

# **Software Design Documentation**

**for**

## **EasyCare Hospital Management System**

**Version 2.0 approved**

**Prepared by Hassan Hisham, Nadder Ragi, Aziz Koura, Hla el Sakka,  
Yahia Diab, Youssef Ahmed, Mohamed Gamal**

**ESLSCA University**

**15 June 2023**

## **1. Introduction**

The software design document (SDD) serves as a blueprint for the development of the EasyCare hospital management system. This document outlines the design and architecture of the software, detailing its various components and functionalities. The purpose of the SDD is to provide a comprehensive understanding of how the software will be implemented and how it will meet the requirements specified in the Software Requirement Specifications(SRS).

## **2. Testing Scenarios**

Testing scenarios help ensure that the software system, as described in the Software Design Document (SDD), functions correctly and meets the specified requirements. Here are some testing scenarios you can consider for your SDD:

### **1. Unit Testing:**

- Test each individual component(classes, functions,etc...) to verify their correctness and functionality
- Validate the inputs, outputs and behavior of each component under different test cases
- Check for proper handling of error conditions and expectations.

### **2. Integration testing:**

- To ensure they function as intended, test the integration of various software components.
- Check the integrated components' compatibility, communication, and data flow.
- Test different integration points and interfaces, such as messaging, data transfers, and API requests.

### **3. System testing:**

- Test the entire system as a whole to validate its behavior and performance.
- Check that the various system features and use cases work as intended.
- Test edge cases, boundary conditions, and usage scenarios that are representative of the real world.

### **4. Performance testing:**

- Measure and evaluate the performance of the system under expected load conditions
- Test response times throughput, resource usage and scalability.

### **5. Security testing:**

- Test the system for any security risks
- Validate security controls

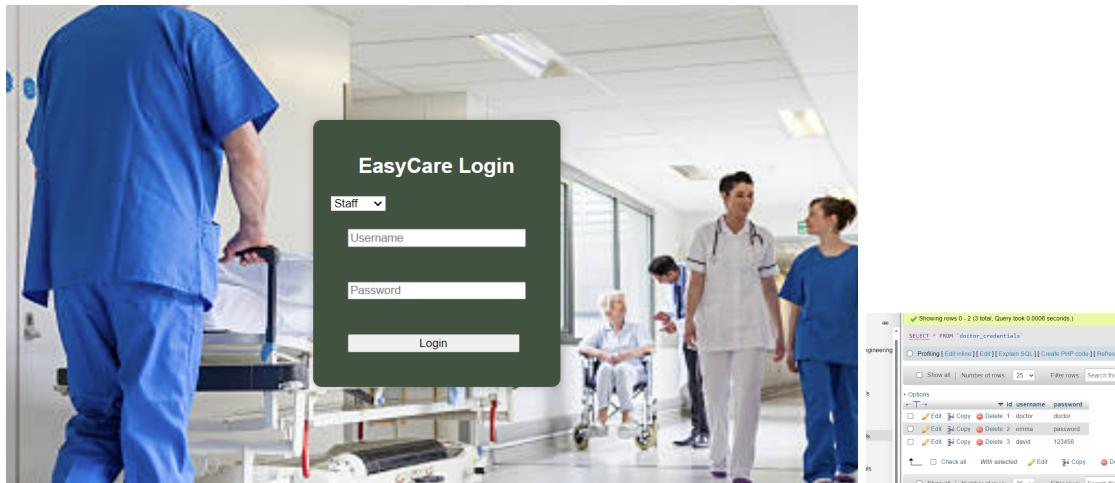
## 2.1 Function Testing

- **Login According to Usertype and Database Credentials**

The login selector is linked to the database's usertype to determine the login type.

According to that login type, the username and password are validated against the set in the database.

**Status: Fully functioning**



- **Patient CRUD**

Composed of two pages, this is a table interface with editing, deleting, and adding of patients to the backend of the website. Validated using various trials at the function.

**Status: Fully functioning**

**EasyCare Hospital Admin / Staff Patient Board**

Add New

UserID	Medical History	CurrentStatus	Allergies	AdmissionDate	
1	none	enrolled	sugar	2023-06-13	<a href="#">Edit</a>   <a href="#">Delete</a>
0	asfas	safasf	asfasf	2313-12-12	<a href="#">Edit</a>   <a href="#">Delete</a>

