INSTITUTE FOR ADVANCED COMPUTING AND SOFTWARE DEVELOPMENT (IACSD), AKURDI, PUNE

Documentation On

CAREHUB

(Hospital Facility App)

PG-DAC March 2023

Submitted By:

Group No: 47

Roll No.	Name:
233103	Swapnil Kumar Srivastava
233088	Shivam Bhosale

Mrs. Geeta Darunte
Project Guide

Mr. Rohit Puranik
Center Coordinator

ABSTRACT

The CAREHUB is a comprehensive software solution designed to streamline and optimize the operations of modern healthcare institutions. It serves as an integrated platform that facilitates the efficient management of patient information, medical records, administrative tasks, and various other processes within a hospital or medical facility.

The primary objective of the Hospital Facility System is to enhance the quality of patient care while simplifying administrative and clinical workflows. This system offers modules for patient registration, appointment scheduling, electronic health records (EHR) management, billing and invoicing, pharmacy and inventory management, laboratory and diagnostic services coordination, and more.

ACKNOWLEDGEMENT

I take this occasion to thank God, almighty for blessing us with his grace and taking our endeavor to a successful culmination. I extend my sincere and heartfelt thanks to our esteemed guide, **Mrs. Geeta Darunte** for providing me with the right guidance and advice at the crucial juncture stand for showing me the right way. I extend my sincere thanks to our respected **Centre Co-Ordinator Mr. Rohit Puranik**, for allowing us to use the facilities available. I would like to thank the other faculty members also, at this occasion. Last but not the least, I would like to thank my friends and family for the support and encouragement they have given me during the course of our work.

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

In the rapidly evolving landscape of healthcare, the effective management of hospital operations and patient care has become increasingly complex. CAREHUB has emerged as essential technological solutions to address these challenges and enhance the overall efficiency, quality, and coordination of healthcare services. An HMS is a comprehensive software platform designed to integrate and auto mate various administrative, clinical, and financial functions within a hospital or medical facility. It plays a pivotal role in modernizing healthcare practices, improving patient outcomes, and optimizing resource utilization.

The traditional paper-based methods of patient recordkeeping, appointment scheduling, and inventory management are not only time-consuming but also prone to errors, leading to potential lapses in patient care. The Hospital Management System aims to revolutionize these processes by providing a digital ecosystem that streamlines workflows, minimizes manual intervention, and ensures data accuracy.

FEATURES

CareHub is a software solution designed to streamline and optimize various administrative, operational, and clinical tasks within a healthcare facility. The features of a hospital management system can vary based on the specific software and the needs of the hospital, but here are some common features you might find in such a system:

1. **Patient Registration and Management:**
    - New patient registration with personal and medical information.
    - Maintain a comprehensive patient database with demographic details, medical history, and contact information.
    - Assign unique patient identifiers for easy tracking.

2. **Appointment Scheduling:**
    - Allow patients to request or schedule appointments with doctors and specialists.
    - Provide a calendar view for medical staff to manage appointments efficiently.
    - Send appointment reminders to patients via email, SMS, or notifications.

3. **Electronic Health Records (EHR) Management:**
    - Store and manage patient health records electronically, including medical history, diagnoses, treatment plans, medications, and test results.
    - Ensure data security and compliance with privacy regulations.

4. **Billing and Invoicing:**
    - Generate and manage invoices for services provided, including consultations, procedures, tests, and medications.
    - Integrate with insurance systems for claims processing.

- Provide detailed billing reports and summaries.

5. **Pharmacy Management:**
    - Manage medication inventory, including stock levels and expiration dates.
    - Generate electronic prescriptions and send them directly to the pharmacy.
    - Automate medication dispensing and tracking.
6. **Laboratory and Diagnostic Management:**
    - Schedule and track various diagnostic tests and procedures.
    - Record and store test results digitally.
    - Facilitate communication between doctors and laboratories.
7. **Inventory Management:**
    - Manage medical equipment, supplies, and consumables.
    - Monitor stock levels, reorder points, and usage patterns.
    - Streamline procurement processes.
8. **Doctor and Staff Management:**
    - Maintain profiles of medical and non-medical staff members.
    - Schedule and track shifts and responsibilities.
    - Assign privileges and access levels based on roles.
9. **Reports and Analytics:**
    - Generate reports on various aspects such as patient demographics, appointments, revenue, and more.
    - Analyze trends and patterns to make informed decisions.
10. **Integration and Interoperability:**
    - Integrate with external systems such as laboratory equipment, billing systems, and electronic health record exchanges.
    - Facilitate data sharing across departments for seamless communication.

PROJECT OBJECTIVE

The objectives of a Carehub is to enhance the efficiency, quality, and coordination of healthcare services within a medical facility. The system is designed to address various challenges faced by hospitals and medical institutions, and it aims to achieve the following key objectives:

1. **Streamlined Patient Care:** The primary objective of an Hospital facility is to streamline patient care processes by digitizing and automating administrative and clinical workflows. This includes patient registration, appointment scheduling, medical history recording, treatment planning, prescription management, and follow-up care.

2. **Efficient Data Management:** It aims to centralize and manage patient information, medical records, and other critical data in a digital format. This centralized repository ensures easy access to accurate and up-to-date patient information, leading to better decision-making by healthcare providers.

3. **Enhanced Patient Experience:** By reducing waiting times, minimizing paperwork, and improving appointment scheduling, it contributes to an overall improved patient experience. Patients feel more valued and cared for when their interactions with the hospital are smooth and well-coordinated.

4. **Accurate Billing and Financial Management:** The system automates billing and invoicing processes, reducing billing errors and ensuring timely and accurate financial transactions. This enhances transparency in financial dealings with patients and insurance providers.

5. **Optimized Resource Allocation:** It helps hospitals allocate resources more efficiently by providing insights into patient flow, occupancy rates, and appointment scheduling. This ensures that the right resources are available at the right time, leading to improved resource utilization.

6. **Real-time Communication:** The system facilitates seamless communication among different departments and healthcare providers within the hospital. This ensures that all relevant parties are informed about patient status, appointments, treatments, and test results in real time.

7. **Improved Clinical Decision-Making:** With access to comprehensive electronic health records (EHRs) and patient histories, healthcare providers can make more informed and accurate clinical decisions. This leads to improved diagnosis and better treatment outcomes.
8. **Effective Inventory and Pharmacy Management:** It optimizes inventory and pharmacy management, ensuring that medications and medical supplies are available when needed, minimizing wastage, and avoiding stockouts.

PROJECT OVERVIEW

This system involves its own database to be maintained. As the information or details about the patients are stored in the database (like RDBMS) for the server-side functionalities. The Server process is for dealing with the patient's detail and the items.

The application design contains two modules one is for the customers who wish to buy the articles. And another is for the store owners who maintain and updates the information regarding the patient details and about doctors schedule.

The application that is deployed on the hospital's database, the information regarding the items is highlighted and forwarded from the database for the hospital staff (front view) based on the schedule through the appointments day to day activity of hospital is carried out. As soon as the authorized personnel feeds the relevant data into the system, several reports are generated based on the security policy used.

PROJECT SCOPE

The scope of the Hospital Management System project encompasses the design, development, testing, and deployment of a comprehensive software solution that will streamline and optimize the operations of [Hospital Name]. The project will focus on creating a user-friendly and efficient system to manage various aspects of patient care, administrative tasks, and clinical workflows within the hospital.

STUDY OF THE SYSTEM

MODULES:

The system after careful analysis has been identified to be presented with the following modules and roles.

The modules involved are:

1. Admin
2. Patients
3. Hospital staff(Doctors, Ward Incharge, Receptionist, Pharmacist, Lab Incharge)

Admin:

The Admin is the super user of this application. Only Admin have access to this Admin page. Admin is the owner of one of the Applications . The Admin has all the

SYSTEM ANALYSIS

System Analysis of Hospital Management

System:

System analysis is a crucial phase in the development of a CAREHUB where the requirements, objectives, and constraints of the system are thoroughly examined and documented. This analysis phase lays the foundation for designing and developing the system to meet the needs of the hospital effectively

SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENTS

Title:

Software Requirement Specification for CAREHUB

Objective (Purpose):

The online **HFS (Hospital Facility System) i.e CAREHUB** which is a Web Application to provide complete solution for patients. It will enable patients to browse with help of Receptionist to make appointments removing the tedious long-standing queues to get fast seamless service of medication in hospital **starting from appointment upto medicines.**

User requirements:

The user requirements specify what the **CAREHUB** (HOSPITAL FACILITY SYSTEM->**HFS**) system must be capable of doing to solve the problems of the set of potential users of HFS. The system owners and end-users write them with information from Quality Assurance. The user requirements are identified using natural language, along with diagrams for a broader understanding.

Scope:

The scope of a hospital management system (HMS) is extensive, covering various aspects of hospital operations. An HMS typically includes functionalities like patient registration, appointment scheduling, electronic medical records (EMR), billing, inventory management, pharmacy management, and laboratory management.

Functional requirements

Functional requirements define the function of the device, explaining the actions taken by the HMS system clearly and quantitatively. These requirements define the capabilities of the HMS system, as well as its process or workflow. It also determines the form of input and output desired. Some of the functional requirements of the HMs system are:

Registration:

The admin is the only user who can register **doctors, receptionist, lab in charge, ward in charge and pharmacists.** The patient can register through the registration module. In addition, the system should allow a receptionist to register new patients into the system.

Authentication:

The user (patients) should be able to create an account in the HMS system through the registration module (form). The following details should be entered in the registration module:

1 User_id
2 firstname
3 lastname
4 address
5 email
6 age
7 username
8 password

The system must enable registered users including all categories of users to log in through the login module, by entering the username and password.

Manage Account

Users of the HMS system should have the ability to manage their account including changing their password.

Input validation

The system should validate all inputs entered by the user to ensure that the user does not leave any blank required fields. This is also important to ensure that the inputs are entered in the correct format and does not exceed the size specifies.

Unique Username (ID)

The username of each user in the HMS system should be unique. Equally important, the system should verify that the username, which is entered in the registration form, is not already used by another user in the system.

Allot patients for doctors

The system shall enable **Receptionist** to view doctors' status (Schedule) to allot patients successfully for the concerned doctors in terms of their problem.

Appointment List

The receptionist shall be able to view the full appointment list.

