B.TECH. (CSE)
V Semester
UE21CS341A –Software Engineering

PROJECT REPORT
on

HOSPITAL MANAGEMENT SYSTEM

Submitted by :

G.Sai Kruthi	PES2UG21CS179
Gowri Sriya R.V.S	PES2UG21CS180

Darsh Patel	PES2UG21CS150
Devanjan Baneerjee	PES2UG21CS159

August – Dec 2023
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
BENGALURU – 560100, KARNATAKA, INDIA

Table of Contents

PROJECT PROPOSAL:

PROJECT DESCRIPITION:

The Hospital Management System with Patient Health Monitoring is a comprehensive software solution designed to streamline and enhance the operations of a healthcare facility. This system focuses on efficiently managing medical records, providing patients with access to their health information, and monitoring their health in real-time. It aims to improve patient care, enhance the efficiency of hospital staff, and ensure data security and compliance with healthcare regulations.

CUSTOMER BASE

1. Hospitals and Healthcare Institutions: Seeking operational efficiency and improved patient care.
2. Healthcare Administrators: Focused on enhancing hospital operations.
3. Healthcare Providers: In need of streamlined patient management.
4. Patients and Caregivers: Looking for access to health records and monitoring.
5. Health Insurance Companies: Interested in claims processing and data accuracy.
6. Patients to view their past medical prescriptions.

END USER

1. **Healthcare Professionals (Doctors, Nurses, Staff):** These users rely on the system for efficient management of patient records, appointment scheduling, and real-time access to critical patient data. They benefit from streamlined workflows, improved data accuracy, and enhanced decision-making capabilities.
2. **Patients**: Patients gain access to their electronic health records, appointment scheduling, and medication information through a user-friendly portal. This empowers them to take an active role in their healthcare, monitor their progress, and communicate with their healthcare providers more conveniently.
3. **Caregivers:** Caregivers, often family members or guardians, use the system to monitor the health of their loved ones remotely. They receive alerts for abnormal health readings, ensuring timely interventions and peace of mind.

FUNCTIONAL REQURIMENTS:

1. Patient Registration and Profile Management: (Kruthi)
    - Develop a user-friendly registration process capturing essential patient information.

2. Electronic Health Records (EHR):(Kruthi)
    - Design a secure database to store and manage electronic health records, ensuring data privacy and compliance with healthcare regulations.

3. Patient Portal:(Gowri)
    - Create a patient portal for secure login, data access, appointment scheduling, and communication with healthcare providers.

4. Appointment Scheduling:(darsh)
    - Implement a scheduling system allowing patients to book, reschedule, and cancel appointments online. Enable notifications.

5. Prescription Management:(devanjan)
    - Develop a module for generating electronic prescriptions, integrating with pharmacies, and notifying patients for medication reminders.

QUALITATIVE PROPERTY:

A crucial qualitative property of the Hospital Management System with Patient Health Monitoring project is **Usability**. This property focuses on creating an intuitive and userfriendly system for healthcare professionals, patients, and caregivers. Usability ensures easy navigation, efficient data entry, and clear interfaces on various devices. A highly usable system promotes user satisfaction, encourages healthcare professionals to adopt it for efficient patient management, empowers patients to actively engage in their healthcare, and aids caregivers in monitoring their loved ones' well-being. Moreover, it reduces training time, enhances productivity, and contributes to the overall success of the project by making it accessible and efficient for all users.

PROJECT PLAN:

1.Identify the lifecycle to be followed for the execution of your project and justify why you have chosen the model.

In our Hospital Management System (HMS) project, we are implementing the Agile methodology as our chosen Software Development Life Cycle (SDLC) approach.

Reasons for Using Agile in HMS:

- Flexibility and Adaptability: Agile allows for changes in requirements as medical practices and regulations evolve, ensuring the system remains upto-date with the dynamic healthcare environment.
- Stakeholder Collaboration: Agile promotes continuous engagement with doctors, nurses, administrators, and patients, ensuring the system meets their specific needs through iterative feedback loops.
- Incremental Development and Quick Deployment: Agile's iterative nature enables the delivery of essential features in short cycles, allowing hospitals to implement critical functionalities rapidly and progressively improve the system.
- Effective Risk Management: Agile provides mechanisms for identifying and mitigating risks early in the development process. Through frequent inspections, adaptations, and regular retrospectives, Agile teams can address potential issues swiftly. By addressing risks proactively, Agile reduces the likelihood of project failures, ensuring that the project stays on track and delivers the desired outcomes within the defined constraints.

2.Identify the tools which u want to use it throughout the lifecycle like planning tool, design tool, version control, development tool, bug tracking, testing tool.

We intend to use planning tools like -

- Microsoft Project: This is a widely-used tool for creating Gantt charts, which help visualize project tasks and their dependencies over time.
- Tools like LeanKit can help us implement Lean principles in our project planning, emphasizing efficiency and reducing waste.
- MindMeister or XMind can help you create visual mind maps to brainstorm and plan different aspects of your project.

We plan to use design tools like -

- Figma: Figma is a collaborative design tool that allows many team members to work on this project in real time. It's a good tool for creating and iterating on designs together.
- Canva: Canva is a user-friendly online design platform that's great for creating marketing materials, social media graphics, and basic UI elements.

Coming to version control, we use -

- Git: Git is the most widely used distributed version control system. It's highly versatile and can be used with various hosting services like GitHub, GitLab, and Bitbucket.
- GitHub: GitHub is a web-based platform that provides hosting for Git repositories. It offers collaboration features and integrates with numerous development tools.

The development tools we plan to use are -

➔ Integrated Development Environments (IDEs):

- Visual Studio Code: A versatile and widely-used code editor that supports multiple programming languages and has a vast ecosystem of extensions.

➔ Backend Frameworks:

- My SQL Connector: A MySQL connector is a software component that allows a programming language to communicate with a MySQL database. It provides an interface for the programming language to send queries and receive results from the MySQL database

➔ Frontend Frameworks:

- Streamlit: Streamlit is an open-source Python library that is used to create web applications for data exploration and visualization with minimal effort. It is designed to make it easy for data scientists and developers to turn data scripts into shareable web apps.

➔ Database tools:

- My SQL: MySQL is an open-source relational database management system (RDBMS). It is widely used for managing and manipulating structured data. MySQL uses SQL (Structured Query Language) to interact with the database, making it a popular choice for many web applications.

The testing tool we are going to use is -

➔ Manual testing: Manual testing involves human testers executing test cases without the aid of automation tools to ensure software functionality and identify potential issues. Testers manually explore and validate various aspects of the software, mimicking user interactions for comprehensive quality assurance."

3. Deliverables for Hospital Management System (HMS) Project:

1. Reusable Components:
    - User Authentication Module:
    - Description: Provides secure user authentication, allowing various users (doctors, nurses, administrators) to access the system securely.
    - Scenario: This module can be applied in multiple projects and healthcare applications to ensure secure access control, making it a reusable component.

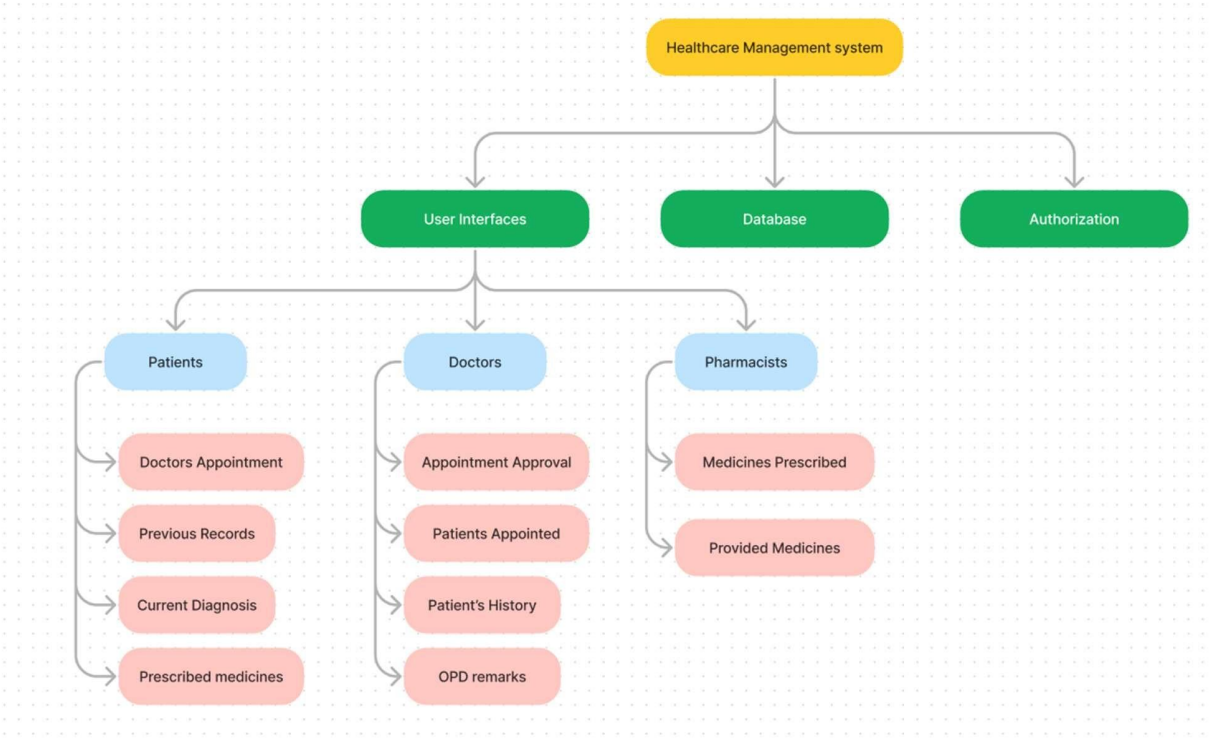
Reusable components streamline development, offering consistent, tested solutions that enhance efficiency. They ensure standardized functionalities, reducing redundancy and development time. By promoting modularity, they enhance system maintainability and facilitate seamless integration across projects, fostering a more agile and cost-effective development process.

