EYECARE
Eye Hospital Management System

Mini Project Report

Submitted by

Liya K Joseph

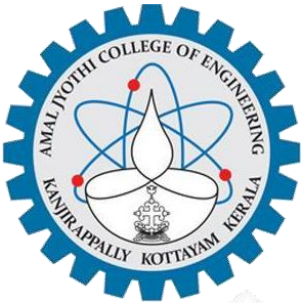
Reg. No.: AJC19MCA-I035

In Partial fulfillment for the Award of the Degree of

INTEGRATED MASTER OF COMPUTER APPLICATIONS

(INMCA)

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY



AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2023-2024

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, "**EYECARE"** is the bona fide work of **LIYA K JOSEPH (Regno: AJC19MCA-I035)** in partial fulfillment of the requirements for the award of the Degree of Integrated Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2023-24.

Miss Sona Maria Sebastian

Internal Guide

Miss Meera Rose Mathew

Coordinator

Rev. Fr. Dr. Rubin Thottupurathu Jose

Head of the Department

DECLARATION

I hereby declare that the project report **"EYECARE"** is a bona fide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Integrated Master of Computer Applications (INTMCA) from APJ Abdul Kalam Technological University, during the academic year 2023-2024.

Date: 31-10-2023 **LIYA K JOSEPH**

KANJIRAPPALLY **Reg: AJC19MCA-I035**

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our Manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev. Fr. Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Miss Meera Rose Mathew** for her valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also express sincere gratitude to my guide **Miss Sona Maria Sebastain** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

LIYA K JOSEPH

ABSTRACT

The Eye Hospital is a specialized healthcare facility that is committed to providing a wide range of eye care services for patients of all age groups. This modern hospital is primarily focused on diagnosing, treating, and managing various eye conditions and visual impairments. The project encompasses features for patient registration, appointment scheduling, medical record management, prescription and surgery reminders, billing and financial management, and medical supply control.

Patient registration involves the process of collecting personal information from patients. This includes details about any previous medical records from other hospitals, the reason for their visit (such as injury, redness, routine eye examination, specific examinations, driving license certification, or vision checks for prescription glasses), allergies to medications, and current medication usage. After completing registration, patients are required to make online payments for their appointments. Registered users can also schedule additional appointments using unique codes provided by the hospital.

For doctors, the system allows them to view their scheduled patient appointments, manage patient medical histories, schedule appointments, request leave, view patient lists, and check available dates and time slots.

In the pharmacy department, responsibilities include managing patient payments, providing instructions on how to use prescribed medications, maintaining patient records, and managing the inventory of medicines by adding, removing, and updating stock items while also monitoring medication expiration dates.

Once patients complete their registration, the receptionist assigns them to different doctors based on their eye conditions. Patients are categorized using a color-coded system and are given a unique code during registration. The receptionist also serves as a point of contact for patients, answering any questions they may have and facilitating communication between different hospital departments.The administration module is crucial for managing the hospital effectively. It involves maintaining a complete list of doctors, nurses, and other staff members by adding, removing, updating, and scheduling their work hours. Additionally, it manages the financial details of staff and various departments, vendor and supplier information, generates reports, conducts audits of each department, and oversees treatment categories and doctor appointment schedules. The administrator also facilitates communication with hospital staff, doctors, patients, and other stakeholders, ensuring that important updates are conveyed through notifications and alerts.

CONTENT

List of Abbreviation

HTML	**Hyper Text Markup Language**
CSS	**Cascading Style Sheet**
MYSQL	**My Structured Query Language**
UML	**Unified Modelling Language**
IDE	**Integrated Development Environment**
AJAX	**Asynchronous JavaScript and XML**
JS	**JavaScript**
SQLITE	**Structured Query Language Lite**
RDBMS	**Relational Database Management System**
IP	**Internet Protocol**
URL	**Uniform Resource Locator**
QMG	**Question Administration Gather**
UAT	**User Acceptance Test**
ML	**Machine Learning**
BDD	**Behavioural Driven Development**

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

To overcome the issues that existed with the traditional manual approach, the "EYECARE" hospital management system was created. The difficulties our current system has are supported by this program in an effort to eliminate and, in some circumstances, decrease them. Additionally, this system is created to meet the specific requirements of the business to conduct operations efficiently and effectively.