Booking an appointment

The system shall provide the available appointment to receptionist and patients in order to book a new appointment.

Updating Schedules

The system shall update the schedules of users automatically whenever a new appointment is booked or a patient is redirected.

Inform (Notify) Users

The system shall inform doctors, receptionist, lab assistants, pharmacist whenever an action has to be taken by them.

Creating Prescription

The doctor shall user the system to create/enter a prescription for the patient. On the other hand, the patient also should have access to view the prescription through his/her account.

Redirecting Patients

The system shall enable doctors to redirect the patient for diagnosis via lab test.

Generating Reports

The system shall enable lab assistants to generate test reports like X-Ray images, CT scan, MRI reports.

View Reports by patient

The patients shall use the system to view their test results reports as well as doctors' advice and prescription.

View Reports by doctor

The system shall allow doctors to view the patients' reports and enter required advice for the patient, as well as new prescriptions if needed.

Examination and Medicine Costs

The system shall allow lab assistants and pharmacist to enter the costs of the examinations and medicines.

Check Out (Payment)

The system shall allow the receptionist to create and order invoice for payment through the billing module after status of patient from in to out . The receptionist shall watch the payment history of the patients.

Non-Functional Requirements

The non-functional requirement defines the operational requirements off the system, as well as the constraints to be followed in order to improve the system's functionality. The following are some of the non-functional requirement that needs to be considered in the HMS system:

Availability

The system must be available 24/7.

Capacity

The system must support a load of 3000 users at a time.

1. Response Time: The system must respond within 2 seconds after verifying the details and other data of the patient. In other words, the time to load a web page over a 56Kbps modem connection should not exceed 2 seconds.

2.User Interface: User interface display shall response within 5 seconds.

3.Conformity: The system should be in accordance with Windows Accessibility.

4.Virus Protection: Devices in the hospital that use the system must have firewalls enabled and an Active Anti-Virus in usage.

Performance

Durability

In case of failure, the system should be recovered by itself within 10 seconds, and the server must receive a comprehensive crash report stating the problem occurred.

Adaptability

The system (web application) must be adaptive and responsive to support devices of all types. 21

Security

1. **Modification**: Changes in the system like (insert, erase, and update) is coordinated and performed by the Admin only.
2. **User Rights**: users' activity of the system should be controlled so that each user can access the allowable activities only.
3. **Data**: the transaction data should be transmitted in an encrypted form.
4. **Database Protection**: the database should be protected by a strong password.

Safety and Maintainability

A backup of the database should be performed every week, so that the system can be recovered in case of any database damage, which may be occurred due to a catastrophic failure, such as a disk crash.

Accessibility

The system can be accessed by the Admin and many other users but the access level is controlled of each user as per their scope of work.

System Requirements

System requirements define the specifications required to use the system. The following are some of the System requirements that need to be considered for the HMS system:

1. The system needs to run seamlessly with 1 GB of Random Access Memory (RAM).
2. The system needs a minimum Internet speed of 500 kbps to successfully refresh and load pages/modules.
3. The system (webApp) needs internet connectivity to work to access.

SYSTEM DESIGN

System design is the solution for the creation of a new system. This phase focuses on the detailed implementation of the feasible system. Its emphasis on translating design. Specifications to performance specification.

System design has two phases of development.

1. Logical Design

2. Physical Design

During logical design phase the analyst describes inputs (sources), outputs(destinations), databases (data sores) and procedures (data flows) all in a format that meets the user requirements. The analyst also specifies the needs of the user at a level that virtually determines the information flow in and out of the system and the data resources. Here the logical design is done through data flow diagrams and database design. The physical design is followed by physical design or coding. Physical design produces the working system by defining the design specifications, which specify exactly what the candidate system must do. The programmers write the necessary programs that accept input from the user, perform necessary processing on accepted data and produce the required report on a hard copy or display it on the screen.

INPUT AND OUTPUT DESIGN

INPUT DESIGN:

Input design is the link that ties the information system into the world of its users. The input design involves determining the inputs, validating the data, minimizing the data entry and providing a multi-user facility. Inaccurate inputs are the most common cause of errors in data processing. Errors entered by the data entry operators can be controlled by input design. The user-originated inputs are converted to a computer-based format in the input design. Input data are collected and organized into groups of similar data. Once identified, the appropriate input media are selected for processing. All the input data are validated and if any data violates any conditions, the user is warned by a message. If the data satisfies all the conditions, it is transferred to the appropriate tables in the database. In this project the customer details are to be entered at the time of registration. A page is designed for this purpose which is user friendly and easy to use. The design is done such that users get appropriate messages when exceptions occur.

OUTPUT DESIGN:

Computer output is the most important and direct source of information to the user. Output design is a very important phase since the output needs to be in an efficient manner. Efficient and intelligible output design improves the system relationship with the user and helps in decision making. Allowing the user to view the sample screen is important because the user is the ultimate judge of the quality of output. The output module of this system is the selected notifications.

DATABASE DESIGN

DATABASE: (8.0.32 MYSQL)

Databases are the storehouses of data used in the software systems. The data is stored in tables inside the database. Several tables are created for the manipulation of the data for the system.

Two essential settings for a database are

1. Primary key - the field that is unique for all the record occurrences
2. Foreign key - the field used to set relation between tables Normalization is a technique to avoid redundancy in the tables.

SYSTEM TOOLS

The various system tools that have been used in developing both the front end and the back end of the project are being discussed in this chapter.

FRONT END: (React Version 18.2.0)

React is a library which is developed by Facebook are utilized to implement the frontend.

React (also known as React.js or ReactJS) is a free and open-source frontend JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single page or mobile applications.

However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

BACKEND: (Spring Boot Version 2.7.2)

The back end is implemented using MySQL which is used to design databases.

MySQL:

MySQL is the world's second most widely used open-source relational database management system (RDBMS).

The SQL phrase stands for Structured Query Language.

Spring-Boot:

This is used to connect MYSQL and fetch data from database and store the data in database.

The Spring Framework is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE (Enterprise Edition) platform. Although the framework does not impose any specific programming model, it has become popular in the Java community as an addition to the Enterprise JavaBeans (EJB) model. The Spring Framework is Open-source Framework.

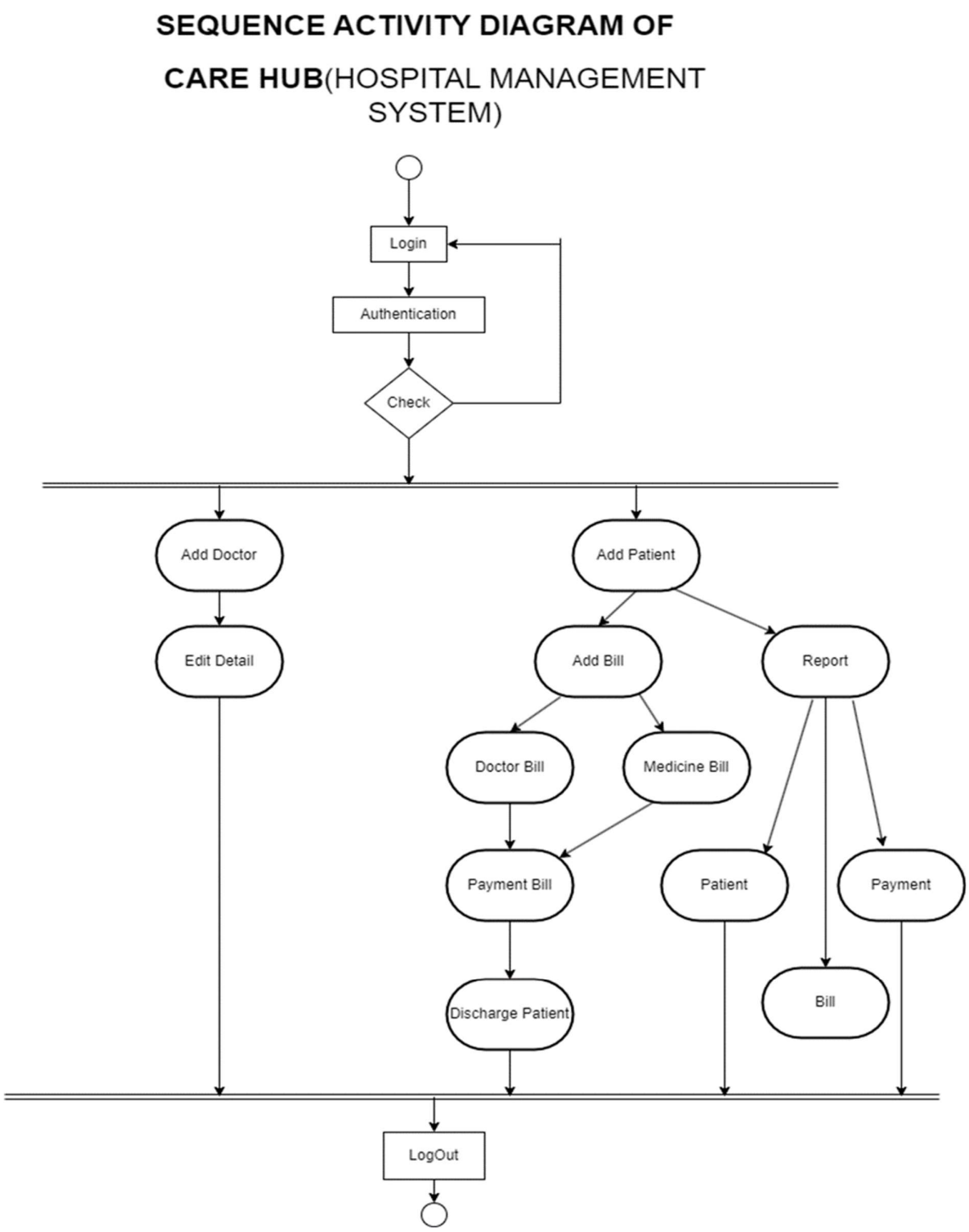


Figure 1 Class Diagram

1. Add Patient

Receptionist are allowed to add patient

2. Add Doctor

Doctor is added by the authorized user which is admin

3. Edit Details

Doctor edit details of Patient

4. Add to Bill

Patient prescription bill is added

5. Add Report

Report bill of test done by lab.