2. Build Components:

- Patient Information Management:
- Description: Manages patient details, medical history, prescriptions, and treatment plans.
- Justification: Patient information structures and workflows often vary significantly between hospitals. Each hospital requires a customized patient management system tailored to its specific needs.
- Appointment Scheduling Module:
- Description: Facilitates the scheduling of patient appointments with doctors, tests, and other healthcare services.
- Justification: Scheduling rules, workflows, and policies are unique to each hospital. Customization is necessary to accommodate diverse scheduling requirements.

Build components are essential for tailoring software to specific project requirements, ensuring precise functionality aligned with organizational needs. They enable customization, accommodating unique workflows and compliance standards. These components provide flexibility, allowing seamless adaptation to evolving project demands. By focusing on project-specific needs, build components enhance system efficiency and user satisfaction, ultimately delivering a tailored and optimal software solution.

4.WBS



5: Do a rough estimate of effort required to accomplish each task in terms of person months.

Effort estimation:

Effort estimation is a critical aspect of project planning as it helps you allocate resources and set realistic timelines. Here's a more detailed effort estimation for the tasks involved in developing a Hospital Management system :

Our project, estimated at 5 KLOC (including git commands), is managed using the organic COCOMO model strategy, which is well-suited for our small-sized team. Since, our project is going to fall in organic category value of constants will be :- a= 2.4, b=1.05, KLOC= 5

Effort = ( ) = 2.4 (4.5) · = 12 PM (person months)

Task 1: Interface Design

KLOC: 1.2

Effort: $2.4(1.2)^{\cdot}$ = 2.91 ≈ 3

Time: $2.5(2.91)^{\cdot}$ = 3.75

Task 2: Database Setup

KLOC: 1

Effort: $2.4(1)^{\cdot}$ = 2.4 ≈ 2.5

Time: $2.5(2.4)^{\cdot}$ = 3.49

Task3: Authentication Development

KLOC: 1

Effort: $2.4(1)^{\cdot}$ = 2.4 ≈ 2.5

Time: $2.5(2.4)^{\cdot}$ = 3.49

Task4: Patient functionality, doctor functionality, pharmacist functionality

KLOC: 1.2

Effort: $2.4(1.2)^{\cdot}$ = 2.91 ≈ 3

Time: $2.5(2.91)^{\cdot}$ = 3.75

Task 5: Git and GitHub (These would be used alongside other tasks)

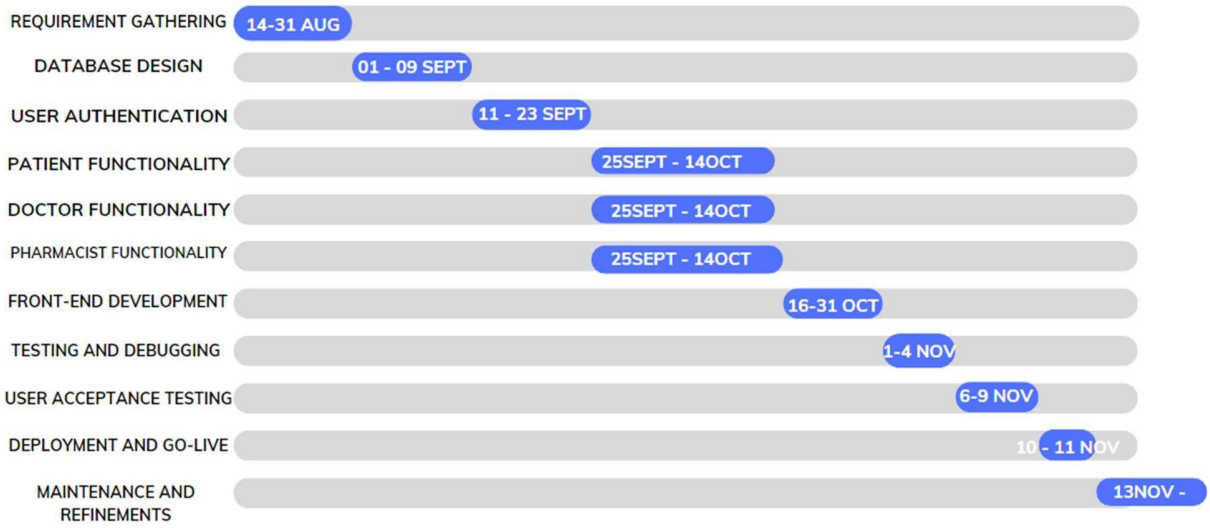
KLOC: 0.3

Effort: $2.4(0.3)^{\cdot}$ = 0.7

Time: $2.5(0.7)^{\cdot}$ = 2.1

6: Create the Gantt Chart for scheduling using any tool.

GANTT CHART


REQUIREMENT GATHERING
14-31 AUG
DATABASE DESIGN
01 - 09 SEPT
USER AUTHENTICATION
11 - 23 SEPT
PATIENT FUNCTIONALITY
25SEPT - 14OCT
DOCTOR FUNCTIONALITY
25SEPT - 14OCT
PHARMACIST FUNCTIONALITY
25SEPT - 14OCT
FRONT-END DEVELOPMENT
16-31 OCT
TESTING AND DEBUGGING
1-4 NOV
USER ACCEPTANCE TESTING
6-9 NOV
DEPLOYMENT AND GO-LIVE
10 - 11 NOV
MAINTENANCE AND REFINEMENTS
13NOV -

SRS

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The Software Requirements Specification (SRS) for a Hospital Management System serves as a comprehensive document outlining the functional and non-functional requirements of the system. By detailing user interactions, data management, security protocols, and performance expectations, the SRS ensures a unified understanding among project team members, enabling the successful design, development, and implementation of an efficient and secure hospital management solution.

1.2 Intended Audience

Intended audience of the Software Requirements Specification (SRS) for a Hospital Management System includes developers, project managers, and stakeholders involved in the system's design, development, and implementation. It provides a detailed blueprint for the technical team while offering a clear understanding of the system's functionality and requirements to non-technical stakeholders.

1.3 Product Scope

The product scope of a Hospital Management System encompasses a comprehensive software solution designed to streamline and optimise various healthcare operations within a hospital or medical facility. It includes modules for patient registration, appointment scheduling, electronic health records (EHR) management, pharmacy management, and reporting. The system aims to enhance efficiency, improve patient care, reduce errors, and facilitate better decision-making for healthcare professionals and administrators within the hospital environment.

The primary goal of a Hospital Management System is to improve the overall efficiency and quality of healthcare services within a medical facility. By automating and integrating various

administrative, clinical, and financial tasks, the system aims to enhance patient care, minimise errors, reduce operational costs, streamline workflow processes, ensure accurate record-keeping, and provide healthcare professionals with timely access to relevant patient information. Ultimately, the product seeks to optimize hospital operations, leading to better patient outcomes and satisfaction while supporting effective decision-making for healthcare providers and administrators.

