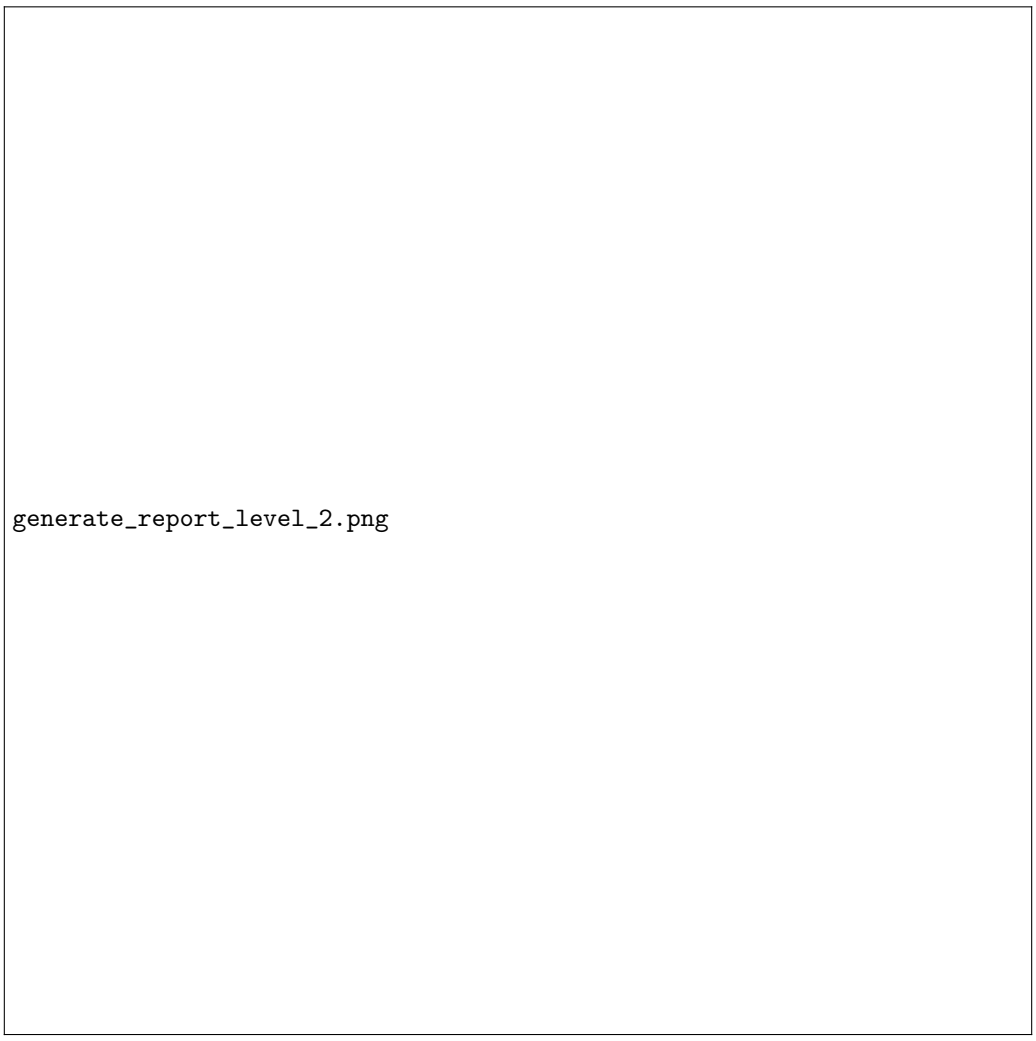

Figure 39: Level 2 Interaction Diagram for Generate Report

4.11 Send Notification

4.11.1 Level 0 Interaction Diagram (High-Level Overview)

- **Actors Involved**: Administrator
- **Action**: Administrator sends a notification.
- **System Interaction**: System delivers the notification.