6. Doctor Bill

Doctor diagnosis bill is added

7. Medicine Bill

Medicine bill given by pharmacist

8. Payment Bill

Receptionist collect all patient bill after receipt from doctor , lab incharge, and medicine bill from pharmacist

9. Patient Report

Patient is given the report after the collection of bill

CAREHUB CLASS DIAGRAM

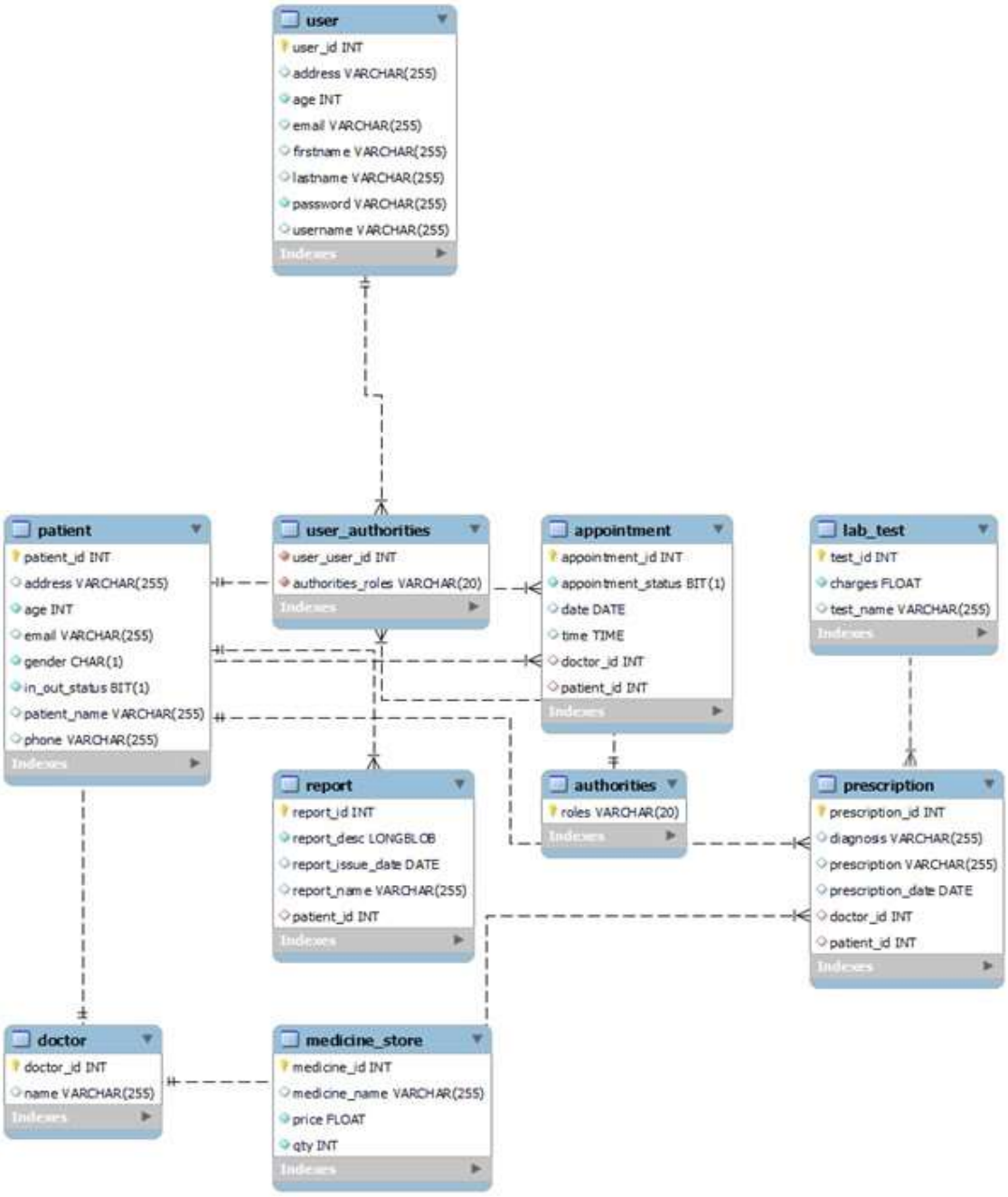


Figure 2 Class Diagram

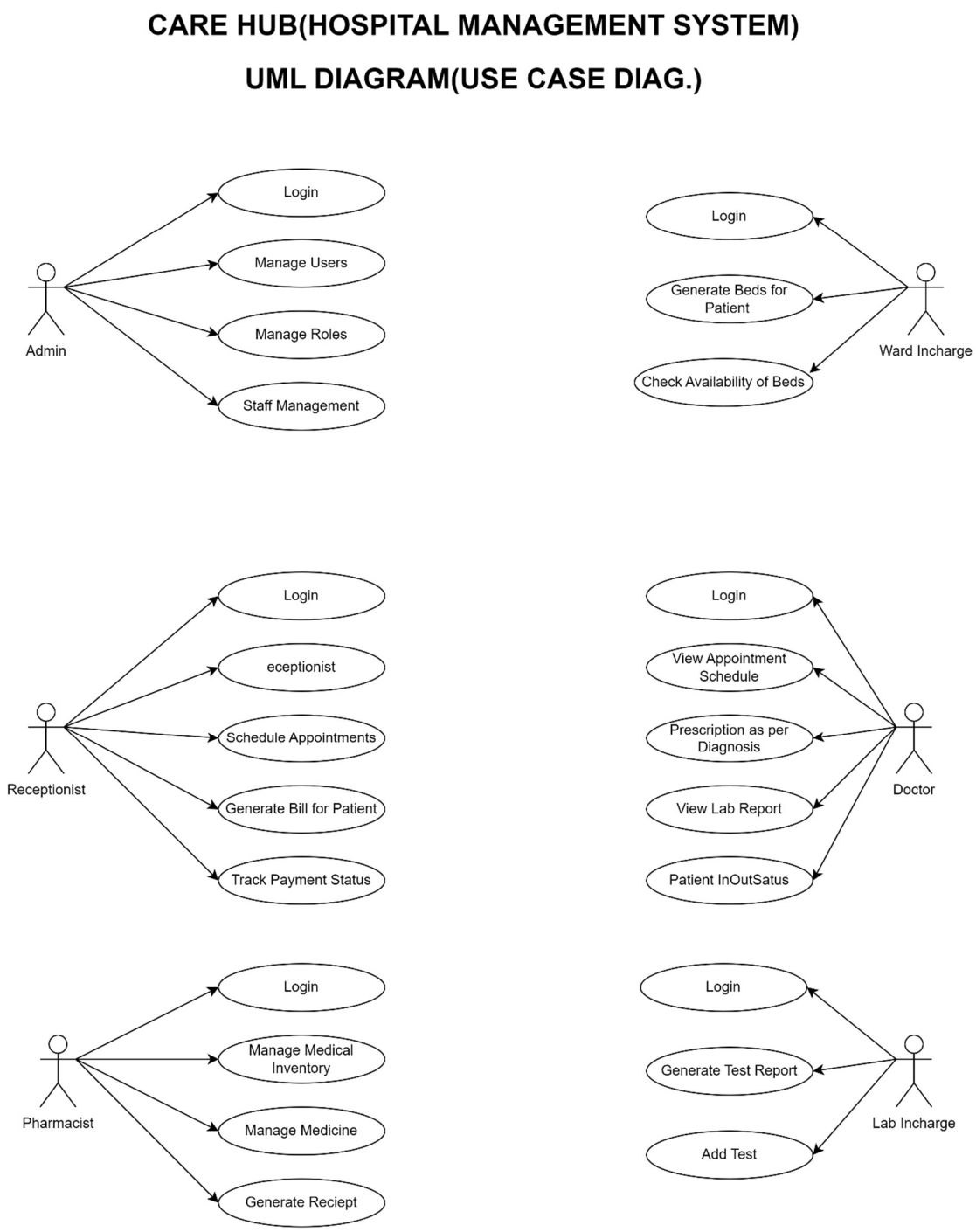


Figure 3 Use Case Diagram

CAREHUB E-R DIAGRAM

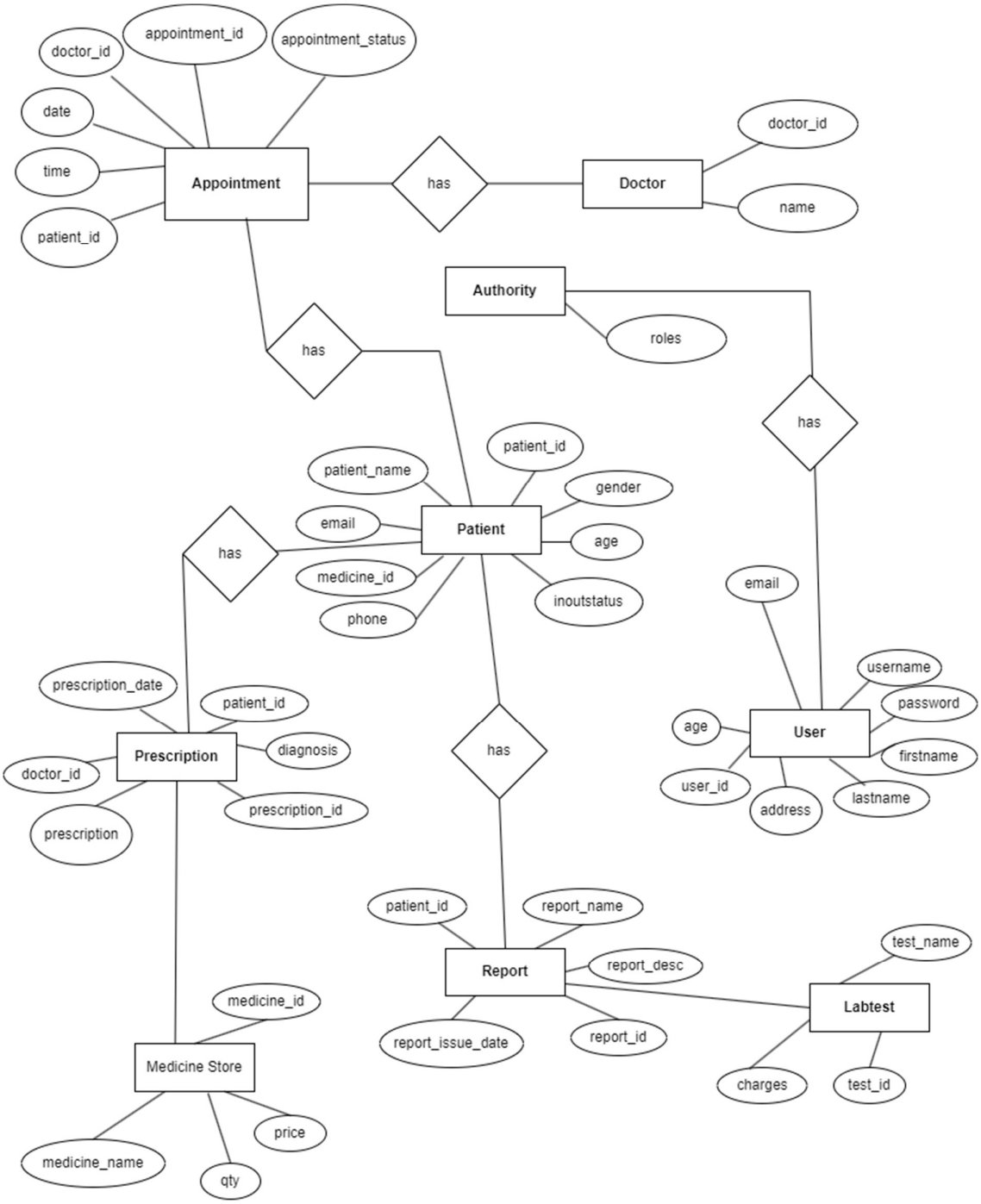
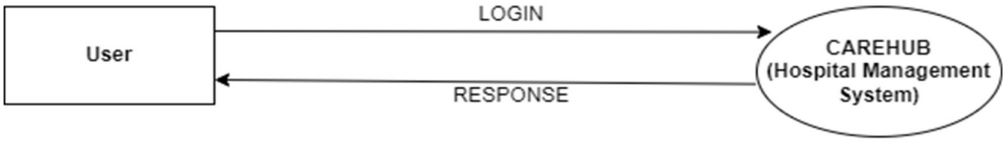