The program is kept as simple as possible to reduce data entry errors. Additionally, it displays an error notice when you enter invalid data. The user doesn't require any formal training to use this system. This alone demonstrates how user-friendly it is. As previously said, the Eye Care Appointment System can result in an error-free, secure, dependable, and quick management system. It might help the user focus on their other tasks rather than keeping records. Consequently, it will aid organizations in making better use of their resources.

1.2 PROJECT SPECIFICATION

The system's goal is to automate the current manual process with the aid of computerized tools and comprehensive computer software, meeting their needs in the process. This will allow for the storage of their important data for a longer period of time with simple access to and manipulation of the data. It is simple to find and use the necessary software and hardware.

The goal is to automate the company's current manual method with the use of computerized tools and comprehensive computer software, meeting their needs, in order to retain their important data and information for a longer period of time with simple access and manipulation.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

To overcome the issues that existed with the traditional manual approach, the "EYECARE" hospital management system was created. The difficulties our current system has are supported by this program in an effort to eliminate and, in some circumstances, decrease them. Additionally, this system is created to meet the specific requirements of the business to conduct operations efficiently and effectively. The system's goal is to automate the current manual system with the aid of computerized hardware and comprehensive computer software, meeting their criteria, in order to preserve their important data and information for a longer period of time with simple access and manipulation. The necessary gear and software are readily available and simple to use.

2.2 EXISTING SYSTEM

The traditional method of scheduling appointments at an eye clinic involves making phone calls, which is a time waster. The system that has been developed allows for online scheduling and works to reduce the need for paper records and patient lines.

2.2.1 NATURAL SYSTEM STUDIED

The eye hospital management system aims to provide comprehensive care for patients with various eye conditions and visual impairments. At the core is an integrated system that handles patient registration, appointment scheduling, medical records, billing, inventory management, and other functions crucial for smooth hospital operations. Patients can book appointments online, provide medical history, and access records through a secure portal. Doctors manage patient appointments, apply leave, and update medical records in the system. The pharmacy handles billing, medication dispatch, and inventory management. The receptionist allocates patients to doctors based on symptoms, maintains communication across departments, and provides support. The administrator oversees staff management, financials, reporting, and hospital analytics. Overall, the system utilizes technology to optimize workflows and enhance coordination across departments to deliver quality eye care services to patients. The various modules and stakeholders work synergistically as an integrated natural system focused on patient-centric care.

2.2.2 DESIGNED SYSTEM STUDIED

To optimize the operations and management of the eye hospital, an integrated digital system has been designed. The system encompasses modules for patient registration, appointment

booking, medical records, prescription management, billing, inventory control, and analytics. Patients can self-register on the portal by providing demographics, medical history, insurance details etc. The system allows booking, modifying or canceling appointments online. Doctors can view scheduled appointments, update medical records with diagnosis details, and e-prescribe medications. The billing module generates invoices, processes payments, and tracks revenue cycle. Pharmacy staff manage medication inventories, dispatches, and refill reminders leveraging the system. Receptionists use the system to allocate patients to doctors based on symptoms. The administrator handles staff management, reporting, system configuration via centralized dashboards. The modular design with role-based access allows specific functionality per user roles. Data is integrated across modules for seamless information flow. The system utilizes latest technologies like ML, cloud computing to enable efficient operations. Together, the designed system aims to provide data-driven, networked and comprehensive eye care services to patients.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Time consuming manual processes for registration, scheduling, billing etc. This can lead to long patient wait times.
- Difficult to retrieve and track paper medical records. Records can be misplaced or lost.
- Prescription errors are more likely with handwritten prescriptions.
- Inventory management of medicines and supplies is difficult without digital tracking. Expiry and stock-outs are common.
- Financial and billing errors are common without automated functionalities.
- No centralized database making it difficult to get holistic view of hospital operations.
- Lack of reminders and notifications for appointments, prescription refills etc. leading to missed/delayed care.

2.4 PROPOSED SYSTEM

- Patient portal for online registration and appointment booking to reduce wait times.
- Electronic medical records readily accessible to authorized staff for better clinical decision making.
- Customized dashboards for doctors to manage patient appointments and e-prescribe
- medications.

- Automated billing system with payment gateways to reduce billing errors and streamline payments.
- Pharmacy module to digitally manage medication inventories, track expiry, automate reordering.
- Role-based access control to ensure privacy and system security based on user privileges.