2. Overall Description

2.1 Product Perspective

The product perspective of a Hospital Management System involves considering the software as a part of a larger healthcare ecosystem. It interfaces with other systems such as laboratory information systems, pharmacy management systems, and electronic health record systems. From a broader perspective, it operates within the context of hospital operations, interacting with staff, patients, and external entities like insurance providers.

2.2 Product Functions

1) For patients –

- Patient Registration
- Appointment Scheduling
- Electronic Health Records (EHR)
- Prescription Management

2.3 User Classes and Characteristics

***Doctors**:*

Characteristics:

Medical Expertise: Doctors are highly trained medical professionals with expertise in diagnosing and treating various medical conditions.

Access to Patient Data: They require access to patient electronic health records (EHR) to make informed medical decisions.

Prescription Management: Doctors need to create and manage electronic prescriptions, specifying medications, dosages, and treatment plans.

Appointment Scheduling: They schedule appointments with patients for examinations, consultations, and follow-ups.

Critical Decision-Making: Doctors rely on the system to provide accurate and up-to-date patient information for critical medical decisions.

Communication: They may use the system for secure communication with patients and healthcare staff.

Monitoring: In some cases, doctors may need access to patient health data, such as vital signs and lab results, for monitoring.

***Patients**:*

Characteristics:

Health Information Access: Patients need access to their electronic health records (EHR) to review their medical history, prescriptions, and lab reports.

Appointment Management: Patients use the system to schedule, reschedule, or cancel appointments with healthcare providers.

Medication Management: They rely on the system for managing prescribed medications, including reminders for taking them.

Health Monitoring: Some patients may have IoT devices integrated with the system to monitor their health, and they need access to this data.

Active Engagement: Patients actively engage in their healthcare by reviewing their health information, treatment plans, and progress.

***Pharmacists**:*

Characteristics:

Prescription Verification: Pharmacists use the system to verify electronic prescriptions received from doctors.

Prescription Fulfilment: Pharmacists update prescription statuses, indicating when medications are ready for patients to pick up.

Patient Verification: Pharmacists verify patient identities before dispensing medications, ensuring patient safety and compliance.

2.4 Operating Environment

The software will be a web-based application mainly built to work on PC's.

2.5 Implementation Constraints

- *Product should be ready by the second week of November*
- *Some implementation constraints include regulatory compliance, data security and privacy requirements, interoperability challenges, scalability considerations, legacy system integration, limited budget, staff training, technical expertise, customization demands, change management, testing and validation requirements, hardware and infrastructure limitations, user feedback and iteration challenges, vendor lock-in risks, and geographic and cultural variations. Successfully addressing these constraints necessitates careful planning, collaboration with stakeholders, and a strategic approach to mitigate risks during the Hospital management system implementation process.*

2.6 Assumptions and Dependencies

- *Dependencies used for the product will be maintained for the lifetime of the product.*
- *The scale of the product will remain the same.*
- *Assumptions include accurately gathered user requirements, a skilled development team, the right technology stack, a realistic project timeline, adequate budget and resources, legal compliance, and robust data privacy and security measures. In addition, to be specific the users need to have a basic knowledge about the environment they interact with.*

Dependencies encompass integration with external data sources and existing hospital systems, the availability of testing environments and a reliable database management system, the quality of hardware infrastructure, user training, ongoing maintenance and support, scalability considerations, user feedback, change management, third-party services, and budget allocation. In addition to be specific, availability of internet connection for the users, database to develop software.

3.0 External Interface Requirements

3.1 User Interfaces

The Hospital Management System's user interface is a critical component of the system's success. It should aim to be user-friendly, visually appealing, and efficient, catering to the needs of various user categories. Modern design principles with a clean and responsive layout should be followed.

User Categories and Interfaces:

1. Doctors Interface:

Dashboard:
Upon logging in, doctors are greeted with a customizable dashboard.
Key patient metrics are displayed, including appointment schedules, critical health alerts, and recent patient interactions.

Patient Profiles:
Doctors can access detailed patient profiles, organized by name or medical record number. Each profile provides a comprehensive view of the patient's electronic health records, including medical history, current medications, and treatment plans.

Appointment Scheduler:
Doctors can efficiently manage appointments using an interactive calendar view.
They can schedule, reschedule, or cancel appointments, with real-time notifications sent to patients.

Prescription Management:
A dedicated section allows doctors to create electronic prescriptions quickly.
Doctors can prescribe medications, specify dosages, and set medication reminders for patients.

2. Patients Interface:

Personal Health Dashboard:
Patients are welcomed by a user-friendly health dashboard upon login.
The dashboard provides a summary of vital health statistics, such as heart rate, blood pressure, and recent lab results.

Electronic Health Records (EHR):

Patients can access their comprehensive electronic health records.
They can view medical history, prescription details, and lab reports, promoting active participation in their healthcare.

Appointment Scheduler:
Patients can schedule appointments with their healthcare providers, including doctors and specialists.
The system offers real-time availability and confirmation notifications.

Medication Management:
Patients can review their prescribed medications, dosages, and schedules.
Medication reminders are provided to ensure adherence to treatment plans.

3. Pharmacy Interface:

Prescription Verification:
Pharmacists access the system to verify electronic prescriptions from doctors.
They can review prescription details, check for potential interactions, and prepare medications accordingly.

Prescription Fulfillment:
Pharmacists can update prescription status, marking them as filled or ready for pickup.
Patients receive notifications when their prescriptions are ready.

Patient Verification:
Pharmacists verify patient identity before dispensing medications to ensure patient safety and compliance.

3.2 Software Interfaces

Different Graphical User Interfaces will be created for Patients, Doctors and Pharmacists.

Doctors:
1. Dashboard
2. Search Patient info page
3. OPD page with option to prescriptions
4. Appointment page

Patients:
1. Dashboard with patient info

2. Prescriptions page
3. Doctors Remarks page

Pharmacists

1. Dashboards with prescriptions by doctors to patients

The following are the tools and libraries used for the front-end of the software:

1. Streamlit

The following are the tools and libraries used for the back-end of the software:

1. *My SQL*
2. *My SQL Connector*

3.3 Communications Interfaces

Protocol: HTTP (Hypertext Transfer Protocol) will be employed as the underlying communication protocol for the RESTful API.

RESTful API: The API will follow REST (Representational State Transfer) principles, emphasizing stateless communication through standard HTTP methods like GET, POST, PUT, and DELETE.

Front-end Consumption: The front-end interfaces, which include dashboards, prescription pages, search functionalities, and appointment management, will consume data and services via this RESTful API.

Interactions: The communication interfaces will facilitate data exchange between users (Patients, Doctors, Pharmacists) and the system's core functionalities, such as patient information retrieval, prescription management, appointment scheduling, and more.

Security: Appropriate security measures, such as authentication and authorization mechanisms, may be implemented to ensure secure data exchange between the front-end and back-end components.

4.0 Analysis Models

Create Table:-

Collection: Doctors

_id: ObjectId (automatically generated unique identifier) firstName: String lastName: String specialization: String contact: Object email: String phone: String appointments: Array of Objects patient: String (references the patient's _id) date: Date status: String prescriptions: Array of Objects patient: String (references the patient's _id) medicineName: String dosage: String datePrescribed: Date