Figure 4 E-R Diagram

CARE HUB
DATA FLOW DIAGRAM

0 LEVEL DFD



1st LEVEL DFD

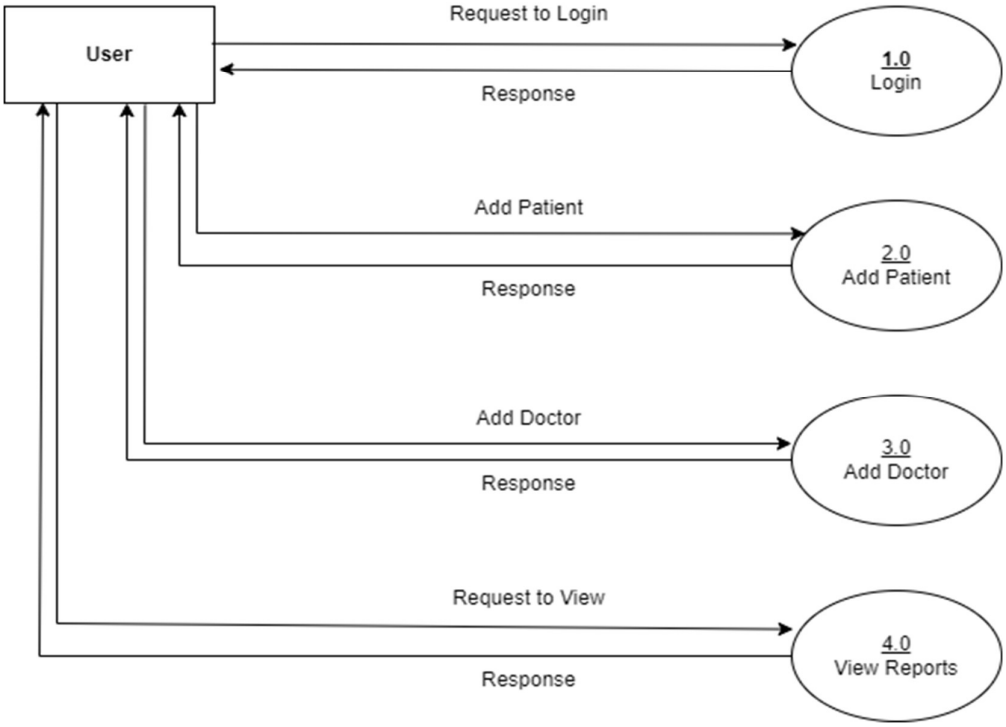


Figure 5 DFD Diagram

TABLE STRUCTURE:

Tables:-

```
mysql> show tables;
+---------------------+
| Tables_in_hospital  |
+---------------------+
| appointment         |
| authorities         |
| doctor              |
| lab_test            |
| medicine_store      |
| patient             |
| prescription        |
| report              |
| user                |
| user_authorities    |
+---------------------+
10 rows in set (0.52 sec)
```

Appointment table:-

```
mysql> desc appointment;
+-------------------+---------+------+-----+---------+----------------+
| Field             | Type    | Null | Key | Default | Extra          |
+-------------------+---------+------+-----+---------+----------------+
| appointment_id    | int     | NO   | PRI | NULL    | auto_increment |
| appointment_status | bit(1) | NO   |     | NULL    |                |
| date              | date    | YES  |     | NULL    |                |
| time              | time    | YES  |     | NULL    |                |
| doctor_id         | int     | YES  | MUL | NULL    |                |
| patient_id        | int     | YES  | MUL | NULL    |                |
+-------------------+---------+------+-----+---------+----------------+
6 rows in set (0.37 sec)
```

Authorities table:-

```
mysql> desc authorities;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| roles | varchar(20) | NO   | PRI | NULL    |       |
+-------+-------------+------+-----+---------+-------+
1 row in set (0.10 sec)
```

Doctor table:-

```
mysql> desc doctor;
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| doctor_id | int          | NO   | PRI | NULL    |       |
| name      | varchar(255) | YES  |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

Lab Test Table:-

```
mysql> desc lab_test;
+-----------+--------------+------+-----+---------+----------------+
| Field     | Type         | Null | Key | Default | Extra          |
+-----------+--------------+------+-----+---------+----------------+
| test_id   | int          | NO   | PRI | NULL    | auto_increment |
| charges   | float        | NO   |     | NULL    |                |
| test_name | varchar(255) | YES  |     | NULL    |                |
+-----------+--------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```

Medicine Store Table:-

```
mysql> desc medicine_store;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| medicine_id   | int          | NO   | PRI | NULL    | auto_increment |
| medicine_name | varchar(255) | YES  |     | NULL    |                |
| price         | float        | NO   |     | NULL    |                |
| qty           | int          | NO   |     | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

Patient Table:-

```
mysql> desc patient;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| patient_id    | int          | NO   | PRI | NULL    | auto_increment |
| address       | varchar(255) | YES  |     | NULL    |                |
| age           | int          | NO   |     | NULL    |                |
| email         | varchar(255) | YES  |     | NULL    |                |
| gender        | char(1)      | NO   |     | NULL    |                |
| in_out_status | bit(1)       | NO   |     | NULL    |                |
| patient_name  | varchar(255) | YES  |     | NULL    |                |
| phone         | varchar(255) | YES  |     | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)
```

Prescription Table:-

```
mysql> desc prescription;
+-------------------+--------------+------+-----+---------+----------------+
| Field             | Type         | Null | Key | Default | Extra          |
+-------------------+--------------+------+-----+---------+----------------+
| prescription_id   | int          | NO   | PRI | NULL    | auto_increment |
| diagnosis         | varchar(255) | YES  |     | NULL    |                |
| prescription      | varchar(255) | YES  |     | NULL    |                |
| prescription_date | date         | YES  |     | NULL    |                |
| doctor_id         | int          | YES  | MUL | NULL    |                |
| patient_id        | int          | YES  | MUL | NULL    |                |
+-------------------+--------------+------+-----+---------+----------------+
6 rows in set (0.10 sec)
```

Report Table:-

```
mysql> desc report;
+-------------------+--------------+------+-----+---------+----------------+
| Field             | Type         | Null | Key | Default | Extra          |
+-------------------+--------------+------+-----+---------+----------------+
| report_id         | int          | NO   | PRI | NULL    | auto_increment |
| report_desc       | longblob     | NO   |     | NULL    |                |
| report_issue_date | date         | YES  |     | NULL    |                |
| report_name       | varchar(255) | YES  |     | NULL    |                |
| patient_id        | int          | YES  | MUL | NULL    |                |
+-------------------+--------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

User Table:-

```
mysql> desc user;
+-----------+--------------+------+-----+---------+----------------+
| Field     | Type         | Null | Key | Default | Extra          |
+-----------+--------------+------+-----+---------+----------------+
| user_id   | int          | NO   | PRI | NULL    | auto_increment |
| address   | varchar(255) | YES  |     | NULL    |                |
| age       | int          | NO   |     | NULL    |                |
| email     | varchar(255) | YES  | UNI | NULL    |                |
| firstname | varchar(255) | YES  |     | NULL    |                |
| lastname  | varchar(255) | YES  |     | NULL    |                |
| password  | varchar(255) | NO   |     | NULL    |                |
| username  | varchar(255) | YES  | UNI | NULL    |                |
+-----------+--------------+------+-----+---------+----------------+
8 rows in set (0.14 sec)
```