**Add Patient**

ID

UserID

Medical History

Current Status

Allergies

Insurance Details

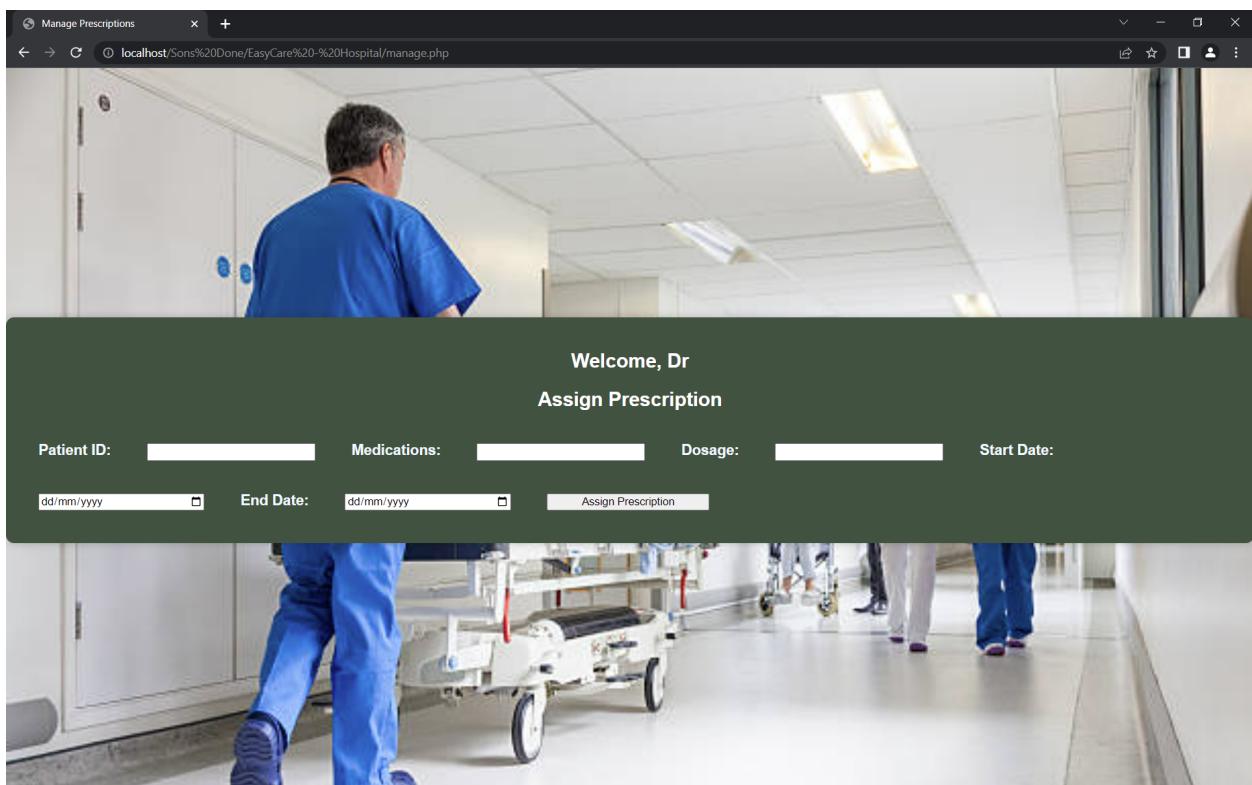
Admission Date

[Go Back](#)

- **Doctor Prescription Adding**

Through various input fields, the doctor adds a prescription to the table in the database and is sent to the table directly, and a short message is sent after.

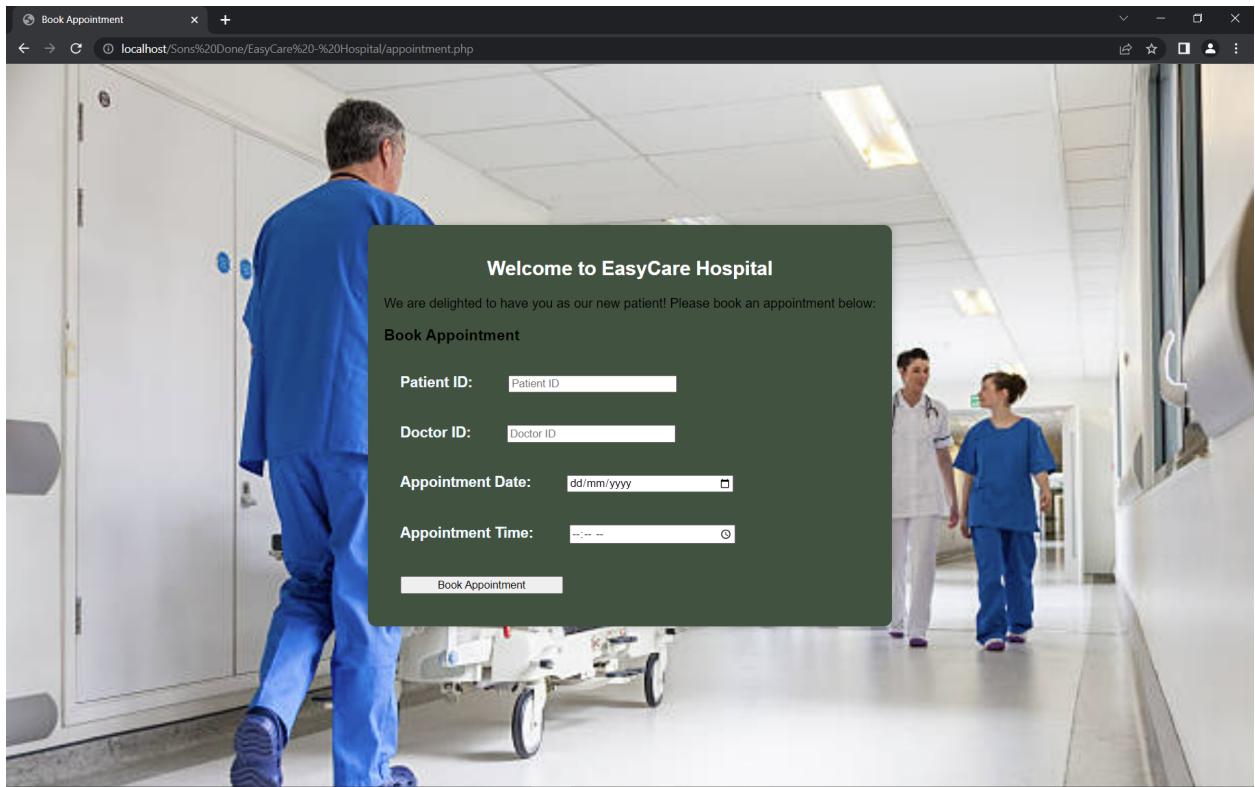
**Status: Fully functioning**



- **Patient Appointment Booking**

By entering the appointment details and their chosen doctor, an appointment is booked and sent directly to the database.

**Status: Fully functioning**



### 3. System overview

The EasyCare hospital management system is a comprehensive software solution designed to streamline and digitalize day-to-day operations in hospitals. It aims to enhance the efficiency and effectiveness of hospital processes, including patient management, appointment scheduling, medical record keeping, employee management, room bookings, billing and laboratory operations.

The system provides a standalone application that serves as a centralized platform for managing various hospital functions. It replaces the manual paper-based system, reducing the risk of data loss and improving organization and accessibility of information. The application is designed to be user-friendly, catering to different user classes such as staff, doctors and patients.

## **Key features:**

1. Appointment management: patients can book appointments with doctors, and doctors can track and manage their schedules. The system provides a convenient way to schedule appointments without the need for manual coordinations
2. Patient management: the system maintains comprehensive patient records, including personal details, medical history and insurance information
3. Medical record keeping: patients can view their medical records stored in the system, eliminating the need for manual file retrieval. The system ensures the confidentiality and security of patient information during record viewing
4. Billing and Payment: The system enables the generation of patient invoices and allows doctors and staff to update bill details. Patients can view their bills, make payments, and receive detailed information about the charges. This feature simplifies the billing process and enhances transparency for patients.
5. Room Booking: Staff members can book rooms for patients who require overnight stays.