2.5 ADVANTAGES OF PROPOSED SYSTEM

- Reduced billing errors and delays through automated billing and payment processing.
- Improved care coordination across departments due to seamless information flow.
- Higher patient engagement with features like appointment reminders and prescription alerts.
- Lower operational costs by reducing paper usage and manual processes.
- Scalable system that can easily be upgraded to keep pace with latest technologies.
- 24/7 system access improving flexibility for hospital staff and patients.
- Better financial planning and forecasting using digitized data.
- Easier regulatory compliance with digital records and audit trails.

CHAPTER 3
REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

This analysis is a critical stage in establishing if a project will satisfy the association's goals in proportion to the resources, effort, and time put in it. It assists the designer in establishing the project's prospective emphasis points and long-term consequences. To determine whether a given framework is feasible and advantageous for further research, all options must be considered, including the impact of the proposed system on the association, assessments of resource efficiency, client satisfaction, and capabilities to meet client requests. As a result, an achievability study is carried out on a regular basis. Recently, permission was granted for the development of a modern application. The scope's particular, monetary, and operational rationale is as it is.

3.1.1 Economical Feasibility

The economic feasibility analysis is a critical step in establishing the value of a new project in terms of cost and time investment. It entails a detailed examination of all elements that may impact the initiative's success. Economic analysis is the most commonly utilized approach for assessing the efficacy of a proposed system. Thus, the suggested system, Eyecare, has undergone cost-benefit analysis and is proven to be both practical and cost-effective given the project's presumptive cost. To determine the system development cost, many cost categories were analysed, including labour expenses, computer costs, supplies and equipment costs, charges for implementing new software and computer equipment, system analysis, website coding costs, and database design costs. These are frequently one-time expenses that will be avoided after the project is completed. These cost categories may be extensively investigated to ensure that the system's development is economically viable and profitable. In case of profit, decision is taken to design and implement the system.

The proposed system is financially feasible because of the following reason:

- The proposed system is developed as a part of project so there is no manual cost to spend for proposed system.
- There is no much hardware and software cost because all the resources are already available.
- The proposed system is economic, as it will reduce the time investment in running the daily transaction

3.1.2 Technical Feasibility

Technical feasibility is the process of determining if it is possible to manufacture and distribute an item or service using the technology and resources that are already available. The tools, materials, labour, logistics, and technology of the proposed plan are assessed as part of the technical feasibility analysis to determine how successful it will be. Before commencing the assignment, it is critical to identify and address any potential project issues. Making a flowchart of the product or service's development might assist in visualizing the system's process.

Eyecare is simple to use and doesn't need much instruction because it is self-explanatory. Even for first-time users, the application is simple to use. Based on the technical feasibility assessment, it is evident that the proposed Eye Hospital Management System aligns well with the hospital's existing technical infrastructure and meets the performance requirements. The system is deemed technically feasible, and its implementation is recommended with due attention to integrating with the legacy pharmacy management system

Once the system has been designed, there are several ways to run it.

• Is the project feasible within the limits of current technology.

➢ Yes

Technical issues raised during the investigation are:

➢ Nothing

• Can the technology be easily applied to current problems?

➢ Yes

• Does the technology have the capacity to handle the solution?

➢ Yes

3.1.3 Behavioral Feasibility

The examination of whether a proposed project or system corresponds with the organization's culture, current procedures, and the readiness of users to embrace and adapt to the changes brought about by the project is referred to as behavioural feasibility. It assesses the human and behavioural elements of project implementation and resolves possible difficulties linked to stakeholder and end-user acceptability, resistance, and support.

Two critical considerations have been taken to ensure the system's success:

(1) if users will receive adequate support, and

(2) whether the system will be damaging.

These challenges were thoroughly investigated to ensure that the system will be usable after deployment. Furthermore, all behavioural components were evaluated during the feasibility evaluation to ensure that the project is behaviourally feasible.

The behavioural feasibility assessment reveals strong acceptance of the proposed Eye Hospital Management System among hospital staff and patients. With comprehensive training, effective change management, and a focus on patient-centric features, the system is deemed behaviourally feasible and holds the potential to enhance patient care and operational efficiency.

3.1.4 Feasibility Study Questionnaire

1. **How are patient eye records currently managed? Are they stored in physical files or documents? How are they organized?**

   Patient eye records are currently managed using physical files stored in cabinets, organized based on patient names and medical record numbers.

2. **What are the key administrative tasks related to eye care currently performed manually, such as patient registration, appointment scheduling, billing, inventory management for eye medications, etc.?**

   Key administrative tasks include patient registration, appointment scheduling, billing and invoicing, prescription management, and basic inventory management for eye medications and supplies.