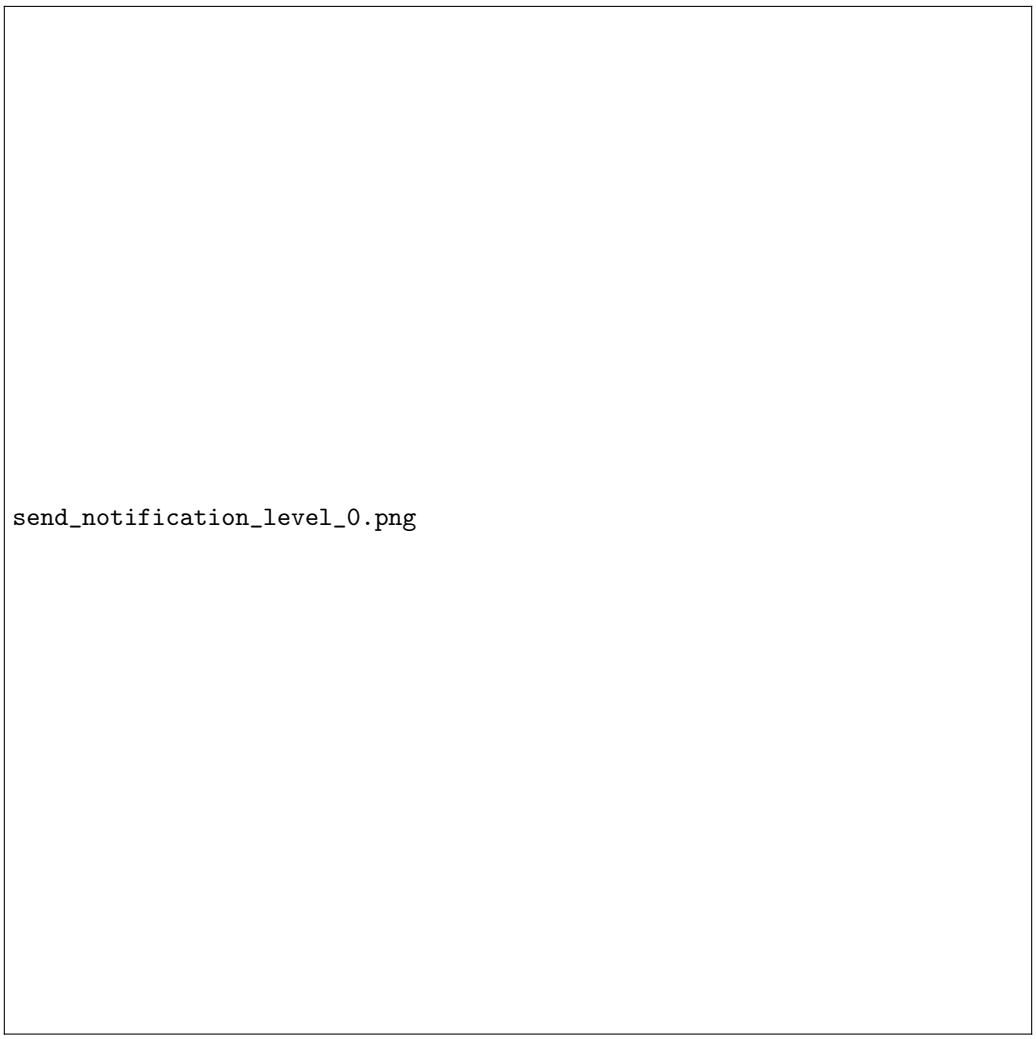

Figure 40: Level 0 Interaction Diagram for Send Notification

4.11.2 Level 1 Interaction Diagram (Key Actions)

- **Step 1**: Administrator selects recipients for the notification.
- **Step 2**: Administrator drafts the notification message.
- **Step 3**: System sends the notification.

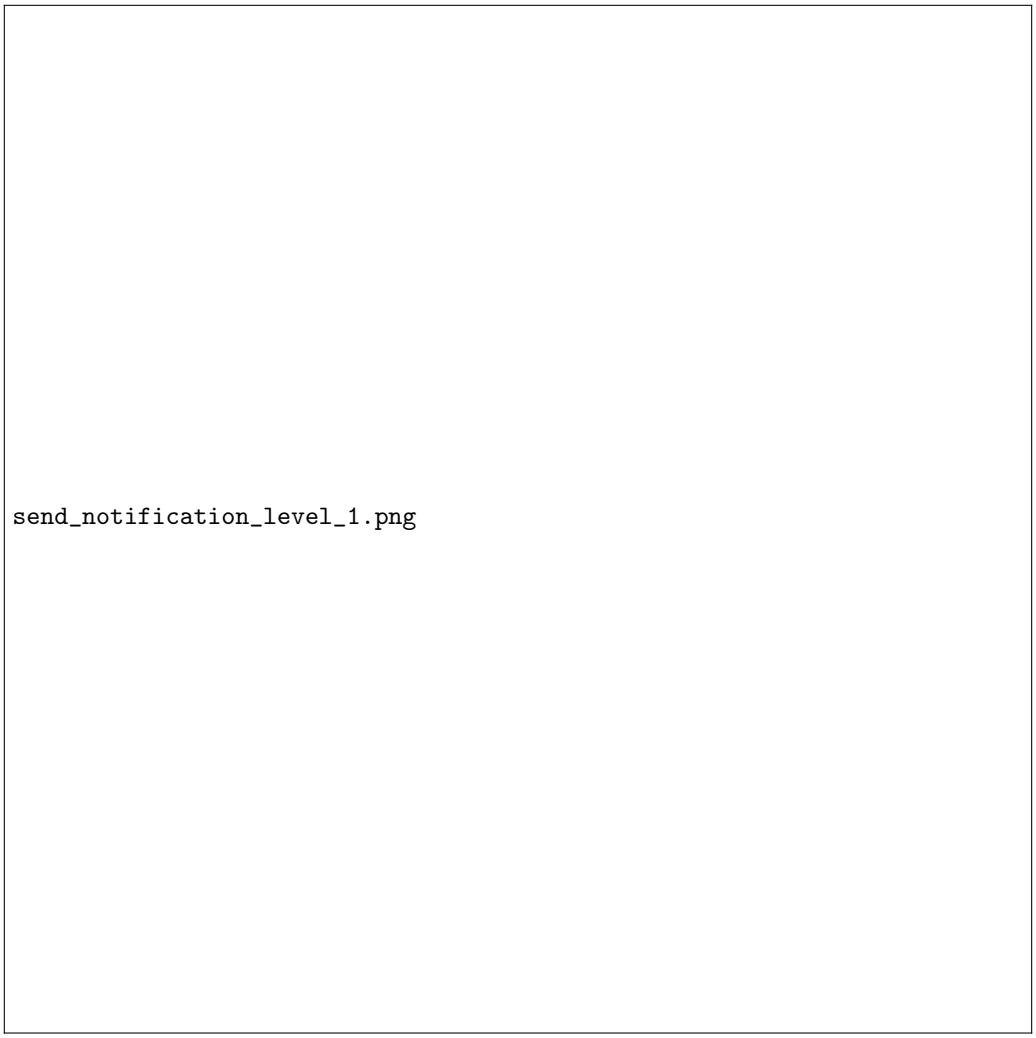

Figure 41: Level 1 Interaction Diagram for Send Notification

4.11.3 Level 2 Interaction Diagram (Detailed System Processes)

- **Step 1**: Administrator logs in and selects "Send Notification."
- **Step 2**: Administrator selects recipients (patients or doctors).
- **Step 3**: Administrator drafts and confirms the message.
- **Step 4**: System interacts with the Notification Service to send emails and SMS.
- **Step 5**: Notification Service delivers the notification to selected recipients.
- **Step 6**: System confirms notification delivery to the administrator.