The system provides an easy-to-use interface for managing room availability and allocations, ensuring efficient utilization of hospital resources.

6. Employee Management: The system maintains comprehensive employee records, including details about roles, employment status, schedules, and departments. It allows administrators to manage employee information and facilitates effective workforce Management.

7. Reporting and Analytics: The system offers reporting capabilities, enabling the generation of financial reports and other analytics. This helps in monitoring and analyzing hospital performance, facilitating data-driven decision-making.

Overall, the EasyCare Hospital Management System aims to provide a secure, efficient, and user-friendly solution for hospitals to streamline their operations. By automating manual processes and centralizing data management, the system improves efficiency, accuracy, and accessibility of information, leading to enhanced patient care and operational effectiveness.

#### **4. Implementation**

The hospital system management software has been developed using a combination of PHP, HTML, and MySQL technologies. The implementation follows a three-tier architecture consisting of the presentation layer (HTML and PHP), application logic layer (PHP), and data storage layer (MySQL).

The integrated development environment (IDE) used in our software for writing and editing source code was primarily Visual Studio Code. It offers a large selection of features and extensions that boost productivity and enable fluid development. Additionally, the MySQL database was managed using phpMyAdmin, a database administration tool.

Using SQL queries run through phpMyAdmin, the necessary database tables are first created as part of the implementation process. The main tables are labeled "patients", "doctors" and "staff" for storing information about patients, staff and doctors, respectively. To guarantee data integrity, these tables are created with the proper attributes and constraints.

The user interface is built using HTML and PHP. The main page, app.php', serves as the entry point for the hospital management system, it makes you select

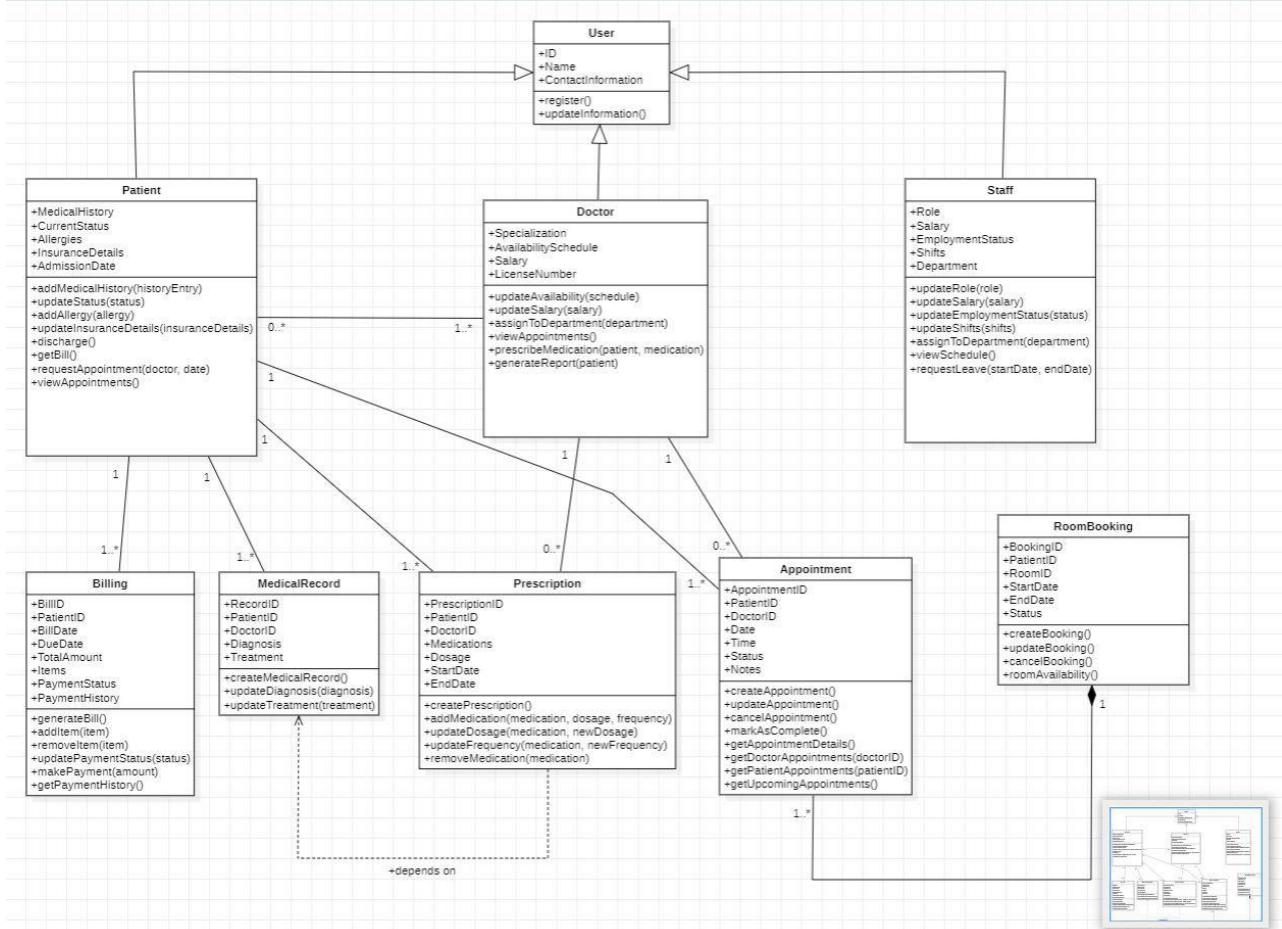
which user type you are and after login it redirects to the applicable page. The form submissions are handled by separate PHP scripts, 'add\_patient.php' and 'add\_doctor.php', which interact with the database to insert the data.

To establish a connection with the database, the PHP scripts utilize the `mysqli_connect()` function, passing in the necessary credentials such as the database hostname, username, password, and database name. If the connection is successful, the script proceeds with retrieving the form data from the submitted HTML forms.