3. **What are the main challenges or pain points experienced with the current manual system in eye care management?**

   The main challenges include time-consuming tasks, difficulty in quickly retrieving patient eye records, potential data loss or misplacement, and limited access to eye care information outside the hospital premises.

4. **Are there any regulatory or compliance requirements that need to be considered for the new eye hospital management system?**

   Yes, the new system needs to comply with relevant medical regulations and ensure the security and privacy of patient eye care data.

5. **Do you need integration with existing systems or third-party services (e.g., diagnostic imaging devices, pharmacy for medication orders)?**

   Yes, we would like integration with diagnostic imaging devices for viewing test results, and with the pharmacy for medication orders and inventory management.

6. **How should the user interface and navigation be designed to ensure ease of use and**

efficiency in eye care management?

The user interface should be intuitive, with easy navigation and clear categorization of functionalities. It should be designed to minimize clicks and streamline tasks specific to eye care management.

7. **How would you like the eye hospital management website to handle patient communication, appointment reminders, and follow-ups related to eye care?**

We would like the website to support automated patient communication, appointment reminders, and follow-up notifications related to eye care through email or SMS.

8. **What is the expected timeline for implementing the eye hospital management website?**

We aim to have the eye hospital management website fully operational within nine months from the start of the development process.

9. **How would you like the eye hospital management website to facilitate communication and collaboration among different eye care specialists and staff?**

We want the website to include a secure messaging system to enable efficient communication and collaboration among eye care specialists and staff.

10. **How do you currently manage appointments and patient scheduling? Is there a system in place for reminders and rescheduling?**

Appointments and patient scheduling are managed manually through appointment books. There is no system for automated reminders or rescheduling.

3.1 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor - 11th Gen Intel(R) Core(TM) i5

RAM - 8.00GB

Hard disk - 1TB

3.2.2 Software Specification

Front End - HTML, CSS, JAVASCRIPT,

Back End - SQLite3

Database - Python, Django

Client on PC - Windows 7 and above.

Technologies used - JS, HTML5, AJAX, J Query, PHP, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 PYTHON

Python is a high-level programming language celebrated for its simplicity, readability, and versatility. Created by Guido van Rossum and first introduced in 1991, Python has garnered widespread popularity. One of its standout features is its readability, which is achieved through a straightforward and elegant syntax that relies on indentation for defining code blocks. Python is an interpreted language, allowing developers to write and test code without the need for compilation. Its dynamic typing means that variables do not require explicit data type declarations, enhancing flexibility. The language boasts a rich standard library that covers a diverse range of functionalities, simplifying tasks from file I/O to web development. Python is known for its cross-platform compatibility, supporting various operating systems. With its object-oriented capabilities and strong community support, Python has become a go-to language for various applications, including web development, data analysis, machine learning, and scientific computing. Its open-source nature makes it freely available for all, contributing to its widespread adoption and continued growth. Python's easy learning curve and active community make it a popular choice, especially for beginners in programming, making it a language that's both accessible and powerful.

3.3.2 DJANGO

Django is a high-level, open-source web framework for building robust and scalable web applications. It follows the Model-View-Controller (MVC) architectural pattern, but in Django, it's referred to as Model-View-Template (MVT). This framework simplifies web development by providing a ready-made structure for handling common web application tasks, allowing developers to focus on building features rather than dealing with repetitive and complex code. Django offers a powerful Object-Relational Mapping (ORM) system that facilitates database interactions and a built-in admin interface for managing application data. It also includes a secure authentication system and supports user management out of the box. Django encourages the use of reusable components through its "app" system, enabling developers to create and share functionality, enhancing productivity. It emphasizes the "Don't Repeat Yourself" (DRY) principle and follows best practices for security, making it a

dependable choice for building web applications, whether they are small-scale projects or large-scale, enterprise-level systems. With a thriving community, extensive documentation, and a wealth of third-party packages, Django is a popular choice for web development, providing developers with the tools they need to create feature-rich and secure web applications efficiently.

3.3.3 SQLite

SQLite is a lightweight, serverless, and self-contained relational database management system (RDBMS) that is often embedded directly into applications. It's a popular choice for mobile and desktop applications due to its simplicity, low overhead, and ease of use. SQLite is self-contained in the sense that it operates as a single, stand-alone file, making it incredibly portable and simple to set up. It uses a subset of the SQL language to manage data, allowing developers to create, read, update, and delete records with ease. Despite its simplicity, SQLite is capable of handling a wide range of data types, indexes, and transactions. It's often used for small to medium-sized databases or situations where a full-scale RDBMS might be overkill. Being serverless, SQLite doesn't require a separate server process, which can make it faster and more resource-efficient for certain use cases. Many programming languages, including Python, have built-in support for SQLite, making it a popular choice for developers who need a local database for their applications.