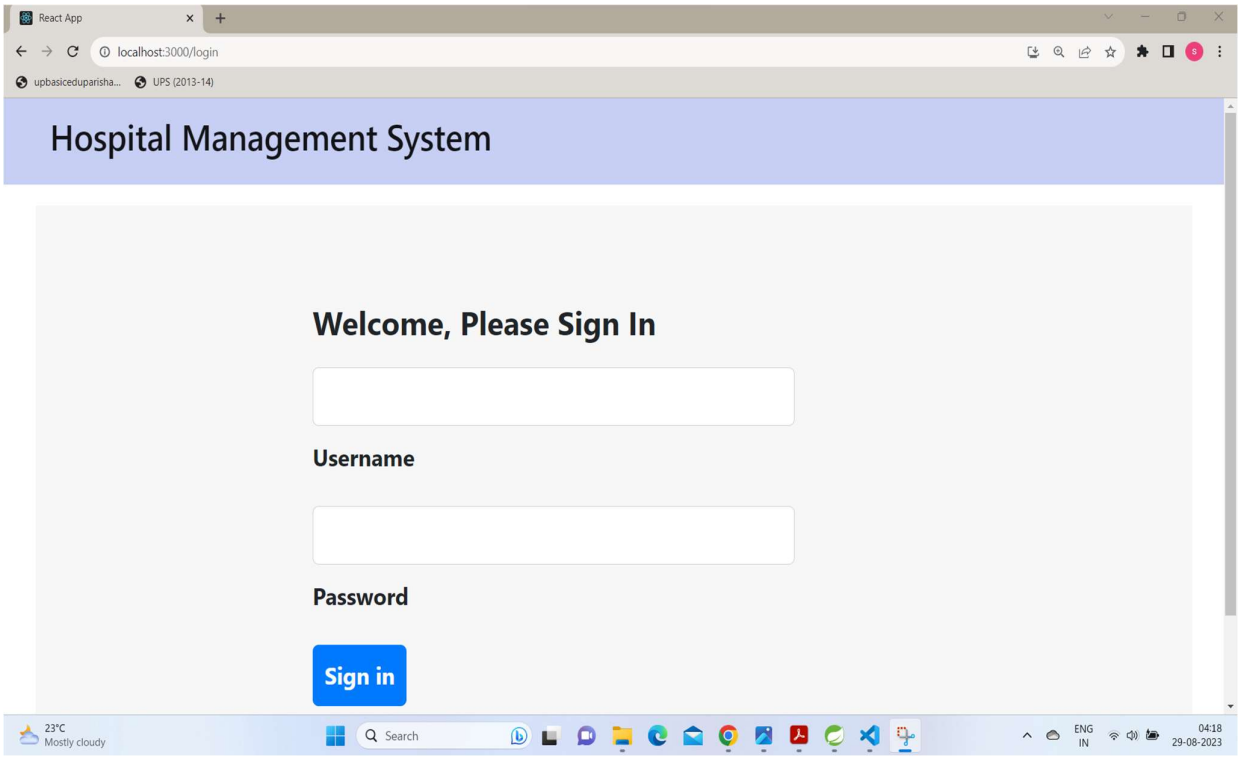
Authorities Table:-

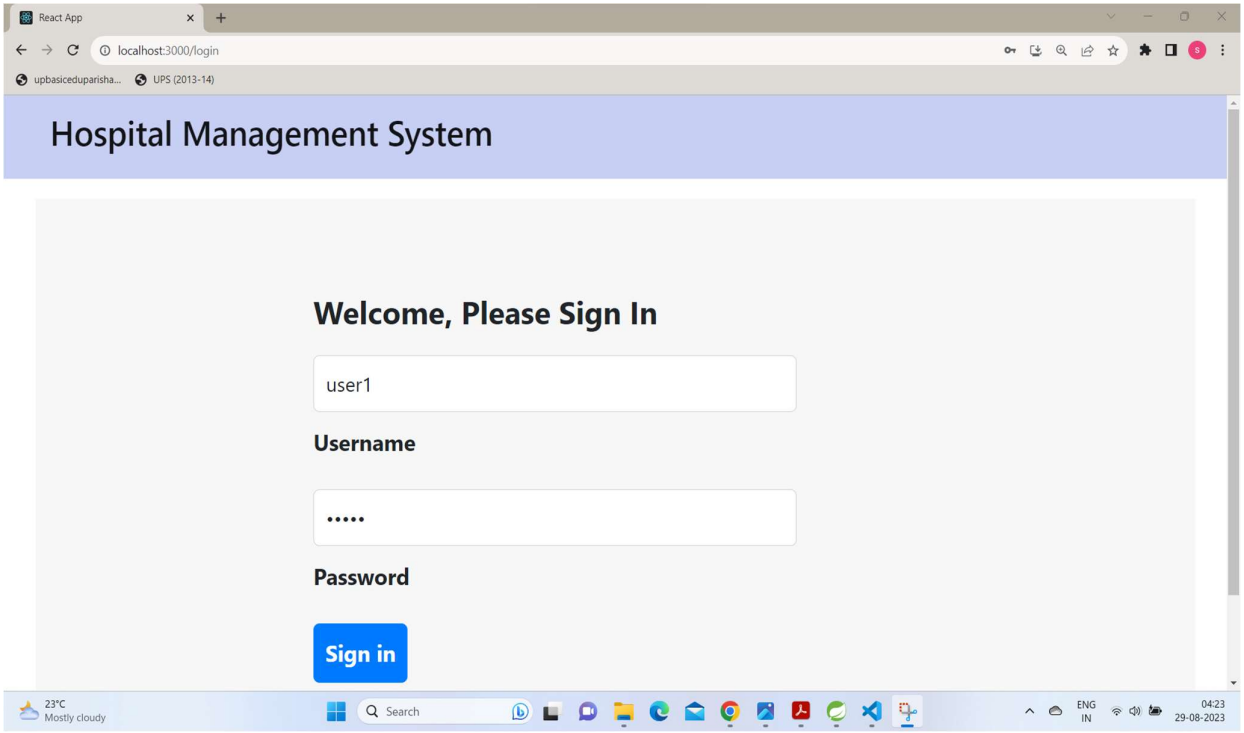
```
mysql> desc authorities;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| roles | varchar(20) | NO   | PRI | NULL    |       |
+-------+-------------+------+-----+---------+-------+
1 row in set (0.00 sec)
```

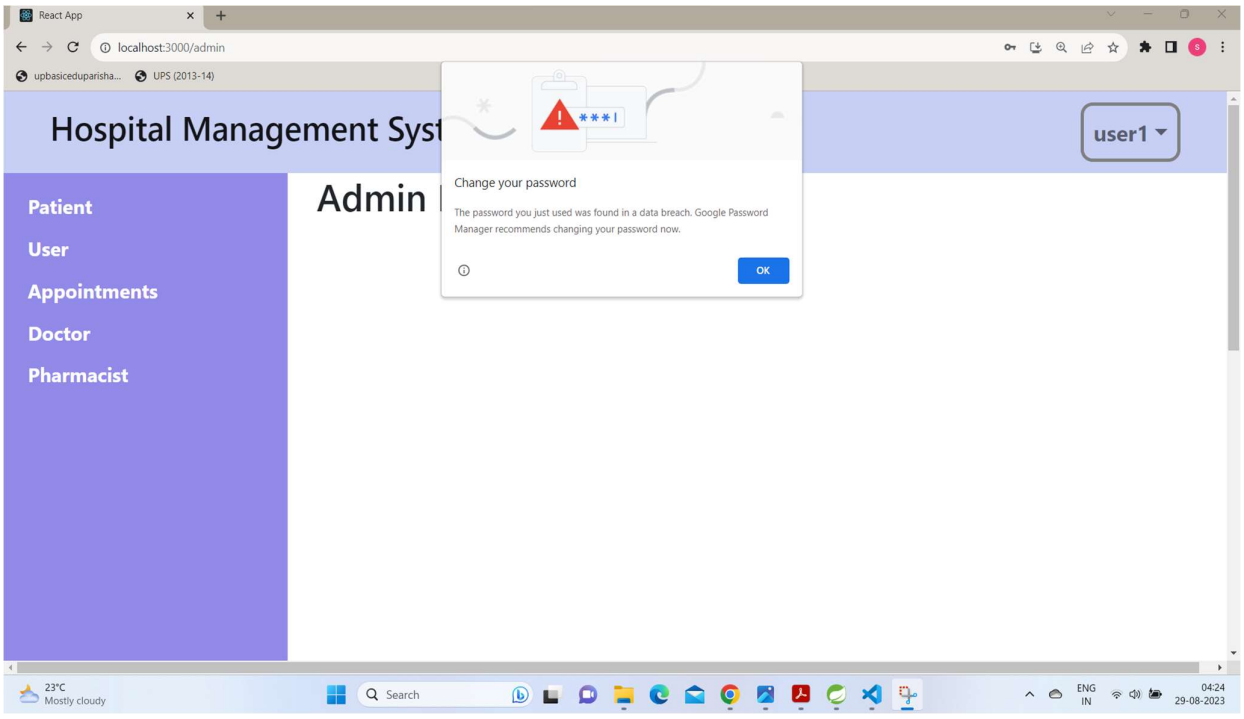
User Authorities Table:-

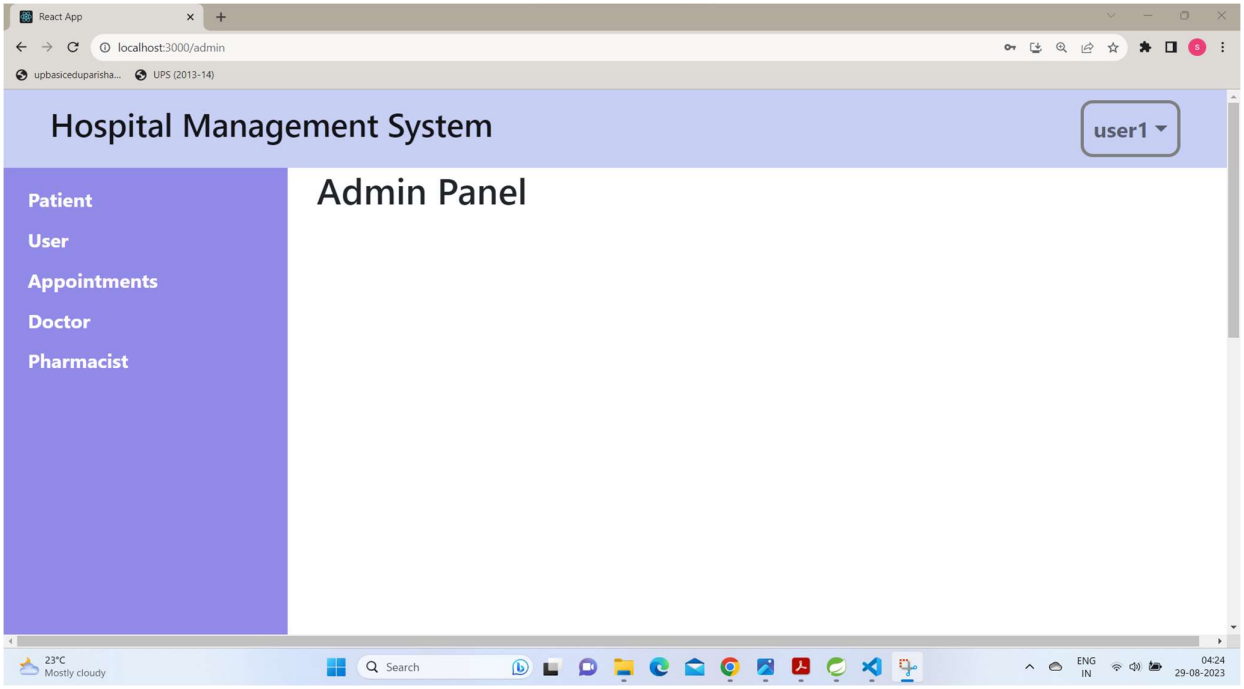
```
mysql> desc authorities;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| roles | varchar(20) | NO   | PRI | NULL    |       |
+-------+-------------+------+-----+---------+-------+
1 row in set (0.00 sec)
```