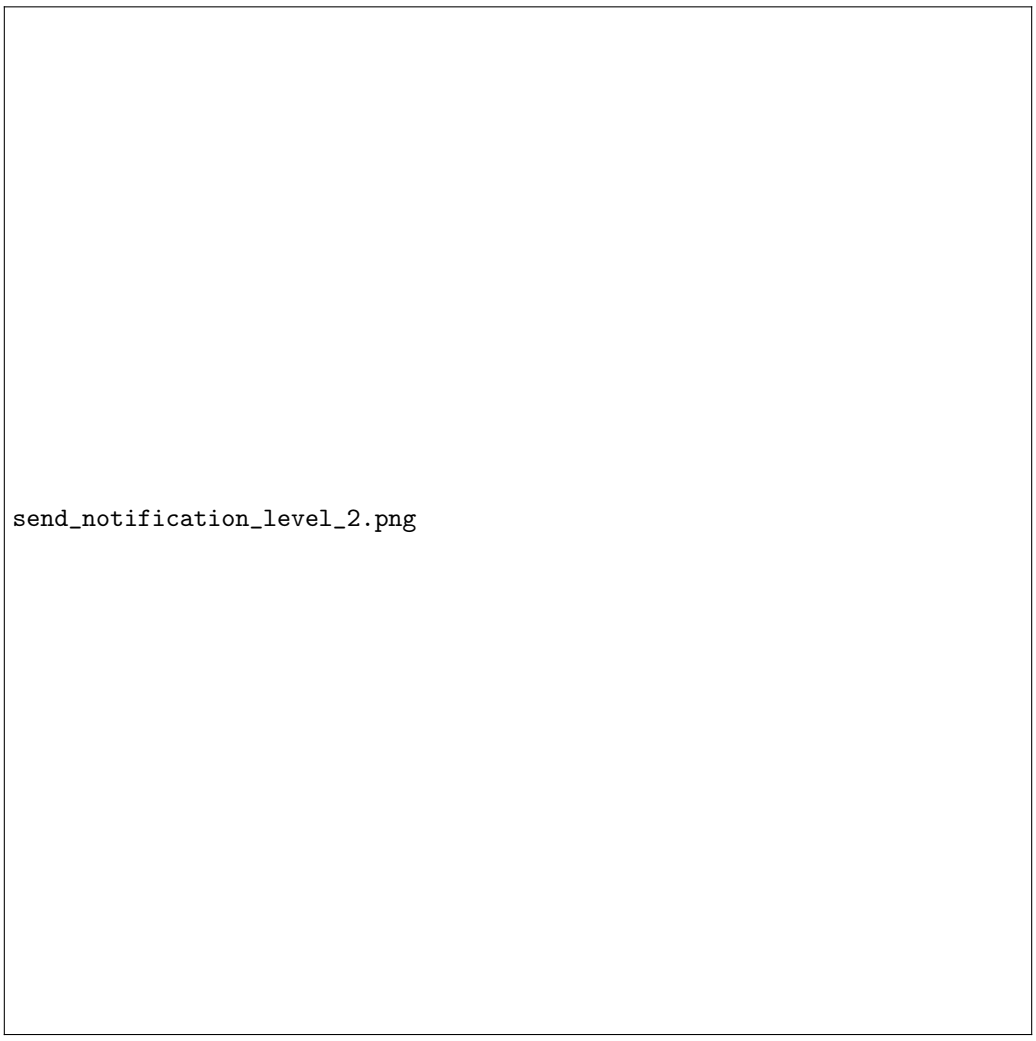

Figure 42: Level 2 Interaction Diagram for Send Notification

5 Class Diagrams for Each Use Case

5.1 Overview

shows the classes, their attributes, methods, and relationships.