Collection: Patients

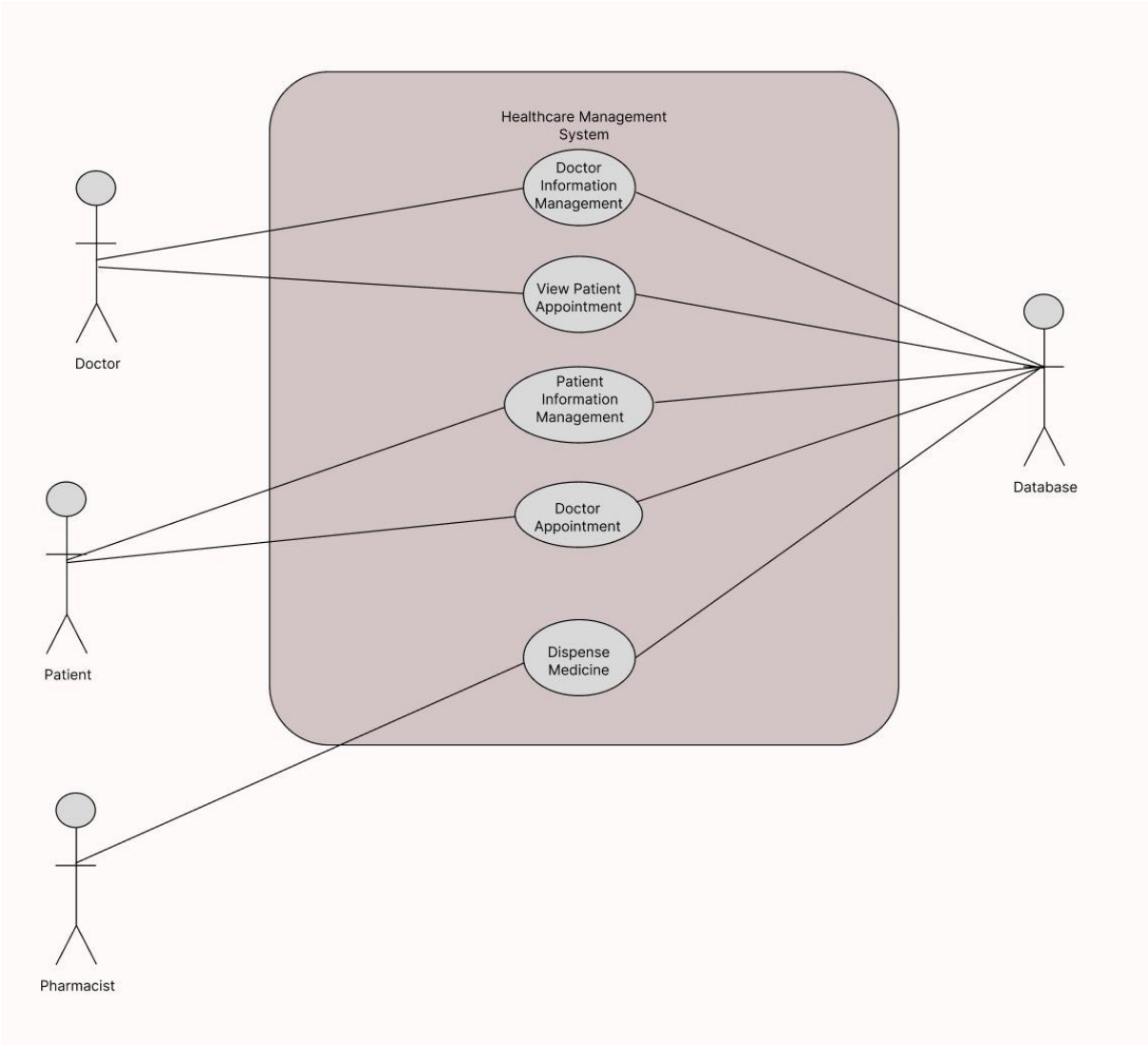
_id: ObjectId (automatically generated unique identifier) firstName: String lastName: String dob: Date contact: Object email: String phone: String healthData: Object bloodType: String allergies: Array of Strings conditions: Array of Objects conditionName: String diagnosisDate: Date appointments: Array of Objects doctor: String (references the doctor's _id) date: Date status: String prescriptions: Array of Objects medicineName: String dosage: String

datePrescribed: Date

Collection: Pharmacists

_id: ObjectId (automatically generated unique identifier) firstName: String lastName: String contact: Object email: String phone: String dispensedMedicines: Array of Objects patient: String (references the patient's _id) medicineName: String dosage: String dateDispensed: Date

USE CASE DIAGRAM:



5. System Functional Requirements

System Feature 1: Patient Appointment Management

5.1.1 Description and Priority

This feature allows patients to book appointments with doctors. It is of High Priority.

5.1.2 Stimulus/Response Sequences

Stimulus: A patient selects a doctor, chooses a preferred date and time, and requests an appointment.
Response: The system checks the doctor's availability and sends a confirmation or rejection notification to the patient.

5.1.3 Functional Requirements

REQ-1: The system must display a list of available doctors.
REQ-2: Patients should be able to select their preferred doctor and appointment time.
REQ-3: The system must check the doctor's schedule for availability.
REQ-4: If the doctor is available, the system should send a confirmation to the patient. REQ-5: If the doctor is not available, the system should send a rejection notification to the patient.

System Feature 2: Doctor Review and Prescription Management

5.2.1 Description and Priority

This feature enables doctors to provide reviews and prescriptions for patient appointments. It is of High Priority.

5.2.2 Stimulus/Response Sequences

Stimulus: A doctor selects a patient's appointment, provides a review, and prescribes medications.
Response: The system stores the review and prescription, making them accessible to the patient.

5.2.3 Functional Requirements

REQ-1: The system should provide doctors with a list of appointments.
REQ-2: Doctors must be able to select an appointment and input a review.
REQ-3: Doctors should be able to prescribe medications.
REQ-4: The system must store the review and prescription information securely. REQ-5: Patients should be able to access their reviews and prescriptions.

System Feature 3: Pharmacist Medication Dispensation

5.3.1 Description and Priority

This feature enables pharmacists to dispense medications to patients based on prescriptions. It is of High Priority.

5.3.2 Stimulus/Response Sequences

Stimulus: A pharmacist selects a patient's prescription and provides the prescribed medications.
Response: The system records the dispensation and updates medication inventory.

5.3.3 Functional Requirements

REQ-1: The system should provide pharmacists with a list of prescriptions to be filled. REQ-2: Pharmacists must be able to select a prescription and dispense the prescribed medications.
REQ-3: The system should deduct dispensed medications from the inventory.
REQ-4: The system must keep a record of dispensed medications for tracking and auditing.
System Feature 4: User Account Management

5.4.1 Description and Priority

This feature allows users to create their accounts. It is of High Priority.

5.4.2 Stimulus/Response Sequences

Stimulus: A user registers for a new account or logs in.
Response: The system processes account-related actions.

5.4.3 Functional Requirements

REQ-1: Users should be able to create new accounts with their details. REQ-2: Users must log in with their credentials securely.

6. Other Nonfunctional Requirements

6.1 Performance Requirements

The software shall be able to support and available to multiple users at the same time. It will also be efficient and systematic. The below performance requirements shall be taken care of in the process.

1) Database query execution times shall be within 5 seconds.

2) We will be defining acceptable response times for various system operations, such as patient registration, appointment scheduling, and report generation within 5 secs.

3) Good throughput will be provided for the number of transactions or operations the system should support 50 users per unit of time.

6.2 Security Requirements

1) Define user roles and permissions to restrict access to patient records and sensitive system functionality.

6.3 Reliability Requirements

1) We define the required uptime for the system, including scheduled maintenance periods.
2) Ensuring fault tolerance.
3) Ensuring data integrity.

6.4 Usability Requirements

We ensure the user interface is intuitive, user-friendly, and accessible to all patients and doctors.

6.5 Domain requirements

The user can perform the following functions -

- Login and authenticate
- Check previous medical records and prescriptions

- Track the ongoing treatments

The doctors can perform the following functions -

- Verify patients and authenticate
- Perform CRUD operations on patient database
- Perform CRUD operations on past records database
- Send data to verify information
- View information and offer suggestions

6.6 Software Quality Attributes

Usability: The system will be user-friendly and intuitive, making it easy for hospital staff to navigate and use efficiently. Usability testing and user feedback are critical.
Reliability: The HMS must operate reliably without unexpected crashes or errors. It will be available 24/7 to support critical healthcare operations.

Performance: The system should deliver high performance to handle a large volume of patient records, appointments, and transactions efficiently. Response times should be minimal.

Scalability: The HMS will be designed to scale easily to accommodate the growing needs of the hospital, including an increasing number of patients and data.

Security: Robust security measures, including authentication, authorization, and data encryption, will be present to protect patient data and comply with healthcare data privacy regulations.

Data Accuracy: The system should ensure data accuracy and consistency across all modules, including patient records, prescriptions, and billing information.

6.7 Business rules

Patient Registration: Authorise personnel to register patients with accurate information.
Appointment Scheduling: Follow guidelines for scheduling appointments based on availability, priority, and patient preferences.
Prescription Management: Ensure proper medication prescriptions following dosage and safety guidelines.
Access Control: Implement strict access controls for data privacy and security.
Emergency Protocols: Establish clear procedures for medical emergencies.
Patient Discharge: Define rules for discharge summaries and follow-up appointments. Inventory Management: Follow procedures to manage medical supplies efficiently.