The use of PHP, HTML, and MySQL have been very adjustable to create the hospital management system. It created an interactive and efficient solution for managing patient and doctor information. The development process is improved and the management of the system's source code and database is made simpler with the use of Visual Studio Code as the IDE and phpMyAdmin as the database administration tool.

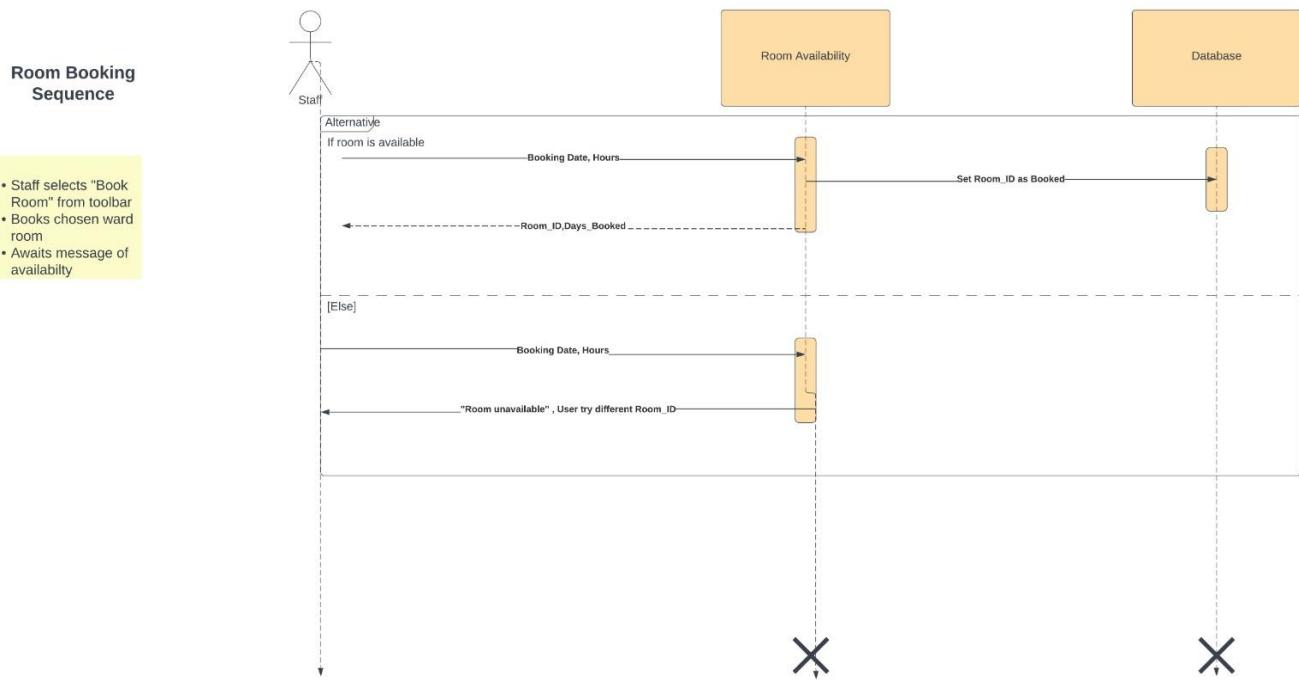
## 5. Diagrams

### 5.1 Class diagram



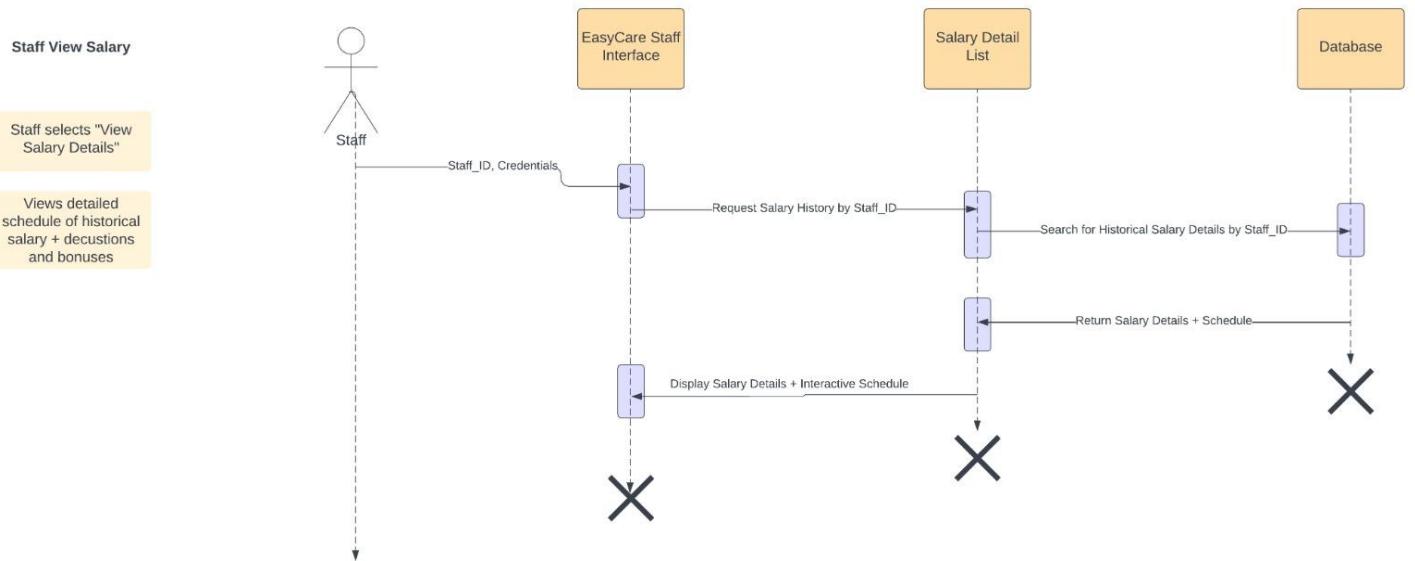
## 5.2 Sequence diagram

### 5.2.1 Booking a room



In this sequence diagram we can see the procedures it takes to book a room by a member of the staff.