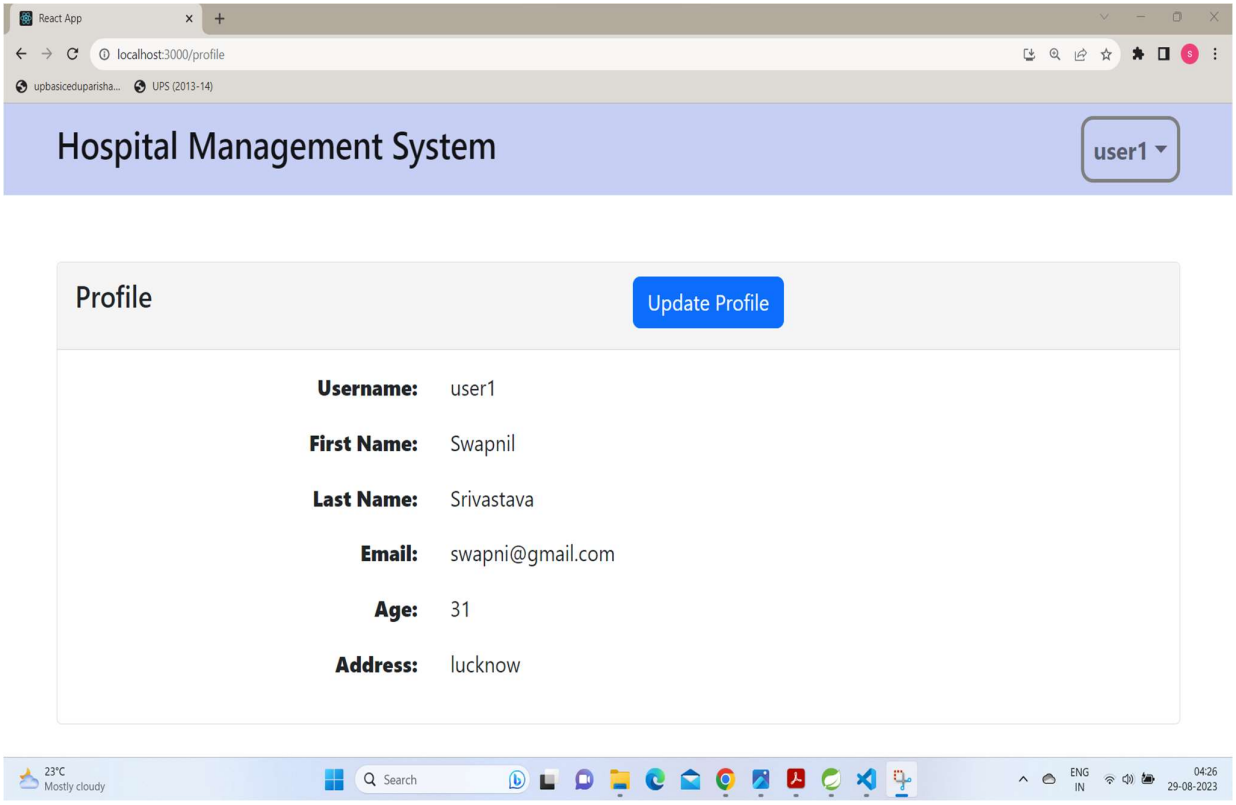
PROJECT DIAGRAM

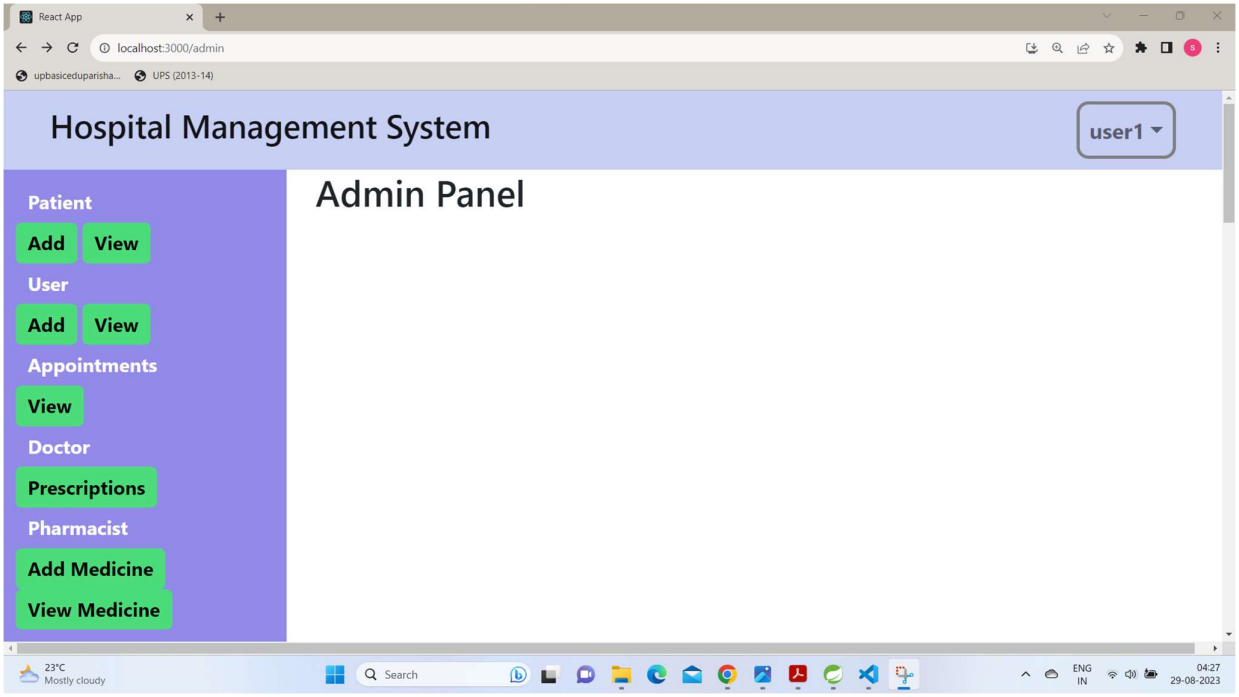


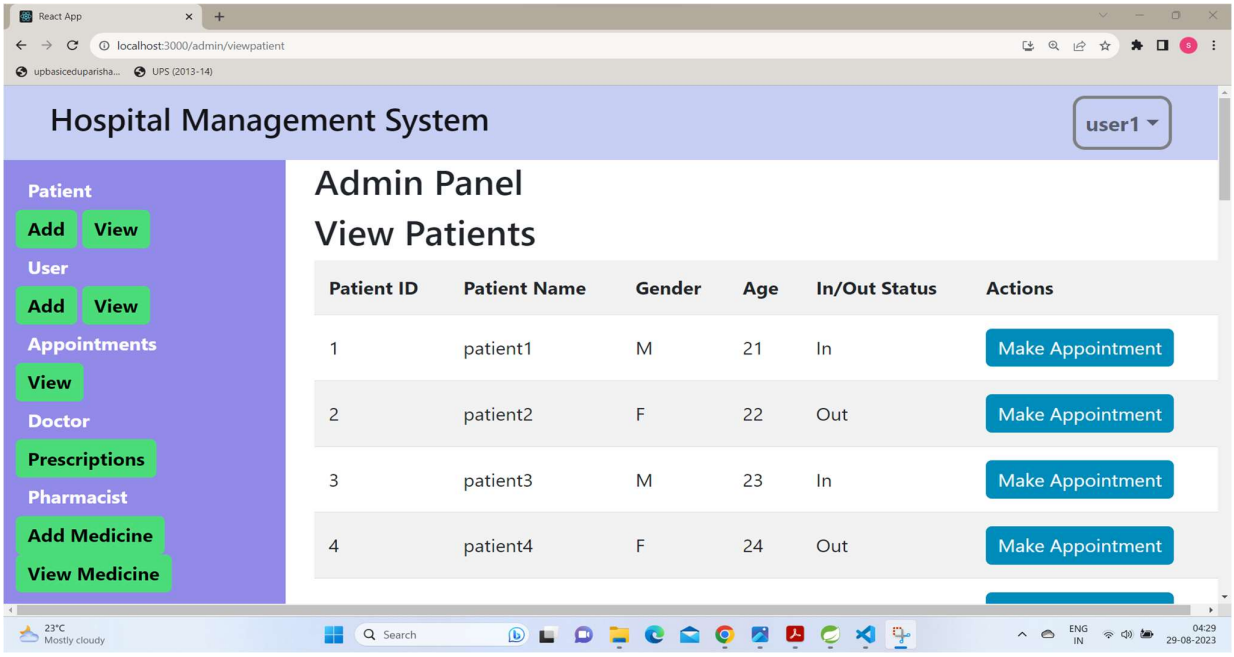

React App
localhost:3000/login
Hospital Management System
Welcome, Please Sign In
user1
Username
Password
Sign in