Appendix A: Glossary

1. *API -> Application Program Interface*
2. *HMS -> Hospital Management System*
3. *CRUD -> Create, Read, Update, Delete*
4. *EHR -> Electronic Health Records*
5. *GUI -> Graphical User Interface*
6. *ID -> Identifier*
7. *PC -> Personal Computer*
8. *SRS -> Software Requirement Specifications*
9. *OPD -> Out patient department*
10. *UI -> User Interface*

Appendix B: Field Layouts

Sheet with information required to register the Doctors

Field	Length	Data Type	Description	Is Mandatory

Doctor Name Specialization Date Email Contact Number Appointments Patient name MedicineName Dosage	20 10 10 30 10 20 30 20 15	String String String Email String Array of objects String String String	Name of the doctor Specialization of doc date Doctor's mail address Doctor contact no. List of appointments Info of patient Name of medicine Dosage of medicine	Y Y Y Y Y N Y Y Y

Sheet with information required to register the patients

Field	**Length**	**Data Type**	**Description**	**Is Mandatory**
Name Phone Number Email Healthdata	20 10 30 10	String String String String	Name of the patient Patient's phone number Email of the patient Patient's health details	Y N Y Y

Diagnosis date	10	Date	Date of diagnosis	Y
Doctor Info	20	String	Info of doctors	Y
Condition name	20	String	Name of condition	Y
Medicine name	20	String	Name of medicine	Y
dosage	20	String	Dosage of medicine	Y

Sheet with information required to register the pharmacist

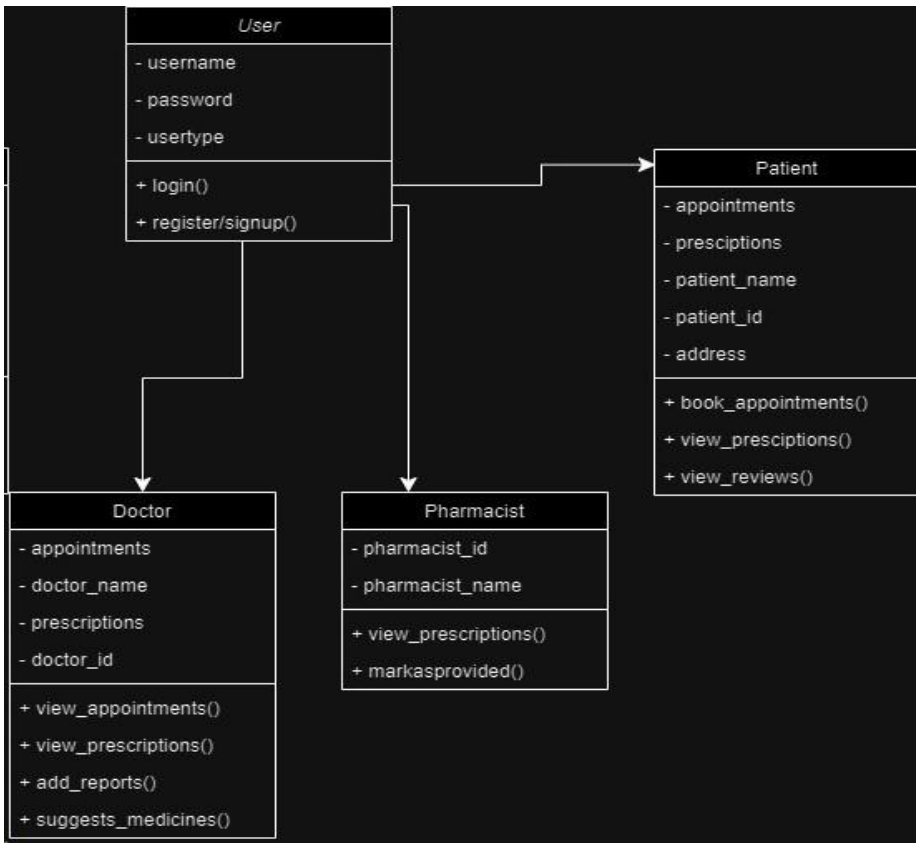
Field	**Length**	**Data Type**	**Description**	**Is Mandatory**
Name	20	String	Name of the Pharmacist	Y
Phone Number	10	String	Phone number	Y
Address	40	String	Pharmacist's Address	N
Email	20	String	Email address	N
Dispensed medicines	40	String	List of medicines	Y
Patient info	20	String	Info about patient	Y

Appendix C: Requirement Traceability Matrix

Sl. No	**Requirement ID**	**Brief Description of Requirement**	**Architecture Reference**	**Design Reference**	**Code File Reference**	**Test Case ID**	**System Test Case ID**
1	REQ-001	User Login	AR-1	DR-2	CFR-2	UA_TC_02	UA_TC_03

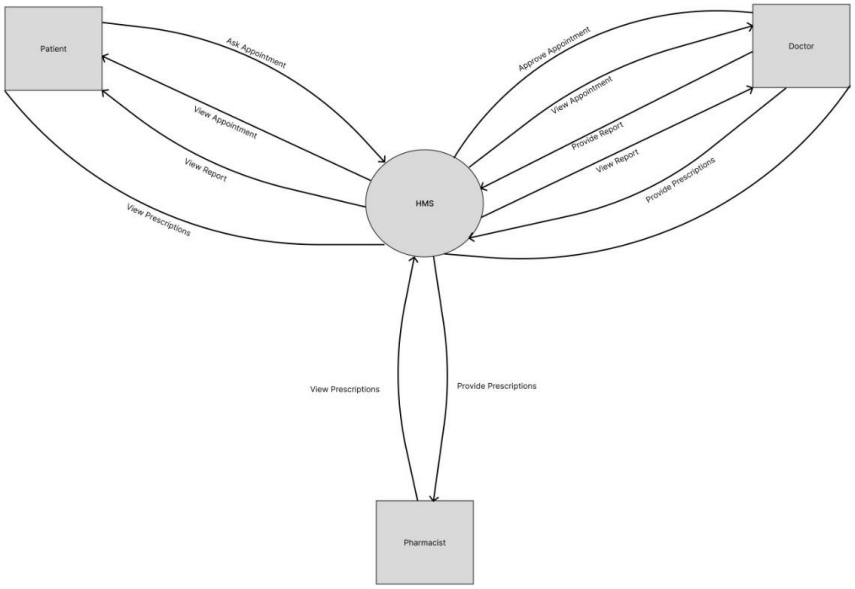
UA_	REQ-002	Doctor dashboard	AR-2	DR-3	CFR-3	DF_01	DF_02
3	REQ-003	Patient dashboard	AR-3	DR-4	CFR-4	PF_01	PF_03
4	REQ-004	Pharmacist dashboard	AR-4	DR-5	CFR-5	PFH_01	PHF_05
5	REQ-005	REPORT GENERATION	AR-5	DR-5	CFR-5	REP_01	REP_03