## 5.2.2 Staff view salary

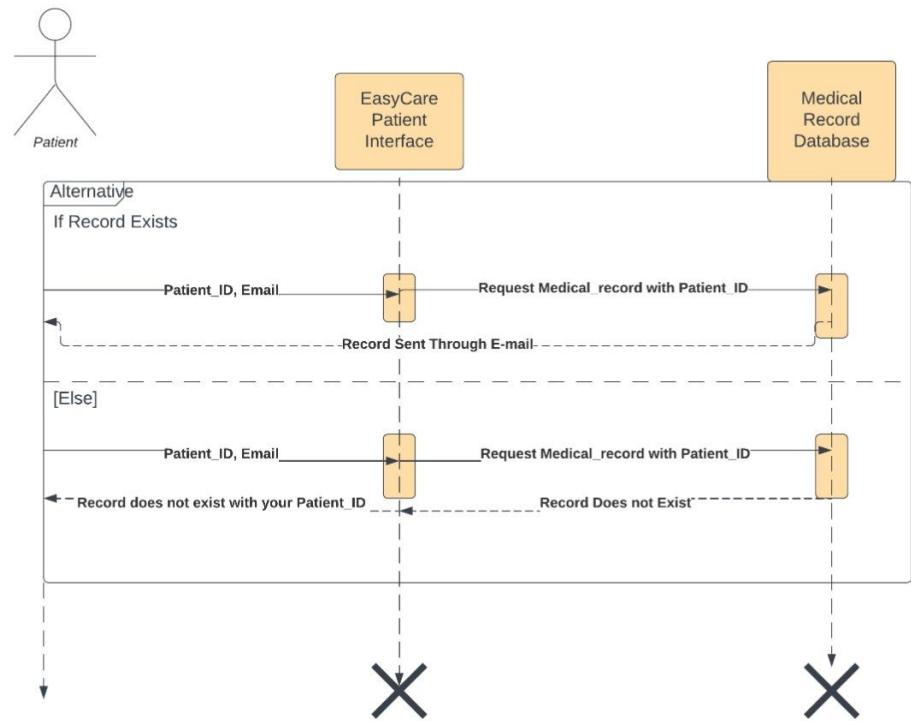


In this sequence diagram we can see the procedures it takes from members of the staff to access the view of salary.

### 5.2.3 Patient Medical Record Viewing

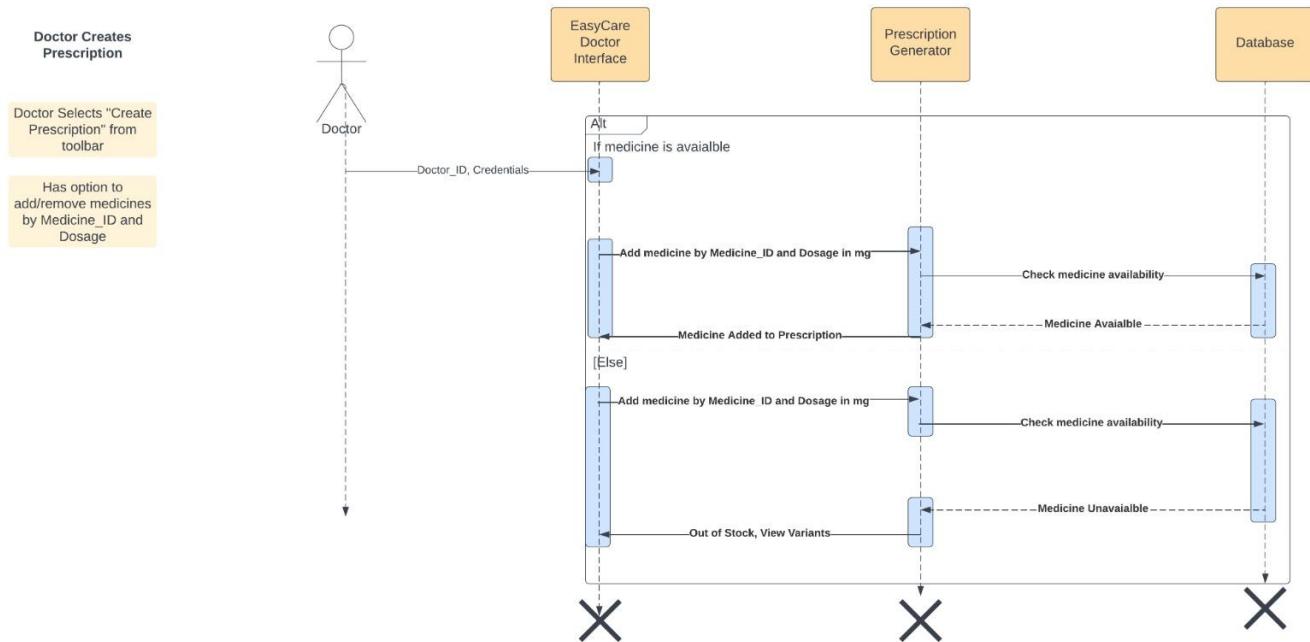
#### Patient Medical Record Viewing

- Patient selects "View Medical Record"
- Patient selects the method of delivery of medical record
- User receives medical record on e-mail