CHAPTER 4
SYSTEM DESIGN

4.1 INTRODUCTION

Design is the inception of any engineered system or product. It's a creative process, where diverse ideas and methodologies converge to meticulously specify a system, device, or process. The quality of a system largely hinges on its design, and it's the cornerstone of efficient systems. In this report, we'll explore the art and science of system design, delving into architectural intricacies, data flows, modules, and technology choices that underpin our project. These design decisions are the result of careful consideration, balancing trade-offs and unique challenges. This report offers a comprehensive guide for stakeholders, ensuring a deep understanding of how our system aligns with project goals and functions. Join us in the exploration of design as we pave the way for our project's realization.

4.2 UML DIAGRAM

The Unified Modeling Language (UML) stands as a cornerstone in the realm of software engineering and systems design. This standardized, visual modeling language offers a comprehensive set of notations and diagrams, which together serve as a lingua franca for software architects and developers. UML's primary function is to provide a visual, standardized way of capturing, documenting, and communicating myriad facets of a system. These facets encompass not only the system's static structure but also its dynamic behavior and the intricate web of interactions among its components. The UML diagram presented here is a tangible representation of a specific facet of our system, one meticulously chosen to encapsulate and convey a core aspect of its architecture or functionality. UML diagrams, in their elegant simplicity, have the power to distill complex systems into understandable visual representations, where each symbol and line carries meaning. As a result, these diagrams become invaluable tools for comprehending, discussing, and, crucially, building complex systems.

4.2.1 USE CASE DIAGRAM

Use case diagrams in software and systems design are graphical representations that visually illustrate how a system interacts with its users (actors) and external entities (e.g., other systems or devices). Actors, depicted as stick figures, initiate specific actions or use cases (ovals), which represent the system's functionalities. These diagrams not only clarify the system's functionality and scope but also serve as a roadmap for developers, ensuring that the project aligns with user requirements and external interactions. Use case diagrams

streamline communication, enhance development processes, and facilitate quality assurance by translating use cases into test scenarios, ultimately contributing to effective and efficient software and system development.