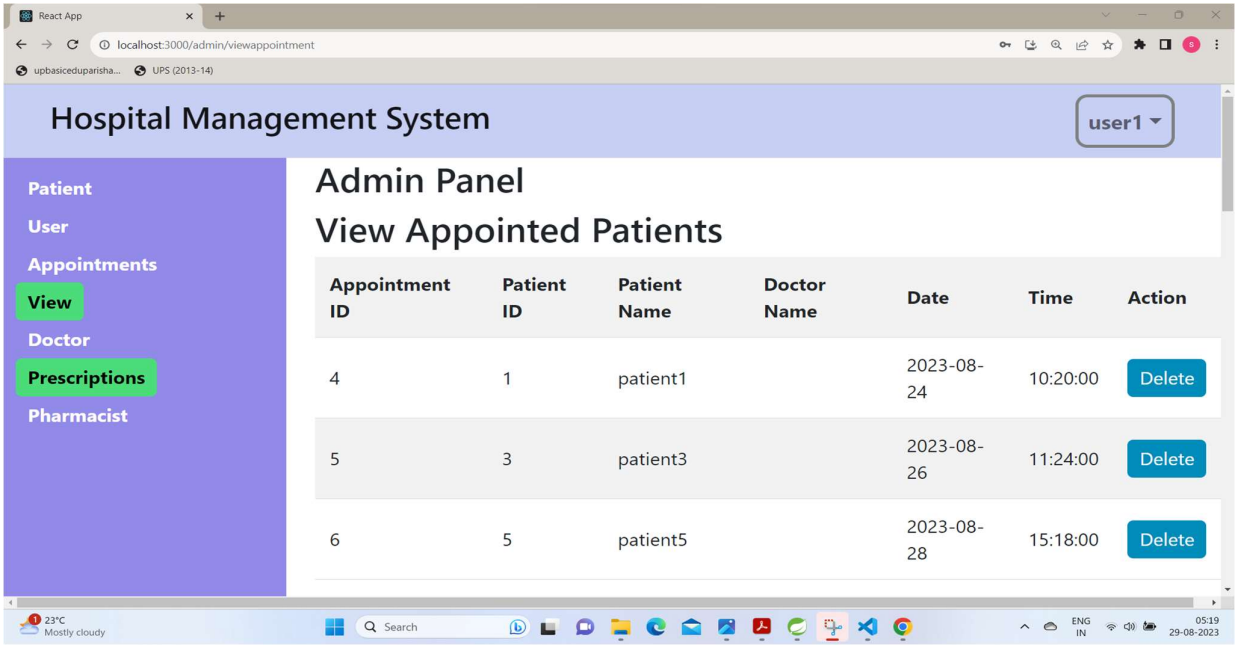

React App
localhost:3000/admin
Hospital Management Syst
user1
Patient
User
Appointments
Doctor
Pharmacist
Admin
Change your password
The password you just used was found in a data breach. Google Password Manager recommends changing your password now.
OK

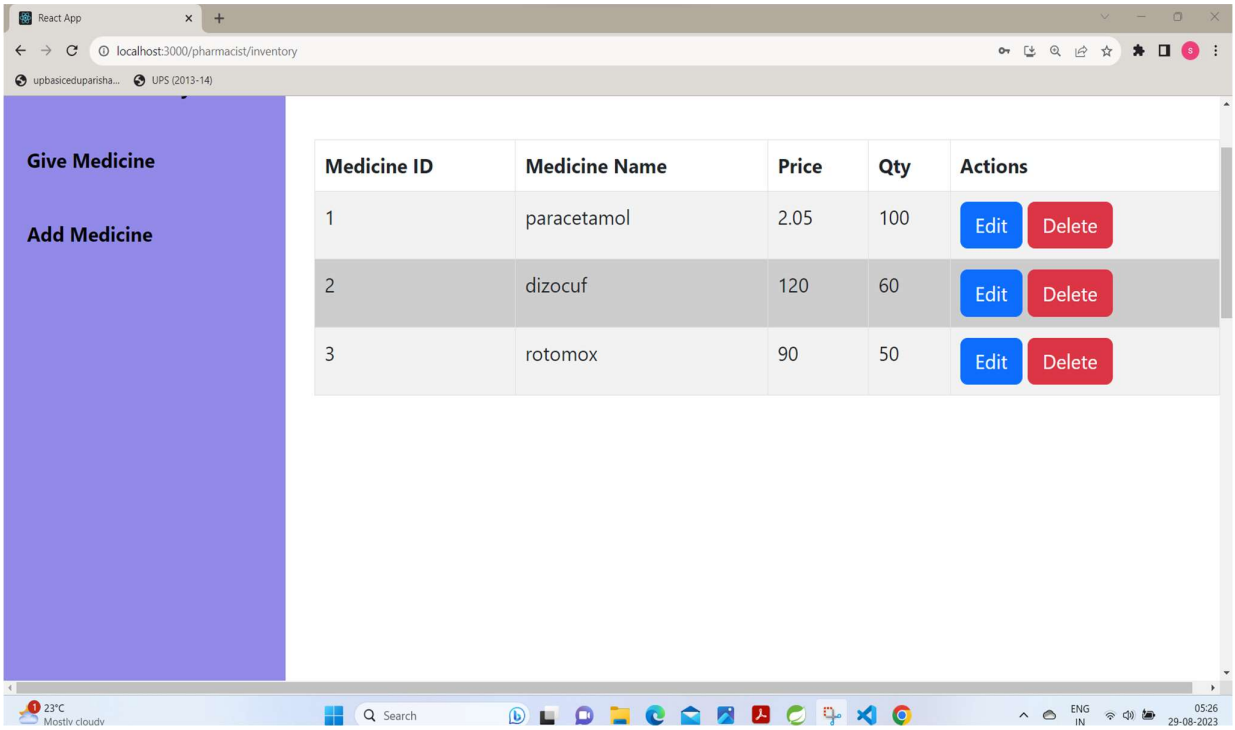

React App
localhost:3000/admin
Hospital Management System
user1
Patient
User
Appointments
Doctor
Pharmacist
Admin Panel

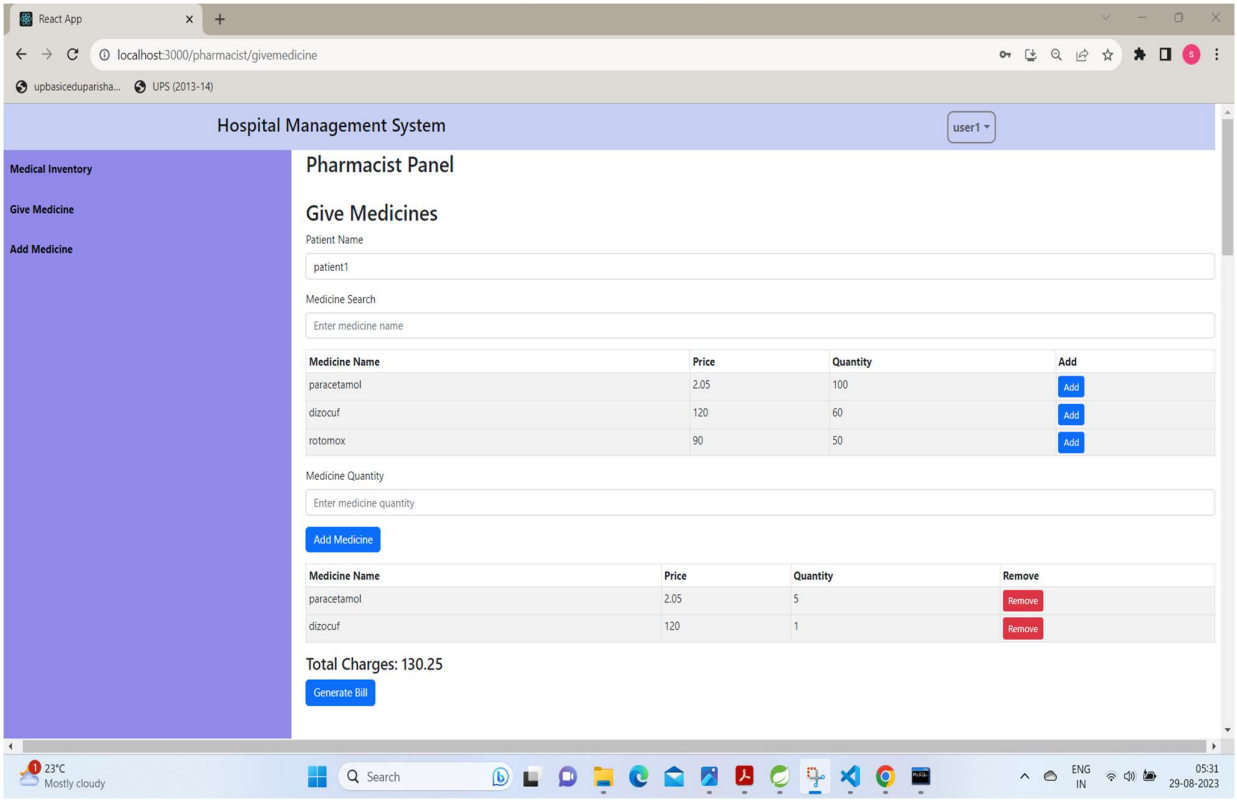

React App
localhost:3000/profile
Hospital Management System
user1
Profile
Update Profile
Username: user1
First Name: Swapnil
Last Name: Srivastava
Email: swapni@gmail.com
Age: 31
Address: lucknow