LLD 1. CLASS DIAGRAM

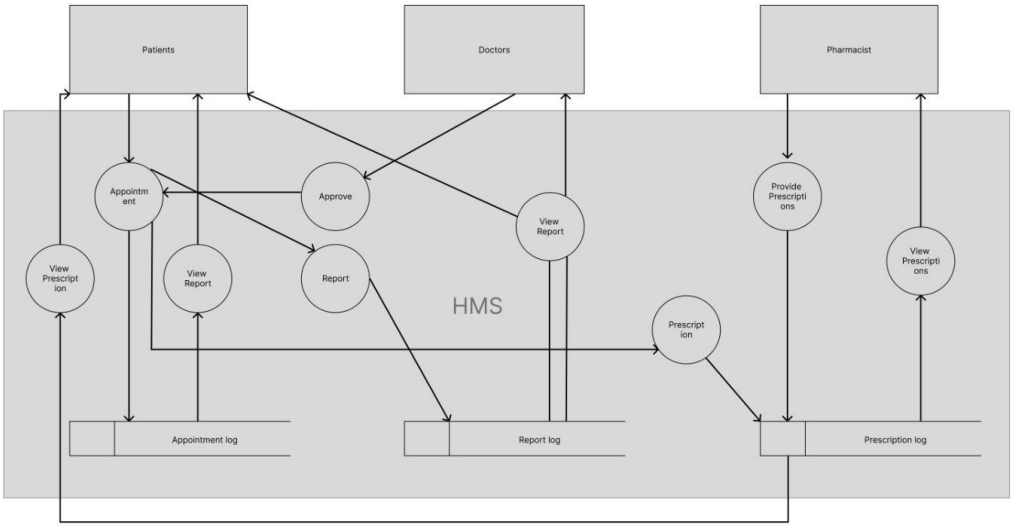


2. DFD

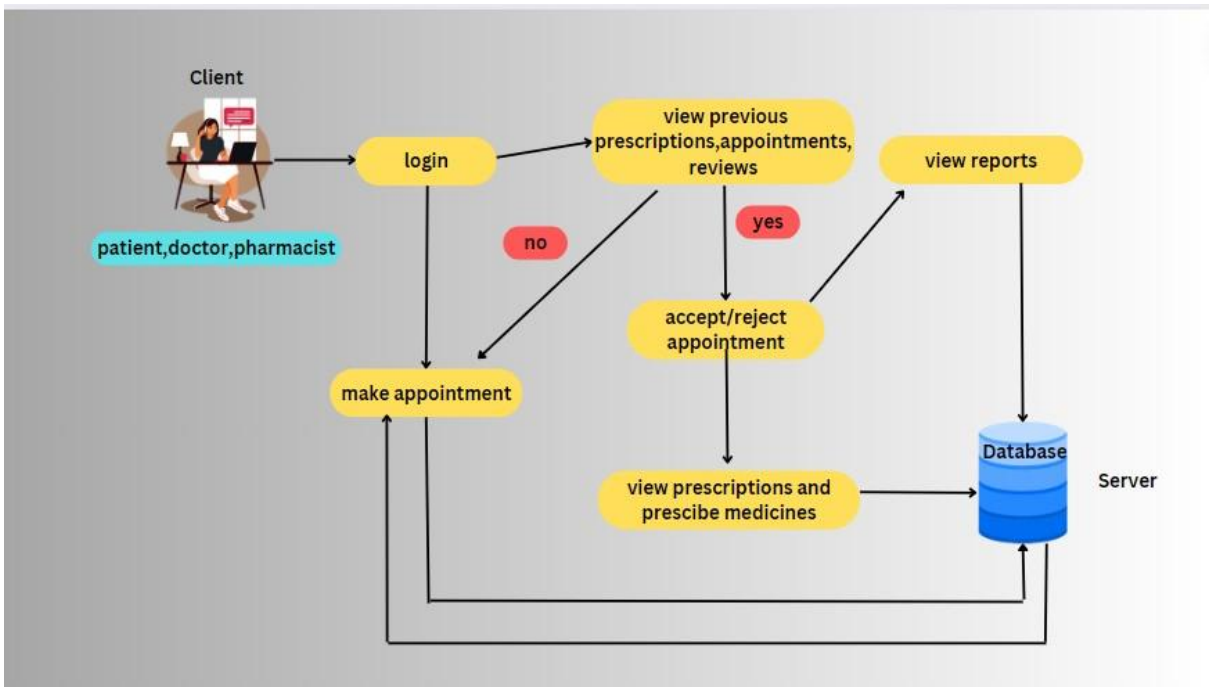
a. 0 level DFD



b. 1 level DFD



3. Architectural Style



For a Hospital Management System implemented using the MySQL the appropriate architecture style would be the Client-Server architecture

Justification:

Scalability: Client-server architecture allows you to scale different components independently. In your case, you may have multiple clients (patients, doctors, and pharmacists) connecting to a central server. As your system grows, you can add more client devices or servers as needed without requiring a complete overhaul of your system.

Centralized Data Management: With a client-server model, you can centralize your data storage and management. Patient information, appointments, prescriptions, and reviews can be stored in a central database, making it easy to access, update, and maintain the data. This centralized approach ensures data consistency and reduces data redundancy.

Security: Client-server architectures typically offer more robust security measures. You can implement user authentication and authorization mechanisms on the server to ensure that only authorized users can access and modify sensitive data. Additionally, you can implement encryption for data transmission between clients and the server to protect sensitive patient information.

Load Distribution: Client-server architectures enable load distribution across multiple servers, which can improve the overall system's performance. For example, you can have a separate server dedicated to handling appointment scheduling and another

server for managing prescription data. This load distribution helps prevent bottlenecks and ensures responsive performance for all users.

Multi-Platform Compatibility: Client-server architectures are platform-agnostic, meaning that clients can connect to the server from different devices and operating systems. This is especially important in a healthcare setting, where doctors and pharmacists may use various devices to access the system.

Centralized Updates and Maintenance: When you need to update or maintain the system, you can do so on the server, and all clients will automatically benefit from these changes. This simplifies the process of deploying updates and patches, ensuring that the entire system remains consistent and up to date.

Separation of Concerns: The client-server architecture allows you to separate the presentation layer (the client) from the business logic and data management (the server). This separation makes it easier to maintain and enhance different components of your system independently.

Reliable Communication: In a client-server model, communication between clients and the server is well-defined and reliable. You can use standard communication protocols (e.g., HTTP/HTTPS) to ensure that data is transmitted securely and efficiently.

In summary, a client-server architecture is a sound choice for your hospital management system because it provides scalability, centralized data management, security, load distribution, multi-platform compatibility, centralized updates and maintenance, separation of concerns, and reliable communication. These advantages will help you create a robust and efficient system that can meet the needs of patients, doctors, and pharmacists while ensuring data integrity and security.

PES UNIVERSITY, BANGALORE
Department of Computer Science and Engineering
B. Tech (CSE) – 5th Semester – Aug-Dec 2023

UE21CS341A - Software Engineering
Test Plan Documentation
Team 10:

PES2UG21CS150	Darsh Patel
PES2UG21CS159	Devanjan Banerjee

PES2UG21CS180	Gowri Sriya
PES2UG21CS179	Sai Kruthi Gollaputi

User Login Test cases:

Test Case ID	Name of Module	Test Case Description	Pre-conditions	Test Steps	Test Data	Expected Results	Actual Results	Test Result
UA_TC_01	User Authentication	Verify that a user can successfully log in with valid credentials.	User account exists with valid login credentials.	1)Navigate to the login page. 2)Enter valid username and password. 3)Click on the login button.	Valid username and password	The user is redirected to their dashboard or the homepage.	The user is successfully logged in, and the expected page is displayed.	Pass

UA_TC_02	User Login - Invalid Credentials	Verify that a user cannot log in with invalid credentials.	User account exists with invalid login credentials.	1)Navigate to the login page. 2)Enter invalid username or password. 3)Click on the login button.	Invalid usernameor password	User receives an error message indicating invalid credentials.	The system correctly displays an error message for the invalid username.	Pass
UA_TC_03	User Logout	Verify that a user can successfully log out.	User is logged in.	Click on the logout button or link.	N/A	User is redirected to the login page.	User is redirected to the login page successfully.	Pass
UA_TC_04	User login	Verify that the system handles an invalid email format during login.	User is on the login page.	Enter an email with an incorrect format (e.g., missing '@' or domain). Enter a valid password. Click on the login button.	Invalid email format.	The system displays an error message indicating that the email format is invalid.	The system displays an error message indicating that the email format is invalid and tells user to enter valid email.	Pass
UA_TC_05	User Authentication	Verify that the application doesn't crash during login.	User is on the login page.	1)Check if successfully able to login to home page. 2)Check for seamless operation.	N/A	System works correctly and user doesn't face any crash/ failure issues during login	System works smoothly and user doesn't face any crash/ failure issues during login	Pass
UA_TC_06	User authentication	Verify that the system doesn't lag between the interfaces.	User switches between login page and different interfaces.	1)Enter valid login details for all different types of users.	N/A	System works accurately and very less lag between switching interfaces.	System works accurately and very less lag between switching interfaces.	Pass

Patient Functionality Test Cases:

Test Case ID	Name of Module	Test Case Description	Pre-conditions	Test Steps	Test Data	Expected Results	Actual Results	Test Result
PF_01	Patient Functionality	Verify that a patient can successfully book an appointment with an available doctor.	Patient is logged in, and there are available appointment slots.	1)Navigate to the "Book Appointment" section. 2)Select a doctor, date, and time preferences. 3)Click on the "Book Appointment" button.	Valid doctor selection, date, and time preferences.	The patient receives a confirmation message, and the appointment is added to the upcoming appointments list.	The patient receives a confirmation message, and the appointment is added to the upcoming appointments list.	Pass
PF_02	Patient Functionality	Verify that a patient cannot book an appointment with invalid date/time preferences.	Patient is logged in, and there are available appointment slots.	1)Navigate to the "Book Appointment" section. 2)Select a doctor and choose an invalid date or time. 3)Attempt to book the appointment.	Invalid date or time selection.	The system displays an error message, and the appointment is not booked.	An error message is displayed, indicating that appointment is not booked	Pass
PF_03	Patient Functionality	Verify that a patient can view a list of their upcoming appointments.	Patient has upcoming appointments.	1)Navigate to the "View Appointments" section.	All patient's appointments	The system displays a list of upcoming appointments for the patient.	The system displays a list of upcoming appointments for the patient.	Pass
PF_04	Patient Functionality	Verify that a patient can view a list of their past appointments.	Patient has past appointments.	1)Navigate to the "View Appointments" section. 2)Select the "Past Appointments" tab.	N/A	The system displays a list of past appointments for the patient.	The system displays a list of past appointments for the patient successfully.	Pass
PF_05	Patient Functionality	Verify that a patient can view prescriptions and doctor reviews from previous visits.	Patient has previous appointments with prescriptions and reviews.	1)Navigate to the "View Prescriptions and Reviews" section. 2)Select the "Previous Visits" tab.	N/A	The system displays a list of previous visits with associated prescriptions and reviews.	The system presents a list of previous visits with associated prescriptions and reviews.	Pass
PF_06	Patient Functionality	Verify that a patient receives a message when there are no prescriptions or reviews for previous visits.	Patient has no previous appointments.	1)Navigate to the "View Prescriptions and Reviews" section.	N/A	The system displays a message indicating that there are no prescriptions or reviews for previous visits.	The system displays a message indicating that there are no prescriptions or reviews for previous visits.	Pass