5.2 Register Patient

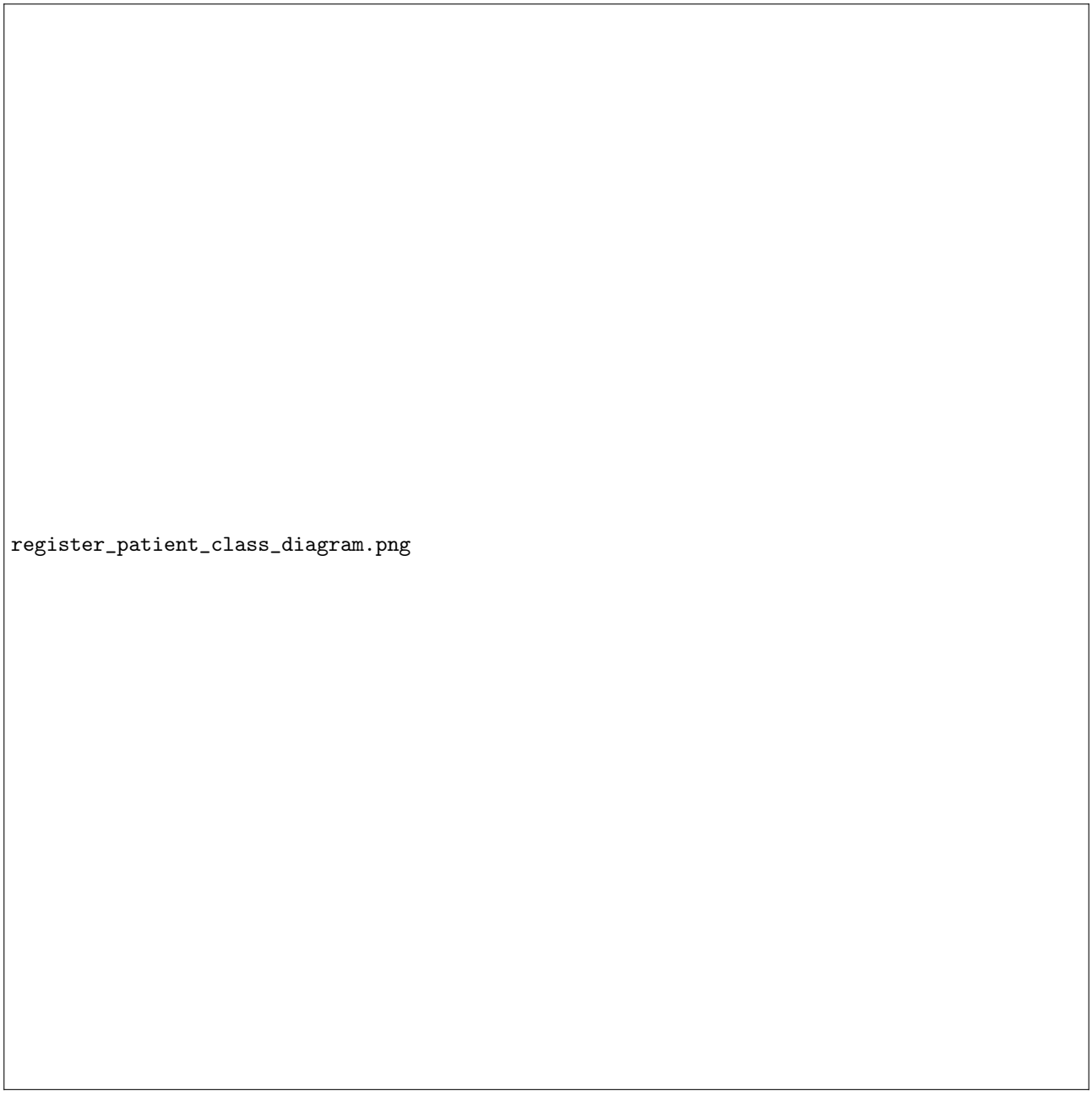

Figure 43: Class Diagram for Register Patient Use Case

5.3 Login User

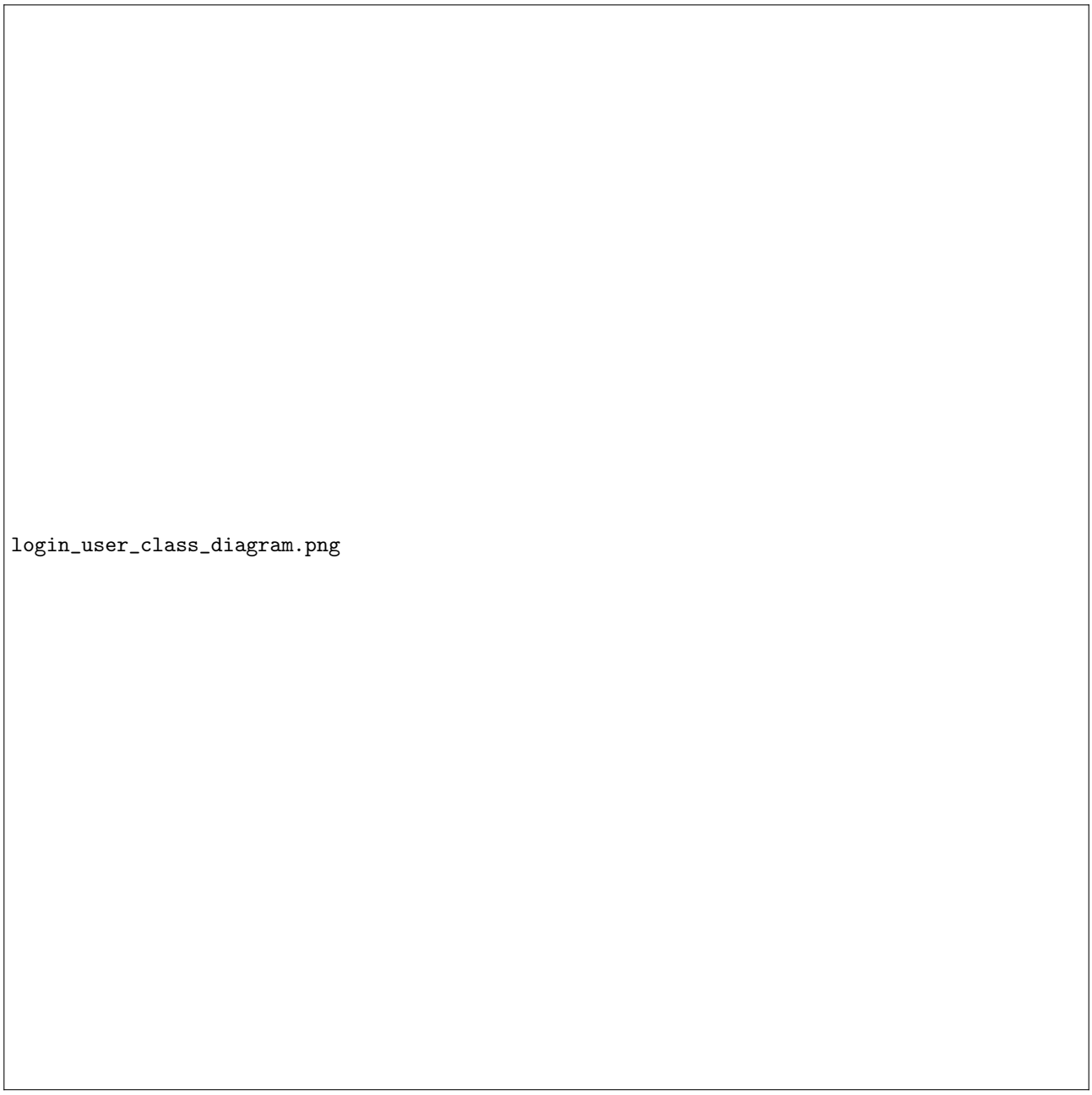

Figure 44: Class Diagram for Login User Use Case

5.4 Schedule Appointment

schedule_appointment_class_diagram.png