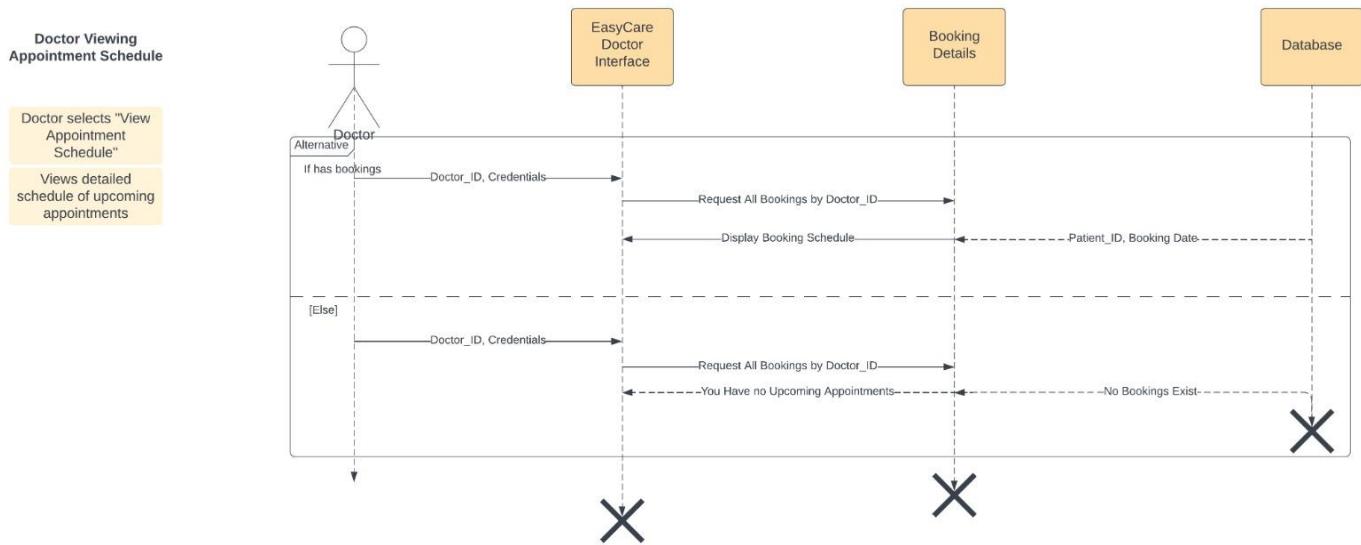
In this sequence diagram, we can see the procedures it takes for a patient to access his medical record.

## 5.2.4 Creating prescription



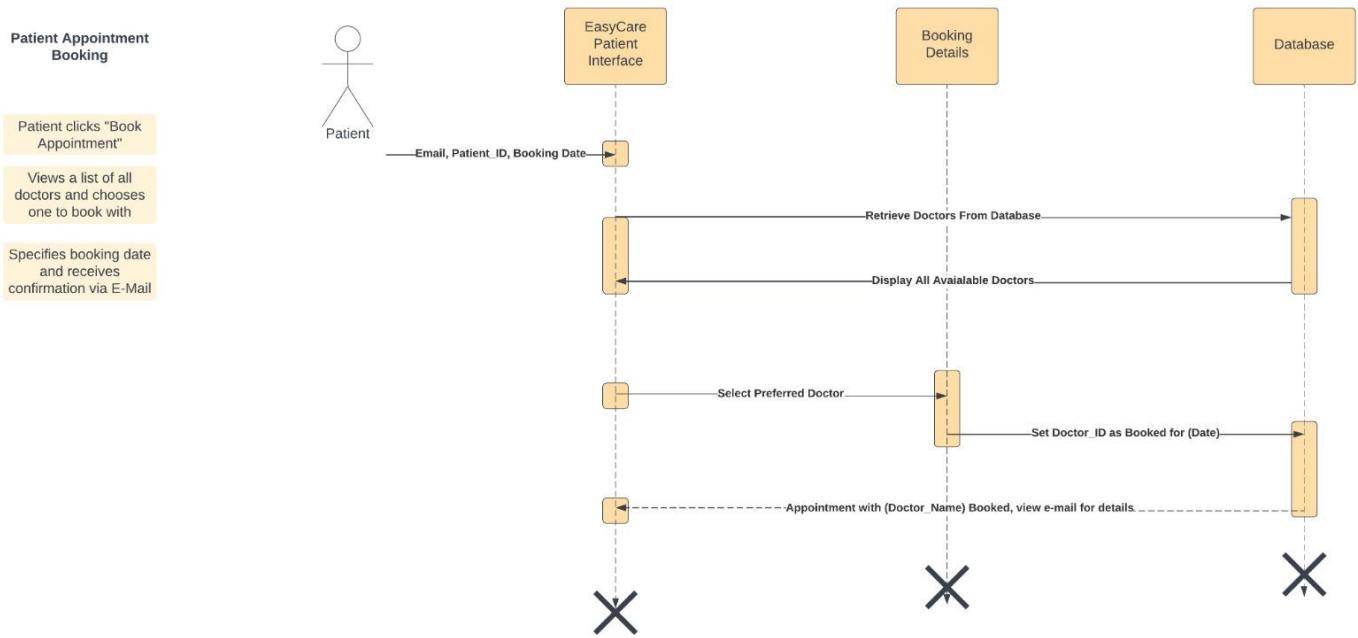
In this sequence diagram we can see the procedures it takes from a doctor to create a new Prescription.

## 5.2.5 Doctor viewing appointment schedule



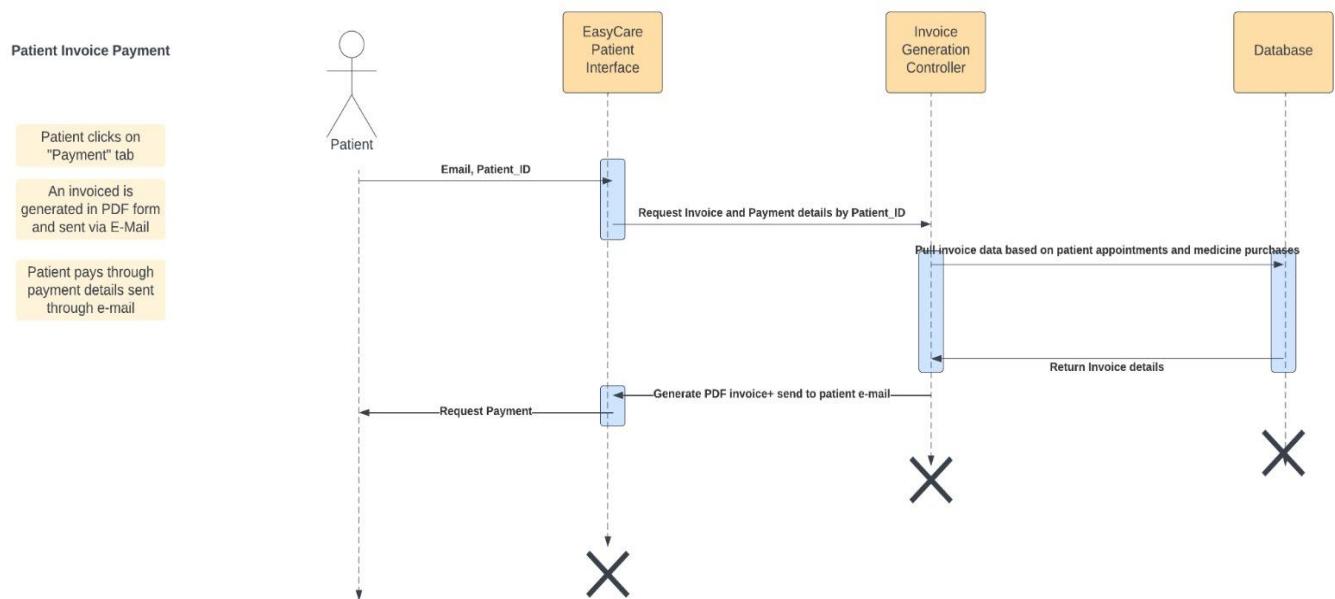
In this sequence diagram we can see the procedures it takes from a doctor to access viewing his appointment schedule.