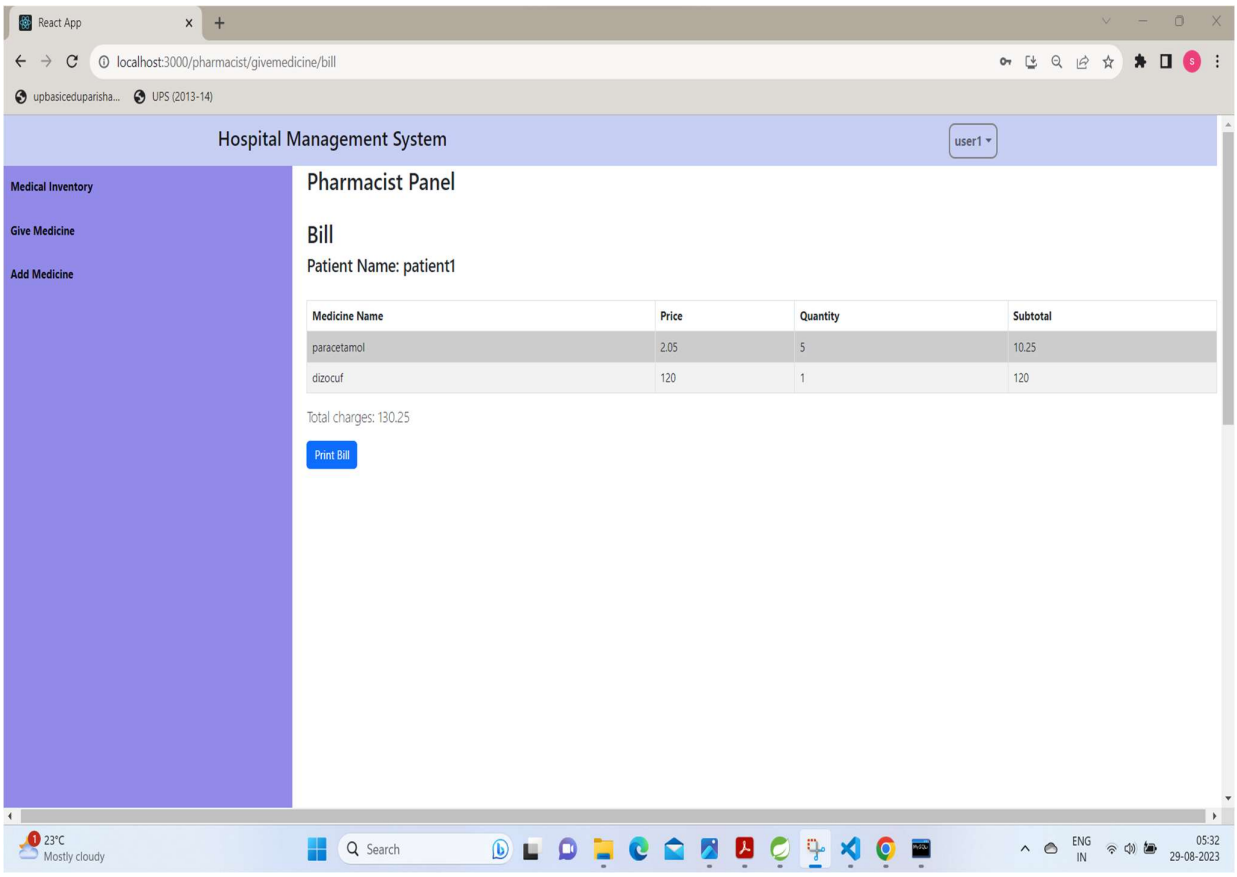

React App
localhost:3000/admin
Hospital Management System
user1
Patient
Add
View
User
Add
View
Appointments
View
Doctor
Prescriptions
Pharmacist
Add Medicine
View Medicine
Admin Panel

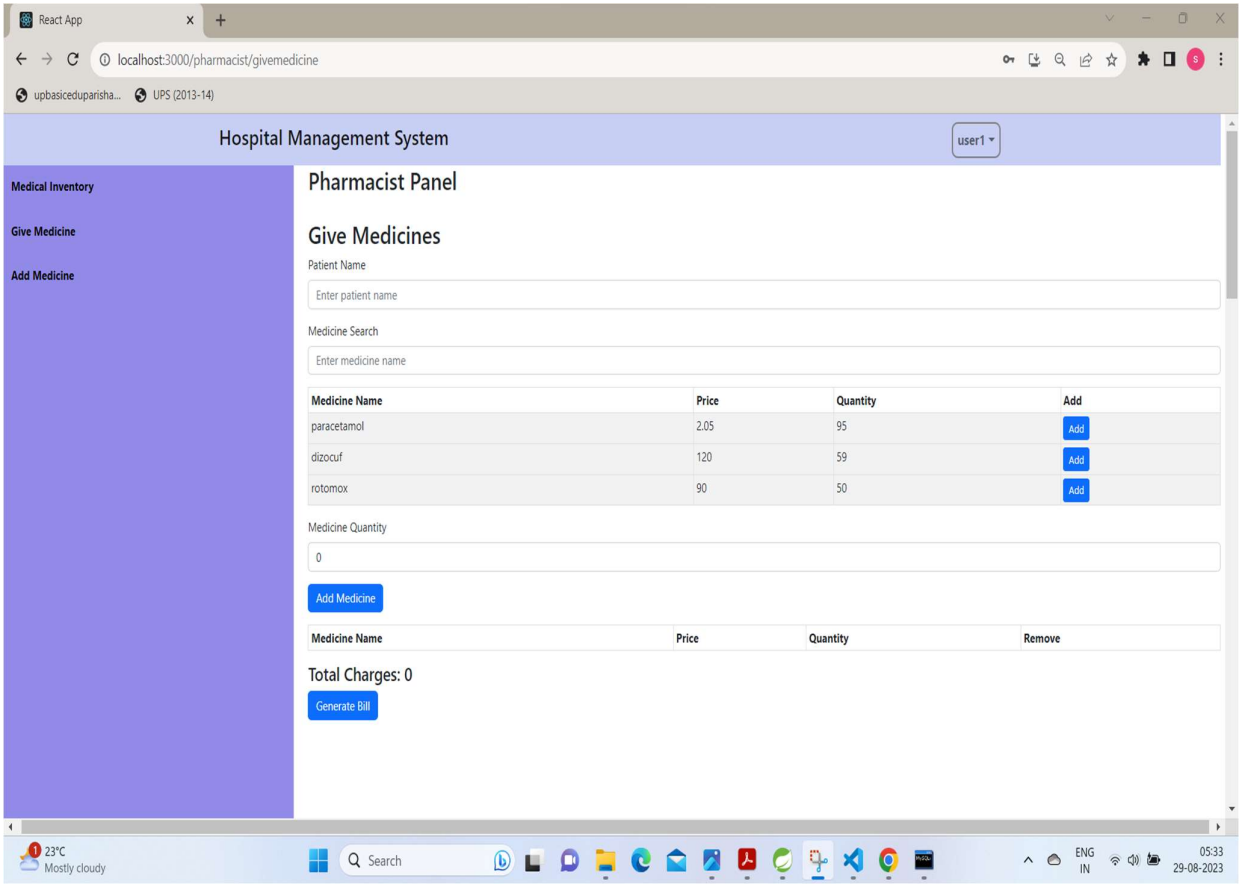

React App
localhost:3000/admin/viewpatient
Hospital Management System
user1
Patient
Add
View
User
Add
View
Appointments
View
Doctor
Prescriptions
Pharmacist
Add Medicine
View Medicine
Admin Panel
View Patients
Patient ID
Patient Name
Gender
Age
In/Out Status
Actions
1
patient1
M
21
In
Make Appointment
2
patient2
F
22
Out
Make Appointment
3
patient3
M
23
In
Make Appointment
4
patient4
F
24
Out
Make Appointment


React App
localhost:3000/admin/viewappointment
Hospital Management System
user1
Patient
User
Appointments
View
Doctor
Prescriptions
Pharmacist
Admin Panel
View Appointed Patients
Appointment ID
Patient ID
Patient Name
Doctor Name
Date
Time
Action
4
1
patient1
2023-08-24
10:20:00
Delete
5
3
patient3
2023-08-26
11:24:00
Delete
6
5
patient5
2023-08-28
15:18:00
Delete


React App
localhost:3000/pharmacist/inventory
Give Medicine
Add Medicine
Medicine ID
Medicine Name
Price
Qty
Actions
1
paracetamol
2.05
100
Edit
Delete
2
dizocuf
120
60
Edit
Delete
3
rotomox
90
50
Edit
Delete


React App
localhost:3000/pharmacist/givemedicine
Hospital Management System
user1
Medical Inventory
Give Medicine
Add Medicine
Pharmacist Panel
Give Medicines
Patient Name
patient1
Medicine Search
Enter medicine name
Medicine Name
Price
Quantity
Add
paracetamol
2.05
100
dizocuf
120
60
rotomox
90
50
Medicine Quantity
Enter medicine quantity
Add Medicine
Remove
Total Charges: 130.25
Generate Bill


React App
localhost:3000/pharmacist/givemedicine/bill
Hospital Management System
user1
Medical Inventory
Give Medicine
Add Medicine
Pharmacist Panel
Bill
Patient Name: patient1
Medicine Name
Price
Quantity
Subtotal
paracetamol
2.05
5
10.25
dizocuf
120
1
120
Total charges: 130.25
Print Bill


React App
localhost:3000/pharmacist/givemedicine
upbasiceduparisha...
UPS (2013-14)
Hospital Management System
user1
Medical Inventory
Give Medicine
Add Medicine
Pharmacist Panel
Give Medicines
Patient Name
Enter patient name
Medicine Search
Enter medicine name
Medicine Name
Price
Quantity
Add
paracetamol
2.05
95
Add
dizocuf
120
59
Add
rotomox
90
50
Add
Medicine Quantity
0
Add Medicine
Medicine Name
Price
Quantity
Remove
Total Charges: 0
Generate Bill
23°C
Mostly cloudy
Search
ENG
IN
05:33
29-08-2023

Conclusion

In conclusion, this report discusses a case study of an online web-based **CAREHUB** -> Hospital Facility System(HFS).

The aim, objectives, background and problem statement of the project, followed by, the system description, its main actors and modules are clearly stated. The UML , DFD ,E-R diagrams and behavioural diagrams described the system and illustrated the relationship between objects, and the processes flow of the HFS system, which helps to better understand the system. The discussed design of the interfaces provides good user experience and offers accessibility features for users of the system. Through the identified verification tools, the HFS system was ensured to work as per the expectations.

References

Hospital Management System(HMS) UI Design. [Online]. Available at: https://www.behance.net/gallery/3431329/Hospital-Management-(HMS)-UI-Design.

Creately, 2019. DataFlow Diagram - Creately. [Online]. Available at: https://app.creately.com/diagram/K73lE9bKGf4/view.

Encyclopedia, 2019. User requirements specification. [Online]. Available at: https://www.encyclopedia.com/computing/dictionaries-thesauruses-pictures-and-press-releases/user-requirements-specification.

GeeksForGeeks, 2019. Unified Modeling Language (UML) | Sequence Diagrams. [Online]. Available at: https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/.

Jamsheer K., 2018. 12 Best Software Development Methodologies with Pros & Cons. [ONLINE]. Available at: https://acodez.in/12-best-software-development-methodologies-pros