Figure 45: Class Diagram for Schedule Appointment Use Case

5.5 View Medical Records

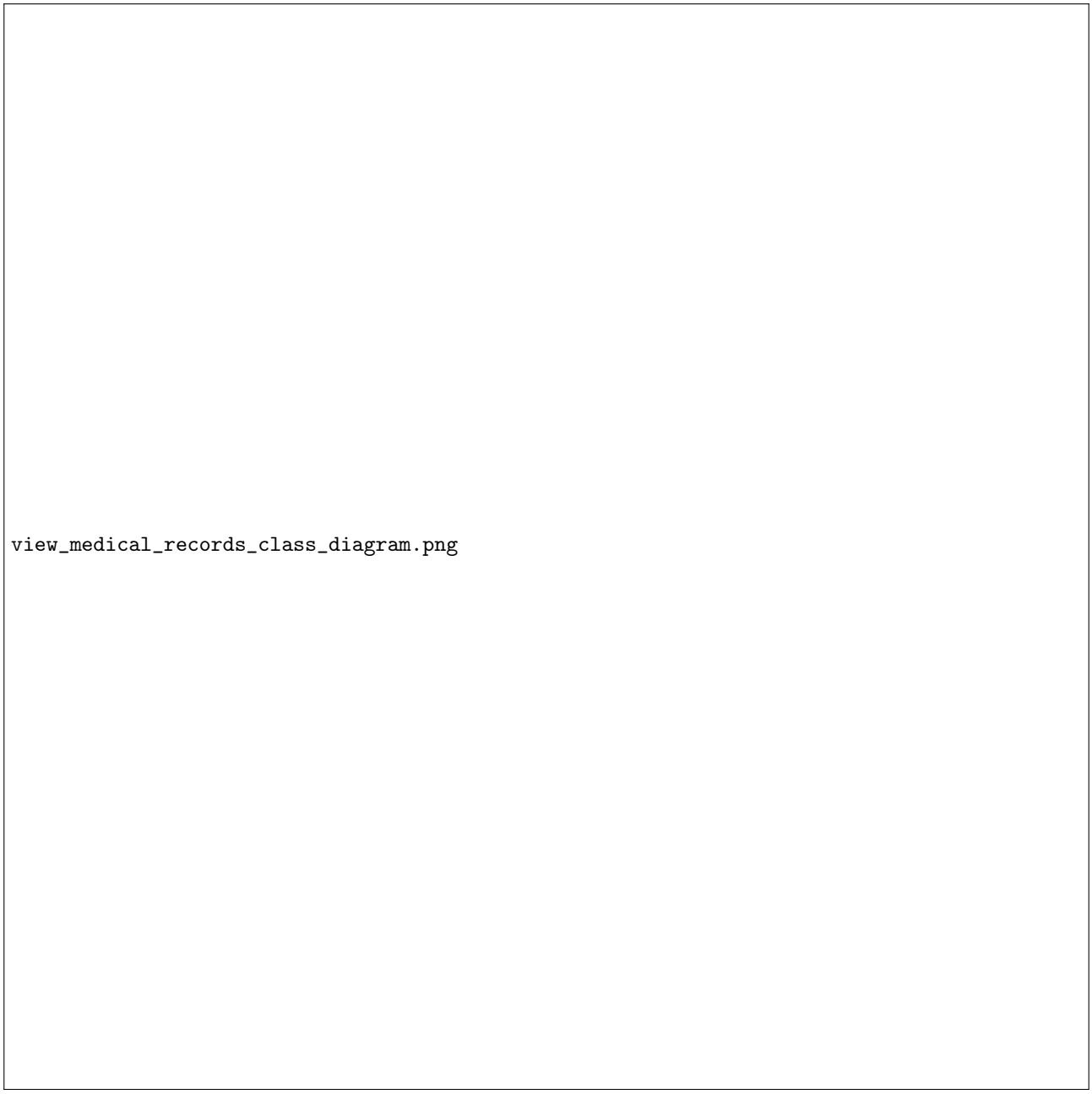

Figure 46: Class Diagram for View Medical Records Use Case

5.6 Update Medical Records

update_medical_records_class_diagram.png

Figure 47: Class Diagram for Update Medical Records Use Case

5.7 Process Payment