## 5.2.6 Patient appointment booking



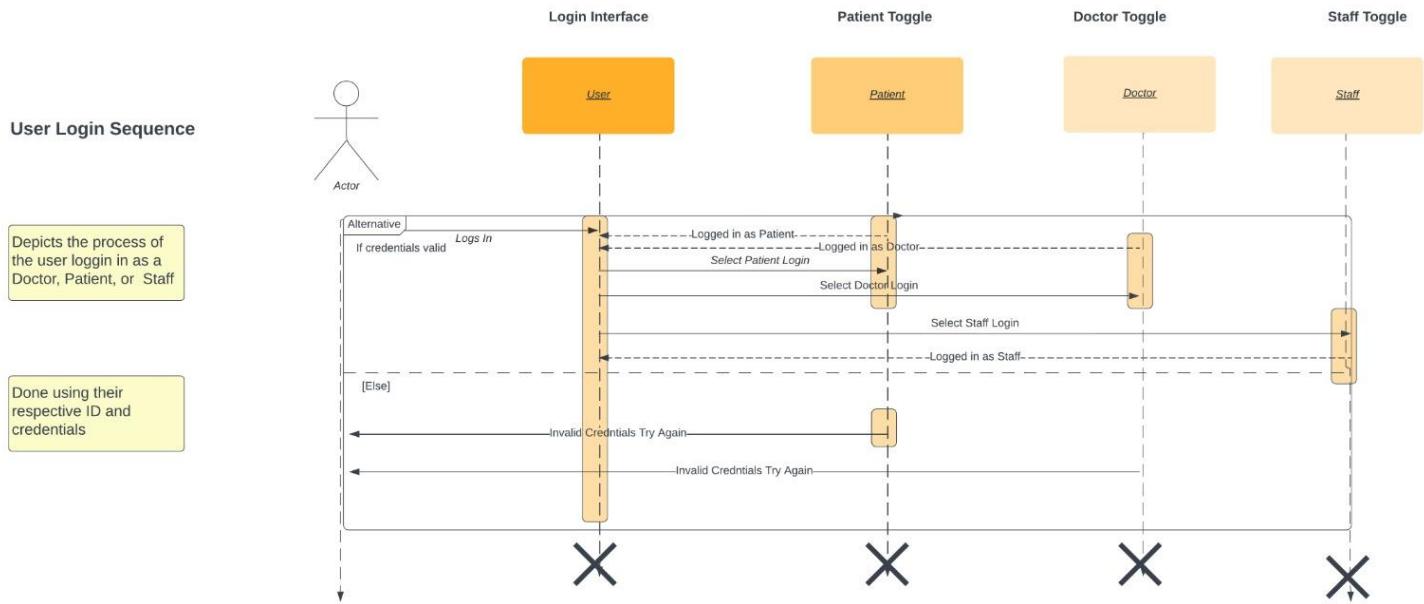
In this sequence diagram we can see the procedures it takes from a patient to book an appointment

### 5.2.7 Patient invoice payment



In this sequence diagram we can see the procedures it takes from a patient to pay his invoice payments