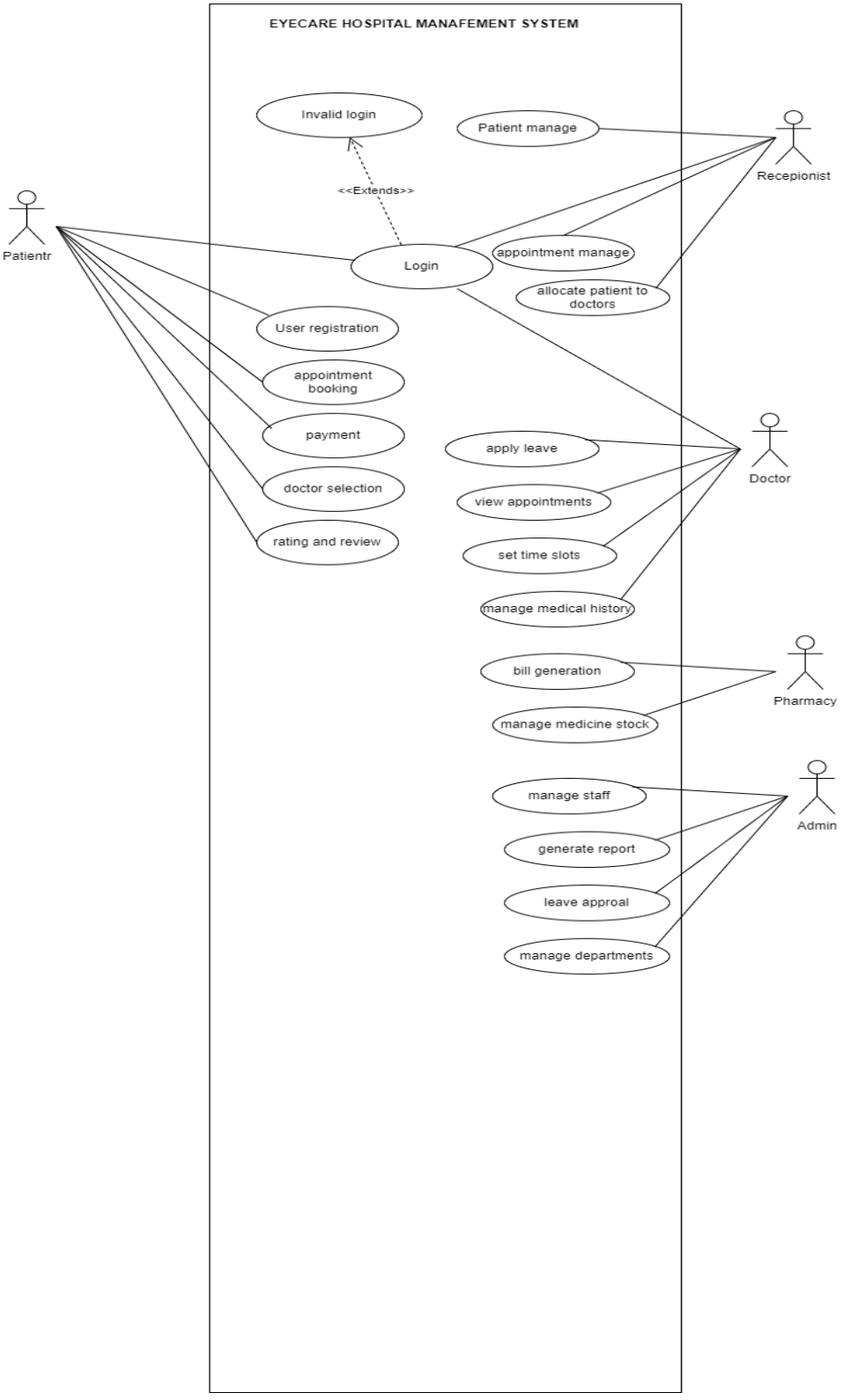


Fig 1: Use Case Diagram

4.2.2 SEQUENCE DIAGRAM

A sequence diagram, a key tool in Unified Modeling Language (UML), offers a visual representation of how objects and components interact within a system over time. It showcases the precise order and flow of messages or actions between these elements, making it an invaluable aid for understanding the dynamic behavior of a system. Sequence diagrams serve to depict complex processes, aiding in system analysis, design, and communication among stakeholders, while also contributing to the development of efficient and error-free systems by highlighting the timing and dependencies of actions.