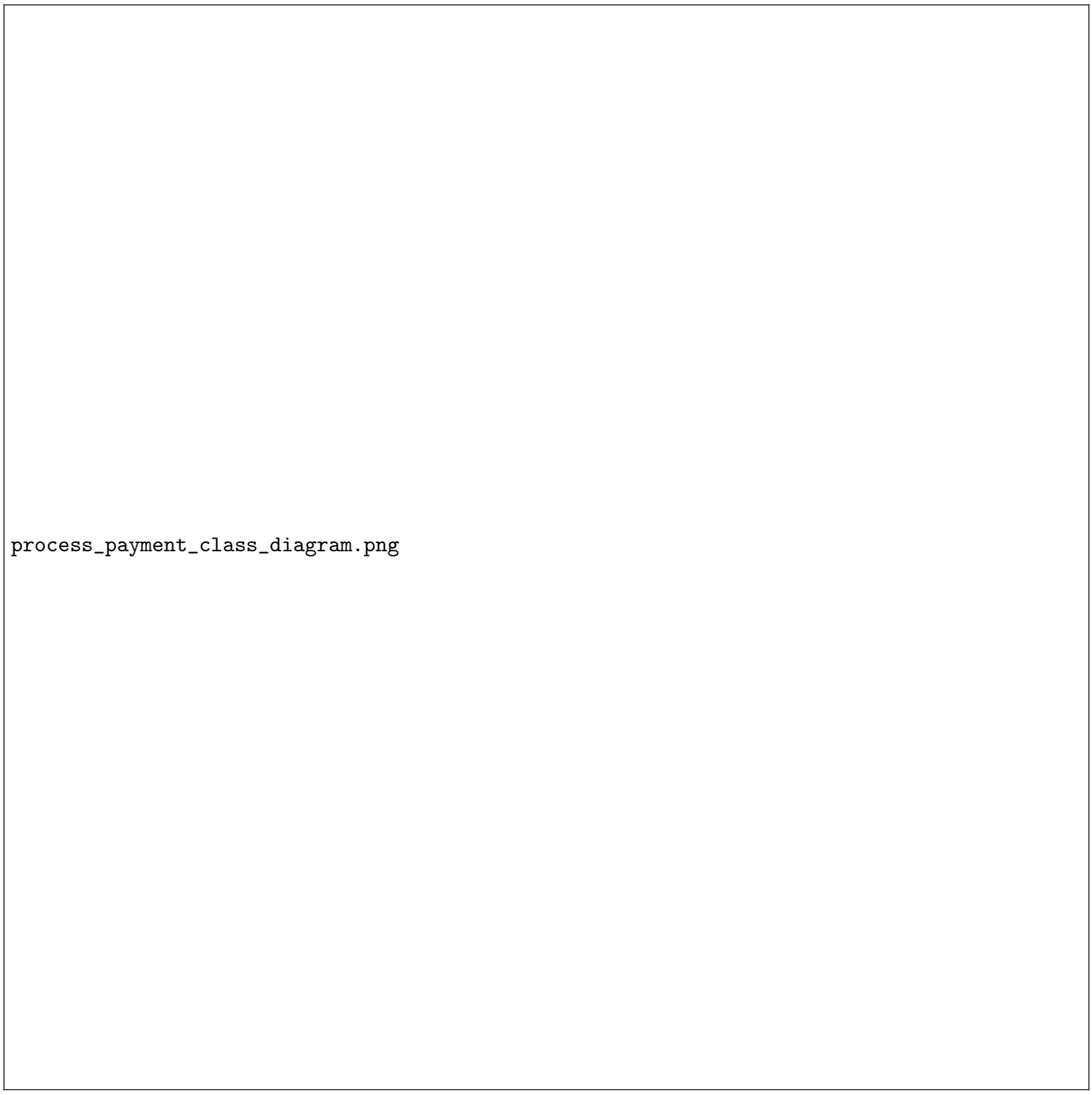

Figure 48: Class Diagram for Process Payment Use Case

5.8 Manage Staff

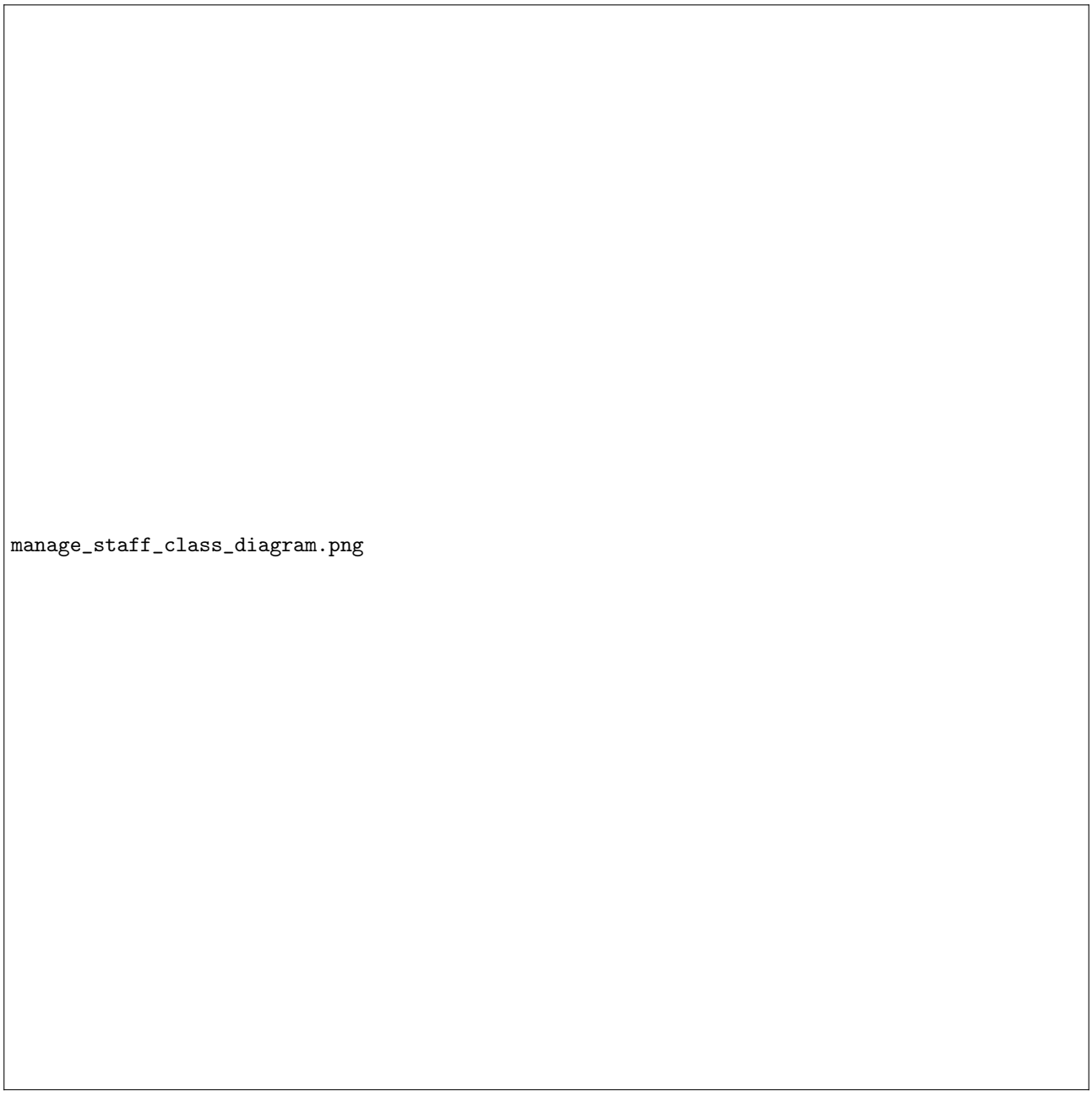

Figure 49: Class Diagram for Manage Staff Use Case

5.9 Manage Hospital Resources

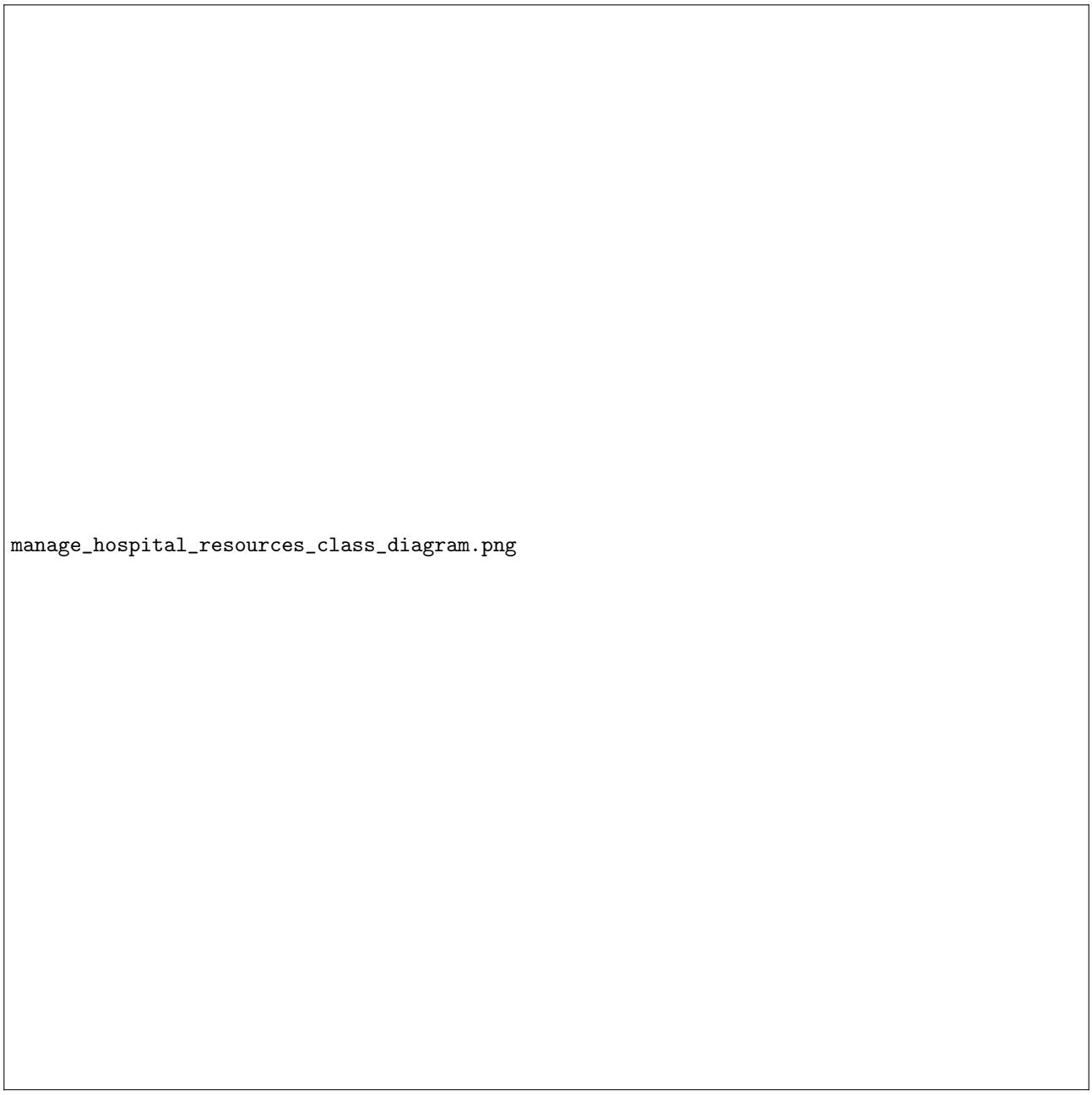

Figure 50: Class Diagram for Manage Hospital Resources Use Case

5.10 Generate Report

generate_report_class_diagram.png

Figure 51: Class Diagram for Generate Report Use Case

5.11 Send Notification

send_notification_class_diagram.png

Figure 52: Class Diagram for Send Notification Use Case

6 Package Diagram

6.1 Overview

groups classes into logical packages.