Doctor Functionality test cases:

Test Case ID	Name of Module	Test Case Description	Pre-conditions	Test Steps	Test Data	Expected Results	Actual Results	Test Result
DF_01	Doctor Functionality	Verify that a doctor can successfully accept a patient's appointment request.	Doctor is logged in, and there is a pending appointment request.	1)Navigate to the "Accept/Reject Appointments " section. 2)Select a pending appointment. 3)Click on the "Accept" button.	Valid pending appointment.	The system updates the appointment status to "Accepted," and the patient is notified.	The patient is notified about the status	Pass
DF_02	Doctor Functionality	Verify that a doctor can reject a patient's appointment request and provide a reason.	Doctor is logged in, and there is a pending appointment request.	1)Navigate to the "Accept/Reject Appointments " section. 2)Select a pending appointment. 3)Click on the "Reject" button and provide a reason.	Valid pending appointment, rejection reason.	The system updates the appointment status to "Rejected," and the patient is notified with the provided reason.	The system correctlyupdates the appointment status to "Rejected," and the patient is accordingly notified.	Pass
DF_03	Doctor Functionality	Verify that a doctor can access a patient's profile, including medical history.	Doctor is logged in, and there is a patient with medical history.	1)Navigate to the "View Patient Information" section. 2)Select a patient. 3)Review the patient's medical history.	Valid patient with medical history.	The system displays the patient's profile with relevant medical history information.	The system correctly displays the patient's profile with relevant medical history information.	Pass
DF_04	Doctor Functionality	Verify that a doctor can view past prescriptions and reviews given to a patient.	Doctor is logged in, and there is a patient with past prescriptions and reviews.	1)Navigate to the "View Patient Information" section. 2)Select a patient. 3)Access the "Past Prescriptions and Reviews" tab.	Valid patient with past prescriptions and reviews.	The system displays a list of past prescriptions and reviews given to the selected patient.	The system rightly displays a list of past prescriptions and reviews given to the selected patient.	Pass
DF_05	Doctor Functionality	Verify that a doctor can successfully view the	Doctor is logged in, and there is patient	1)Navigate to the "OPD “ section 2)Select a	Valid patient with a history of discharge.	The system updates the patient's records with	The system updates the patient's records with	Pass

		OPD details.	ready to be discharged.	patient with visit id.		Discharge details.	Discharge details.	
DF_06	Doctor Functionality	Verify that a doctor can successfully enter the OPD details for a patient.	Doctor is logged in, and there is patient ready to be discharged but not given approval yet.	1)Navigate to the "OPD " section 2)Select a patient with visit id. 3) Enter review, diagnose, prescription details.	Valid patient with a history of discharge.	The system updates the patient's records with Discharge details and prepares them for next discharge.	The system updates the patient's records with Discharge details and prepares them for next discharge.	Pass

Pharmacist Functionality Test cases:

Test Case ID	Name of Module	Test Case Description	Pre-conditions	Test Steps	Test Data	Expected Results	Actual Results	Test Result
PFH_01	Pharmacist Functionality	Verify that a pharmacist can successfully view a list of all medicines.	Pharmacist is logged in, and there are medications in the system.	Navigate to the "View medications to be provided" section.	Past medications and reviews.	The system displays a list of all medications need to be provided by the pharmacist.	The system displays a list of all medications need to be provided by the pharmacist successfully.	Pass
PFH_02	Pharmacist Functionality	Verify that a pharmacist receives a message when there are no medications to provide in the system.	Pharmacist is logged in, and there are no medications in the system.	Navigate to the "View medications" section.	No medications available.	The system displays a message indicating that there are no medications available.	The system displays a message indicating that there are no medications available	Pass
PFH_03	Pharmacist Functionality	Verify that a pharmacist can successfully mark a medication as provided.	Pharmacist is logged in, and there is a medication with a status indicating it hasn't been provided.	1)Navigate to the "View medications" section. 2)Select a medication with the status "Not Provided." 3)Mark the medication as provided.	Medication with a status indicating it hasn't been provided.	The system updates the medication status to "Provided."	The system updates the medication status to "Provided" accordingly.	Pass
PFH_04	Pharmacist Functionality	Verify that a pharmacist cannot mark a medication as provided if it is already marked as provided.	Pharmacist is logged in, and there is a medication with the status "Provided."	1)Navigate to the "View medications" section. 2)Select a medication with the status "Provided." 3)Attempt to	Medication with the status "Provided."	The system displays an error message.	The system displays an error message, and the medication status remains "Provided."	Pass

				mark the medication as provided again.				
PFH_05	Pharmacist Functionality	Verify that a pharmacist cannot mark an invalid or non-existent medication as provided.	Pharmacist is logged in.	1)Navigate to the "View medications" section. 2)Attempt to mark a non-existent or invalid medication as provided.	Invalid or non-existent medication .	The system displays an error message.	The system displays an error message, and no changes are made to the medication status.	Pass
PFH_06	Pharmacist Functionality	Verify that a pharmacist successfully login after user login and identify as pharmacist	Pharmacist is logged in.	1)Navigate to pharmacist home page after user login and see medications list.	N/A	The system successfully enables login to pharmacist home page.	The system successfully enables login to pharmacist home page without throwing any errors.	Pass

Reporting Test Cases:

Test Case ID	Name of Module	Test Case Description	Pre-conditions	Test Steps	Test Data	Expected Results	Actual Results	Test Result
REP_01	Reportin g	Verify that the system can successfully generate a prescription report of patient	Prescriptions that have been currently generated in the system.	1)Navigate to the home page. 2)View and Generate medication report.	Valid date range or no date range.	The system generates a report detailing the prescription history.	The system generates a report detailing the prescription history within the specified date range or for all prescriptions..	Pass
REP_02	Reportin g	Verify that the system allows users to export generated reports to a CSV file.	Medication reports and prescription reports present in the system.	1)Generate a report 2)Look for an option to export the report to CSV. 3)Click on the export option.	N/A	The system exports the report to a CSV file, and the file is downloaded to the user's device.	The system exports the report to a CSV file, and the file is downloaded to the user's device successfully.	Pass
REP_03	Reportin g	Verify that the system handles the scenario where no data is available for the selected report.	No appointments or prescriptions	1)Navigate to the home page. 2)View medication report.	N/A	The system displays a message indicating that there is no data available for the selected report type.	The system displays a message indicating no data accordingly.	Pass

INTEGRATION TESTING

1)Database Connection:
Test: Verify that the hospital management system connects successfully to the MySQL database.
Expected Result: Successful connection without errors.

2)User Authentication Integration:
Test: Check if the login page allows authentication for Patients, Doctors, and Pharmacists.
Expected Result: Users can log in with valid credentials; unauthorized access is restricted.

3)Appointment Workflow:
Test: Integrate the appointment process from Patients booking to Doctors' acceptance or rejection.
Expected Result: Appointments are successfully booked, and Doctors can approve or reject them.

4)Prescription and Review Display:
Test: Ensure that Patients can view prescriptions and reviews from their previous visits.
Expected Result: Patient access to accurate and updated prescription and review information.

5)Pharmacist Integration:
Test: Confirm that Pharmacists can view medications and mark them as provided.
Expected Result: Pharmacists can access medication details and update the status as "Provided."

6)End-to-End Workflow:
Test: Simulate a complete workflow from Patient booking to Pharmacist marking prescriptions as provided.
Expected Result: Seamless end-to-end functionality with data integrity maintained.

7)User Roles and Permissions:
Test: Validate that Patients, Doctors, and Pharmacists have the correct permissions and restrictions.
Expected Result: Each user role can only access and modify data within their specified permissions.