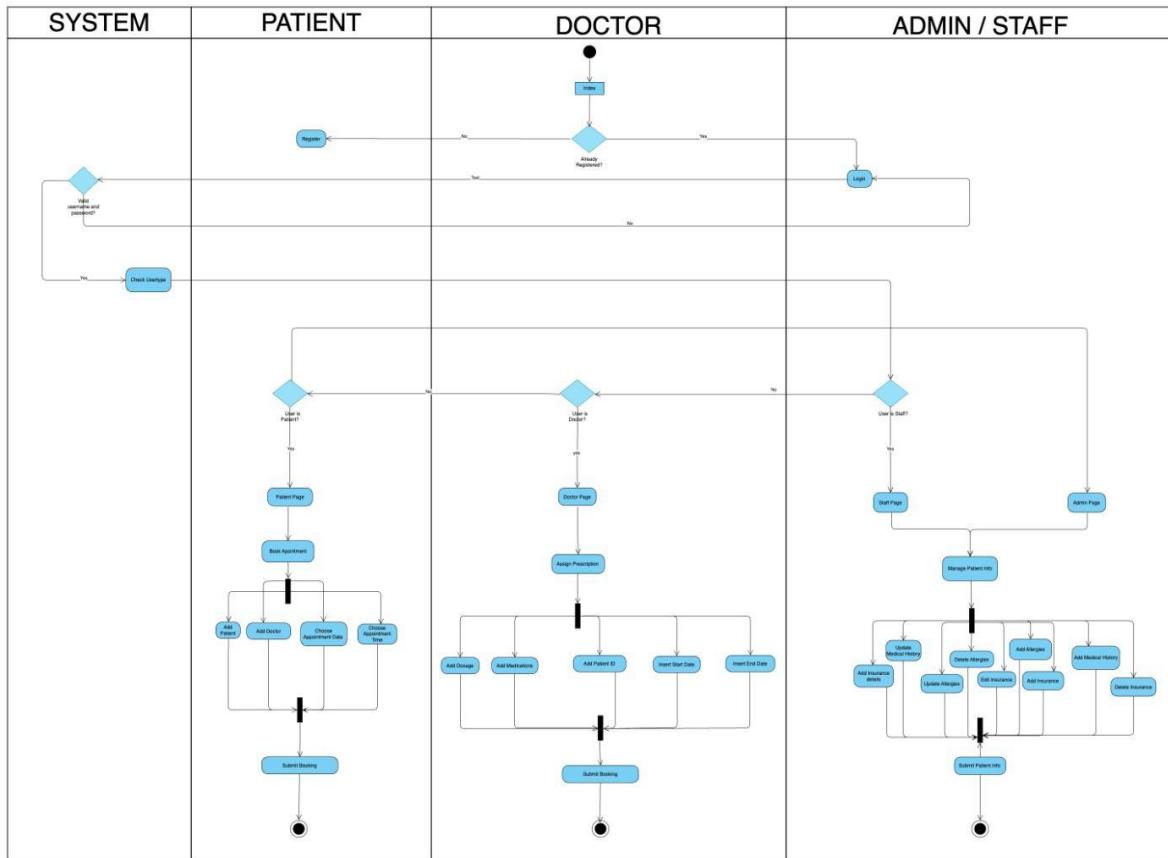
## 5.2.8 User login sequence



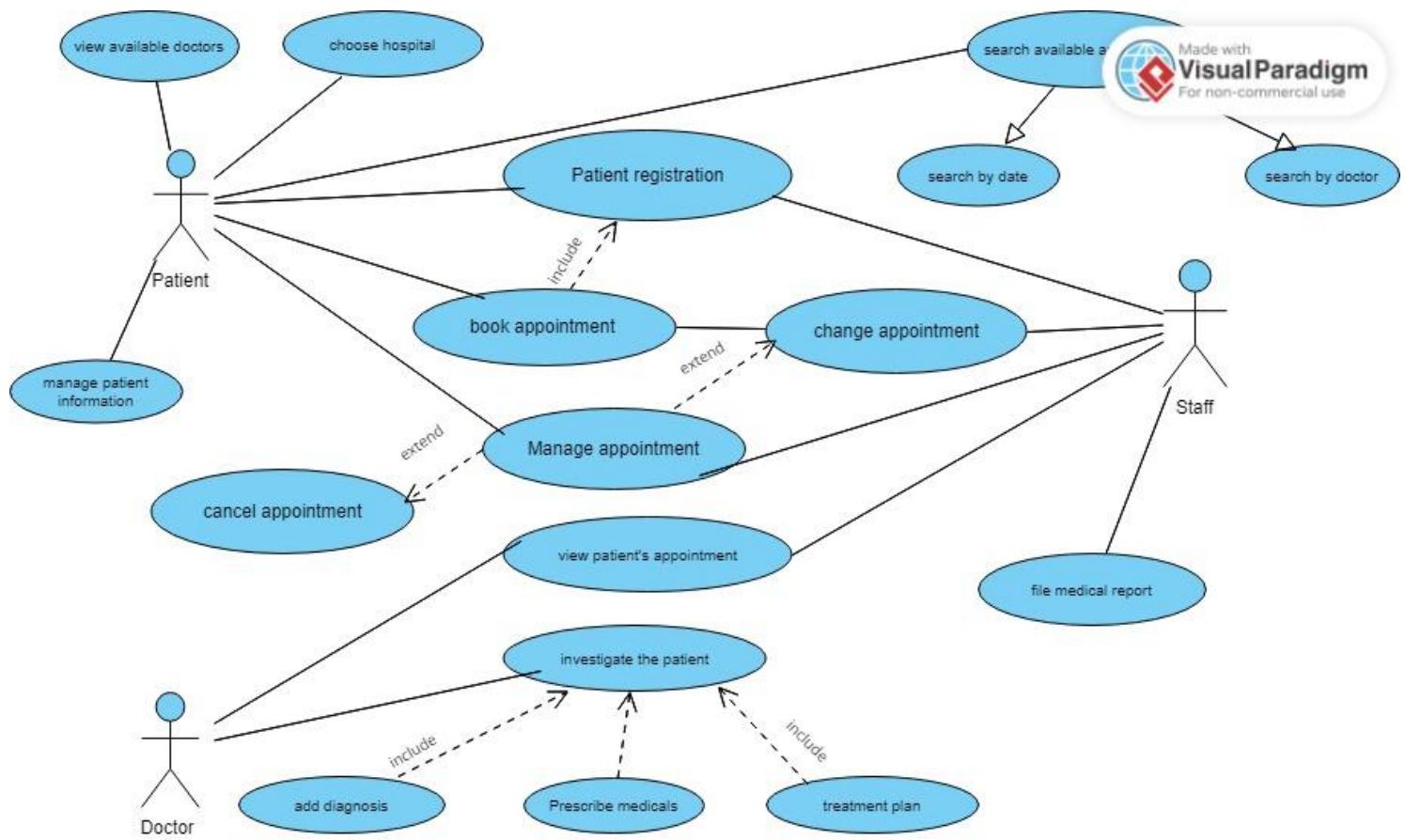
In this sequence diagram we can see the procedures it takes from a user to login.

## 5.3 Activity Diagram

Made with  
 VisualParadigm  
 For non-commercial use



### 5.3 Use case Diagram



## **6. Maintenance**

To ensure the SDD remains accurate and up to date throughout the software development lifecycle, proper maintenance has been scheduled. Here are some maintenance considerations for your SDD document:

1. Bug fixes: The software's development team tracks down and eliminates bugs to ensure a smooth operation.
2. Adding new features or enhancing existing ones in the program based on consumer demand can improve the user interface and overall performance.
3. Performance optimization involves examining and enhancing the software's functionality to make it quicker and more responsive.
4. Security and patching: Maintenance makes sure the program is secure by routinely scanning for vulnerabilities and applying patches or upgrades to prevent unauthorized access.
5. Shared knowledge and documentation: Part of maintenance, which is the act of keeping the software's documentation up to date, is updating the user manuals and technical manuals.
6. Schedule for maintenance: The times and procedures for evaluating, updating, and validating the SDD should be laid out in a maintenance schedule. This ensures that maintenance tasks are routinely scheduled and completed.

By incorporating these maintenance practices, you can ensure that your SDD remains a reliable and valuable reference throughout the

software development process. Regular updates and reviews will help keep the document accurate, up to date, and in sync with the evolving system design.