6.2 Package Diagram

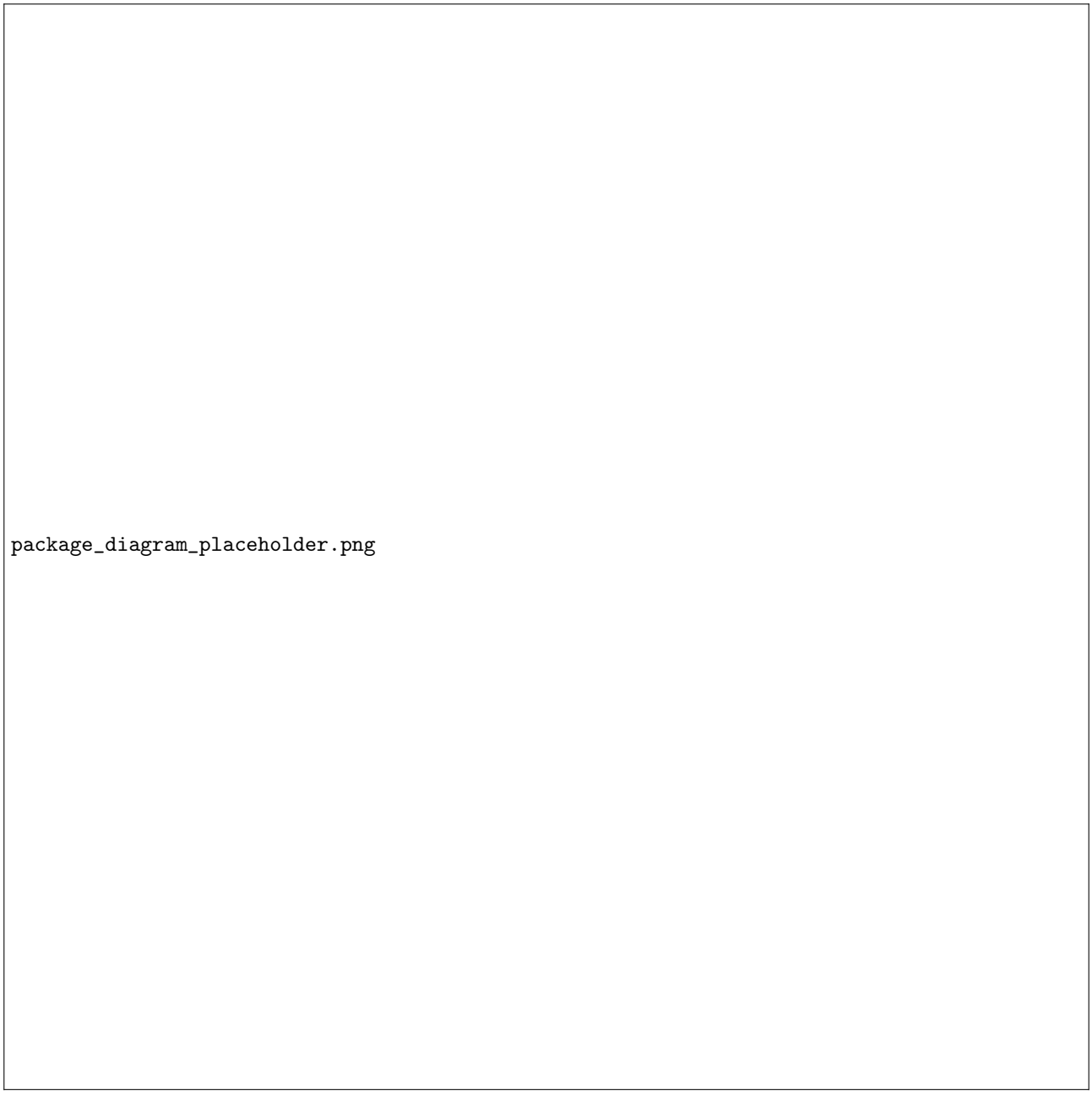

Figure 53: Package Diagram for Hospital Management System

6.3 Description of Packages

- **User Management Package**: Handles user registration, login, and profile management for all actors.
- **Appointment Management Package**: Manages all activities related to appointment scheduling, updating, and confirmation.

- **Billing Package**: Handles billing and payment processing, including integration with external payment gateways.
- **Medical Record Package**: Manages access to and updates of electronic medical records.
- **Notification Package**: Handles notifications (such as reminders and alerts) for patients, doctors, and administrators.
- **Resource Management Package**: Manages hospital resources such as rooms, equipment, and consumables.
- **Staff Management Package**: Handles hospital staff information, including adding, updating, and removing staff members.

7 CRC Cards for Each Class

7.1 Overview

Identify the roles and interactions of each class in the system.

7.2 User

- **Responsibilities**:
  - Store common user attributes (username, password).
  - Provide login functionality for all users.
- **Collaborators**:
  - SessionManager: Manages active sessions for logged-in users.

7.3 Patient

- **Responsibilities**:
  - Register for a patient account.
  - Schedule and manage appointments.
  - View medical records.
- **Collaborators**:
  - Appointment: Schedule and manage patient appointments.
  - MedicalRecord: Access patient's medical records.
  - Billing: Manage billing and payment for services.
  - NotificationService: Receive reminders and alerts.

7.4 Doctor

- **Responsibilities**:
  - Login to the system.
  - View patient records.
  - Update patient treatment details.
  - Manage appointment schedules.
- **Collaborators**:
  - MedicalRecord: View and update patient records.
  - Appointment: Manage appointments with patients.
  - NotificationService: Receive alerts for new appointments.

7.5 Administrator

- **Responsibilities**:
  - Login to the system.
  - Manage hospital staff.
  - Manage hospital resources.
  - Generate management reports.

  - Process patient billing.
- **Collaborators**:
  - Staff: Add, update, or remove staff information.
  - Resource: Manage hospital resources.
  - Billing: Process billing information.
  - Report: Generate various management reports.

7.6 Appointment

- **Responsibilities**:
  - Store appointment details.
  - Confirm or modify appointment schedules.
- **Collaborators**:
  - Patient: Schedule appointments.
  - Doctor: Assign appointments.
  - NotificationService: Notify patients and doctors about upcoming appointments.

7.7 MedicalRecord

- **Responsibilities**:
  - Store patient medical information (diagnosis, treatment, prescriptions).
  - Allow updates to patient records.
- **Collaborators**:
  - Patient: Provide read-only access to their own records.
  - Doctor: Update patient treatment information.

7.8 Billing

- **Responsibilities**:
  - Generate billing information for services.
  - Track payments.
- **Collaborators**:
  - Patient: Generate bills for services provided.
  - Payment: Process payments.
  - Administrator: Oversee billing processes.

7.9 NotificationService

- **Responsibilities**:
  - Send notifications (e.g., reminders, alerts).
  - Log notifications sent.
- **Collaborators**:
  - Patient, Doctor, Administrator: Send notifications related to appointments, updates, or alerts.

7.10 Staff

- **Responsibilities**:
  - Store details of hospital staff (e.g., nurses, technicians).
- **Collaborators**:
  - Administrator: Add, update, or remove staff information.

7.11 Resource

- **Responsibilities**:
  - Manage hospital resources (e.g., equipment, rooms).
  - Track resource availability.
- **Collaborators**:
  - Administrator: Manage and update resource details.

8 Application of SOLID Principles

8.1 Overview

SOLID principles guide the design of object-oriented software. It mostly used cause it helps with modularity, maintainability, and scalability.

8.2 Single Responsibility Principle (SRP)

Each class in the system has a distinct, well-defined responsibility. In our case, the **User** class handles login functionality, while specialized classes like **Patient**, **Doctor**, and **Administrator** handle unique responsibilities.

8.3 Open/Closed Principle (OCP)

Classes are open for extension but closed for modification, allowing new features to be added without altering existing code. For instance, new notification channels can be added by extending the **NotificationService** without modifying its core functionality.

8.4 Liskov Substitution Principle (LSP)

Subclasses can be substituted for their base classes without affecting the correctness of the application. The **Patient**, **Doctor**, and **Administrator** classes all inherit from **User** and can be used in any context where **User** is expected.

8.5 Interface Segregation Principle (ISP)

Specific, small interfaces are used, ensuring that classes are not forced to implement methods they do not need. In our case, the **PaymentGateway** interface includes only methods required for payment processing, allowing different payment processors to implement it as needed.

8.6 Dependency Inversion Principle (DIP)

High-level modules do not depend on low-level implementations. Instead, both depend on abstractions, making the system more flexible and easier to maintain. The **Payment** class depends on the **PaymentGateway** interface, rather than a specific payment processor implementation.

9 Conclusion

9.1 Summary of the Design Approach

The hospital management system has been designed to follow the SOLID principles as much as possible. But also for ease of use.

By doing this, the system is easier to maintain, extend, and understand

for example some future work and extensions could be:

- New user roles, easily
- Additional notification channels
- Intergration with third party services

A Appendix

A.1 Glossary

- **SRS**: Software Requirements Specification.
- **EMR**: Electronic Medical Record.