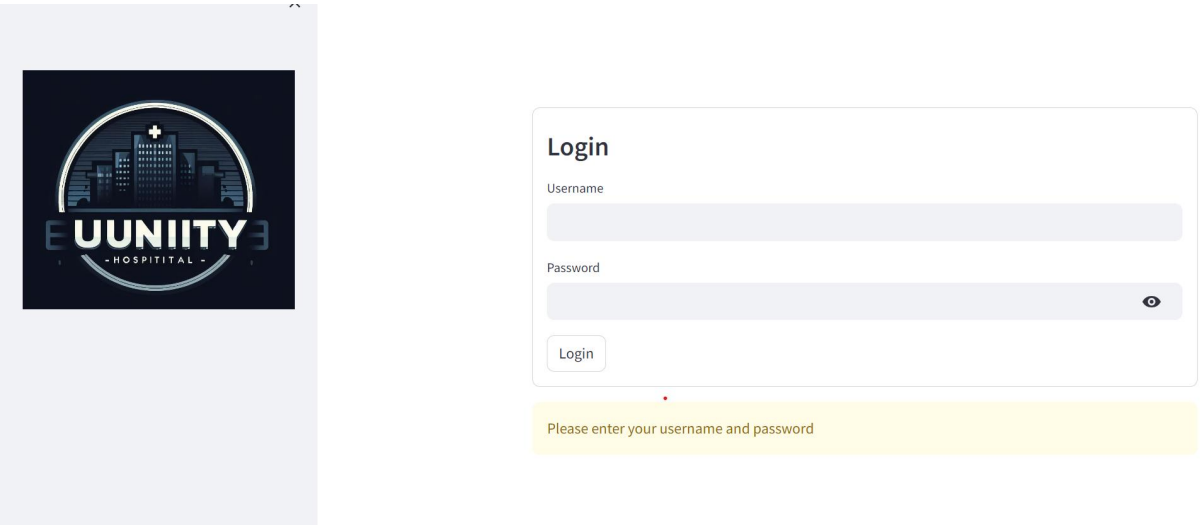
8)Error Handling:
Test: Integrate error scenarios, such as incorrect login credentials or invalid data submissions.
Expected Result: System gracefully handles errors, providing appropriate feedback to users.

9)Data Consistency:
Test: Verify that data consistency is maintained across the system, especially between appointments, medications, and reviews.
Expected Result: Data is consistent and accurately reflects the interactions within the system.

10)Responsive UI:
Test: Ensure that the Streamlit-based UI is responsive and functions well across different devices.
Expected Result: The UI is user-friendly and adapts to various screen sizes without loss of functionality.

FRONT END DEVELOPMENT:

LOGIN PAGE:



LOGINING AS A PATIENT:

Welcome *Jane*

Your Medical History:

	Date-Time	Doctor Name	Diagnose	Review	Prescription
0	2023-11-22 14:30:00	Dr. Johnson	Headache	Stable condition	Ibuprofen

APPOINTMENTS SCHEDULED FOR PATIENT(JANE IN OUR EXAMPLE):

Your Appointment:

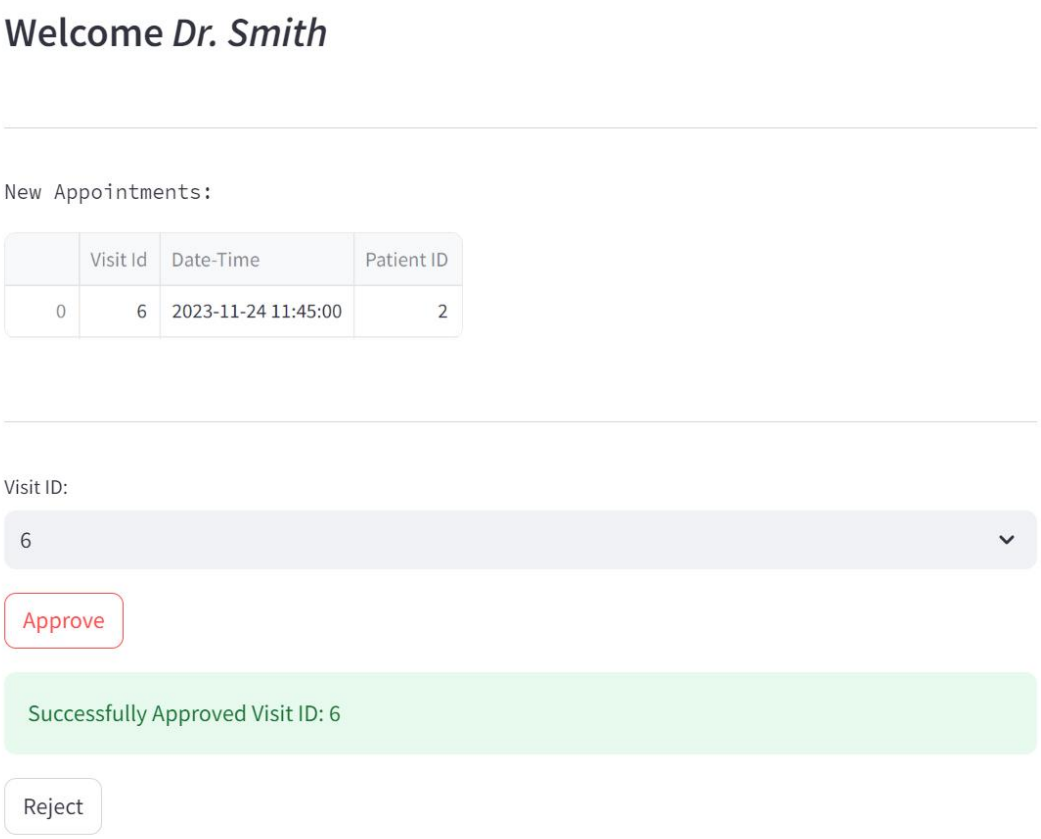
Date: 2023-11-24

Time: 11:45:00

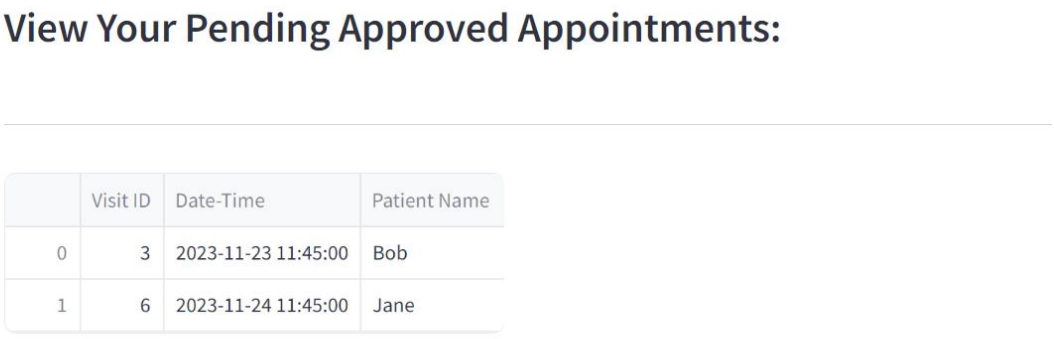
Your appointment is not yet approved by any Doctor

LOGING AS A DOCTOR:

New appointment request received by doctor:



VIEWING PENDING APPROVED APPOINTEMNET



OPD:

OPD

	Visit ID	Date-Time	Patient Name
0	3	2023-11-23 11:45:00	Bob
1	6	2023-11-24 11:45:00	Jane

Select Visit ID:

3

Patient ID: 3

Patient Name: Bob

Patient Age: 40

No Patient History Available

VIEWING FROM PHARMACIST:

Welcome *PharmaCare*

Provide Medications:

	Visit ID	Patient	Prescription
0	4	Alice	Cough syrup

Visit ID:

4

Provide Medicine to Visit ID: 4

JIRA CERTIFICATION:
1.KRUTHI

Credential Earned

Jira Work Management Fundamentals Badge

Completed by GOLLAPUTI SAI KRUTHI 2022 Batch PES University EC on November 21, 2023

Score:88 Completion ID: 286176447

GOWRI SRIYA:

Credential Earned

Jira Work Management Fundamentals Badge

Completed by GOWRI SRIYA R V S 2022 Batch PES University EC on November 21, 2023

Score:84 Completion ID: 286177096

Credential Earned

Jira Work Management Fundamentals

Badge

Completed by Darsh Patel on November 21, 2023

Score: 88 Completion ID: 286176872

Credential Earned

Jira Work Management Fundamentals Badge

Completed by DEVANJAN BANERJEE 2022 Batch PES University EC on November 21, 2023

Score: 96 Completion ID: 286177517