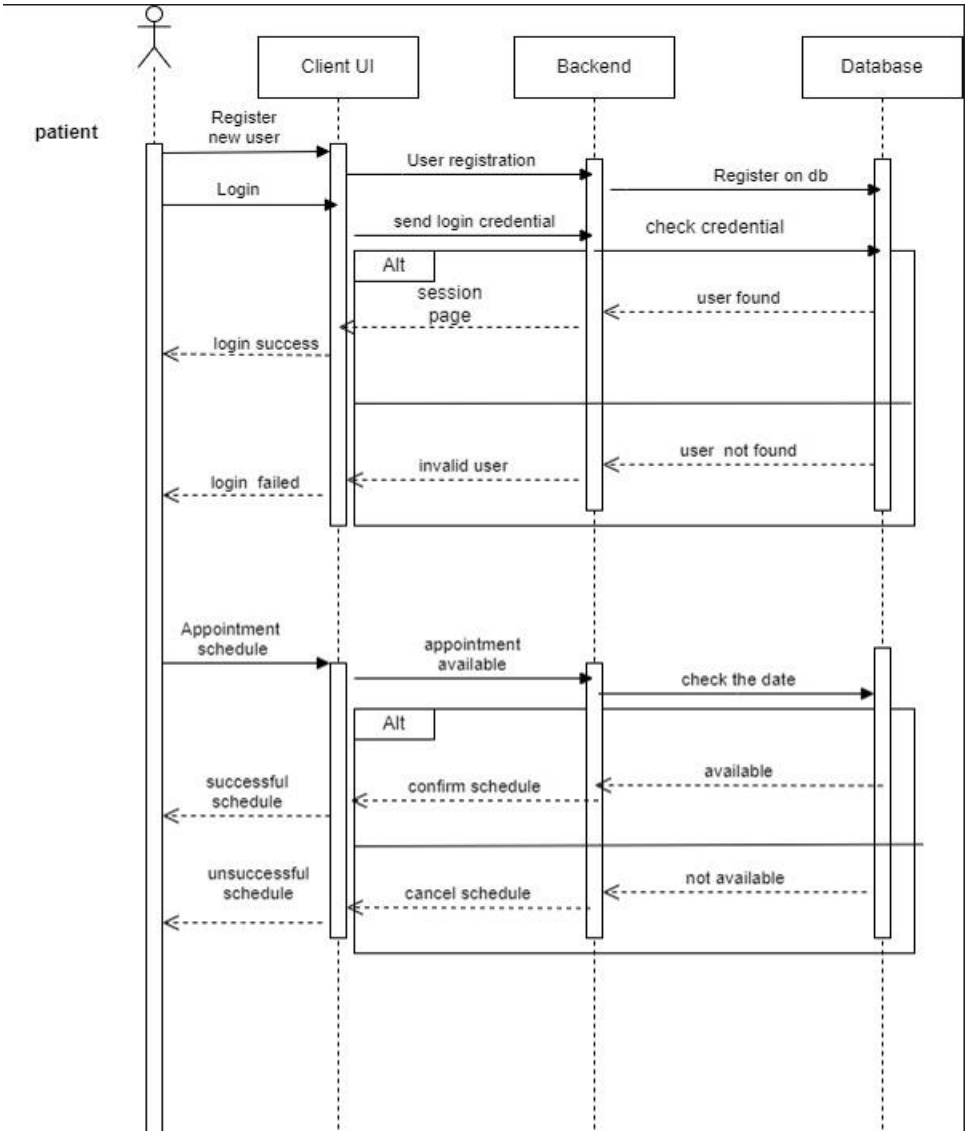


Fig 2: Sequence Diagram

4.2.3 STATE CHART DIAGRAM

A state chart diagram, a fundamental component of Unified Modeling Language (UML), provides a visual representation of the various states and transitions that an object or system can undergo in response to external events or conditions. These diagrams are essential for modeling the dynamic behavior of systems, emphasizing states, events, and transitions. State chart diagrams are particularly valuable for depicting the lifecycles of objects, processes, or entities, offering clarity on the sequential progression and conditions under which state changes occur. They are indispensable tools for system analysis, design, and communication, enabling stakeholders to grasp the intricate behavioral aspects of a system while facilitating the development of robust and responsive software or processes.

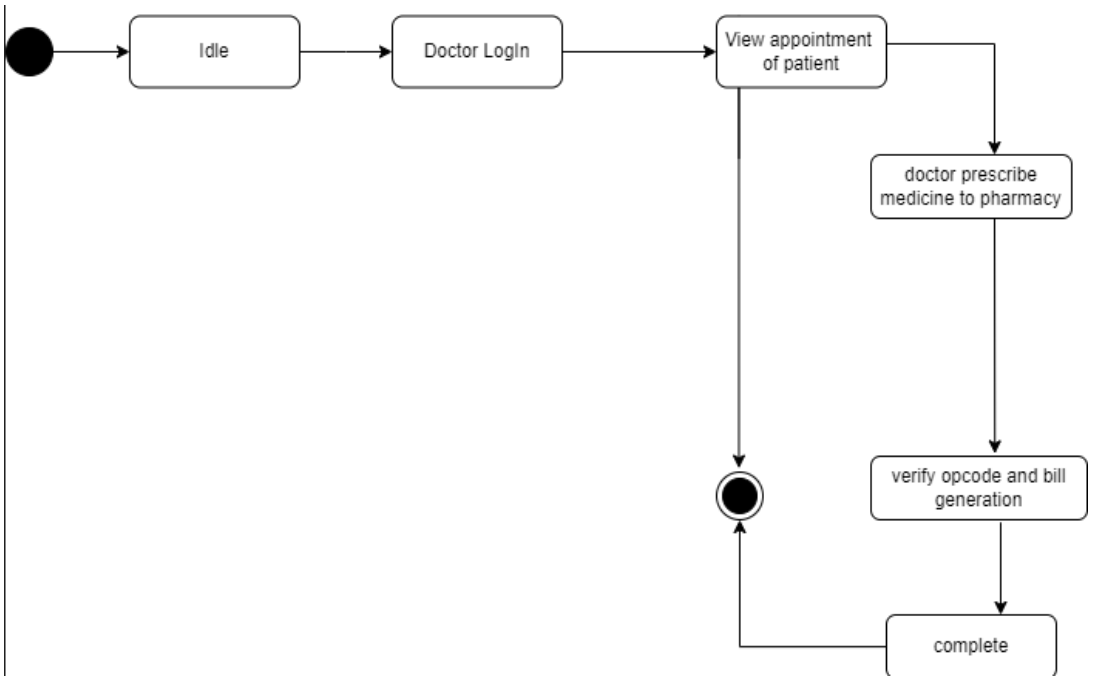


Fig 3: State Chart Diagram

4.2.4 ACTIVITY DIAGRAM

An activity diagram, a core element of Unified Modeling Language (UML), serves as a visual representation of the dynamic aspects of a system, emphasizing the flow of activities or actions within a process. These diagrams are instrumental for modeling business processes, workflows, and complex algorithms, focusing on actions, decisions, and their sequencing. Activity diagrams offer a clear and intuitive means of depicting the logic and control flow within a system or process, making them indispensable for system analysis, design, and communication among stakeholders. They aid in the development of efficient procedures, ensuring that complex operations are visually represented and easily comprehensible, thereby contributing to effective problem-solving and process optimization.

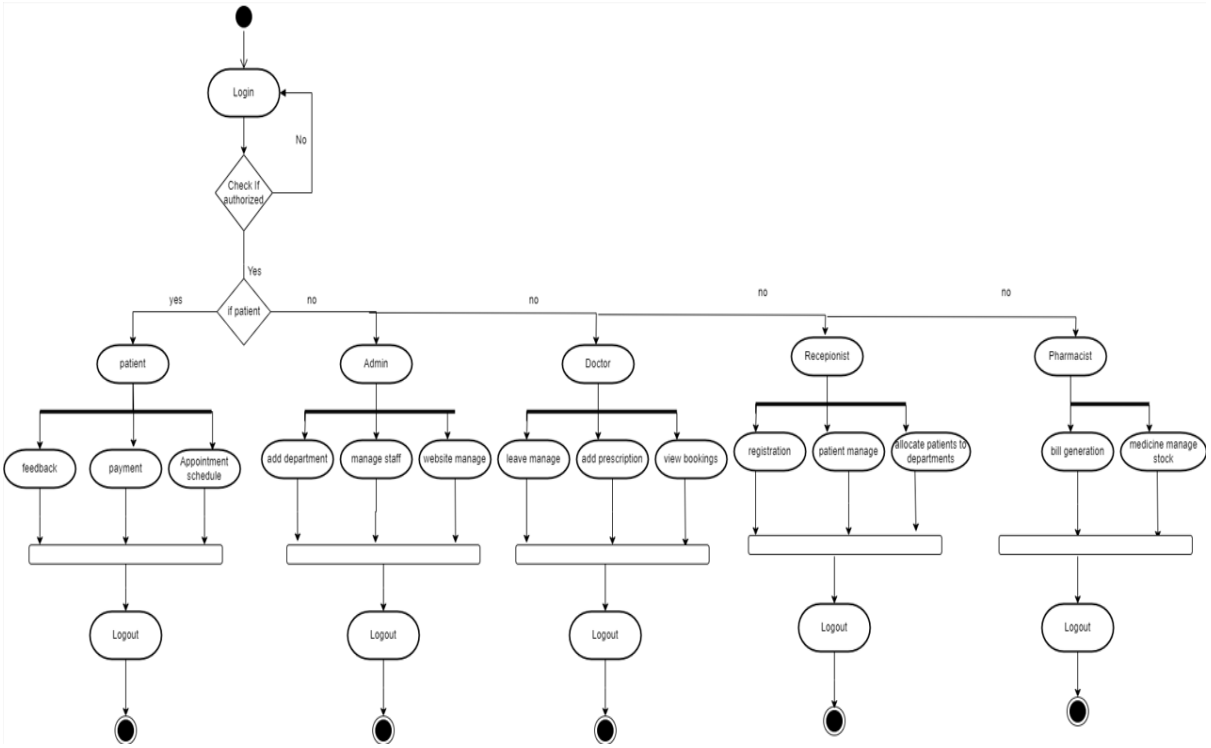


Fig 4: Activity Diagram

4.2.5 CLASS DIAGRAM

A class diagram, a fundamental component of Unified Modeling Language (UML), provides a visual representation of the structure of a system or application, focusing on the classes, their attributes, and the relationships between them. These diagrams are pivotal for modeling the static aspects of software systems, allowing for the clear definition of classes, their properties, methods, and associations. Class diagrams are essential for designing and communicating the structural blueprint of a system, aiding in the organization and planning of software development projects. They facilitate a deep understanding of the system's architecture, the relationships between classes, and the distribution of responsibilities, thereby contributing to efficient, maintainable, and well-structured software solutions.

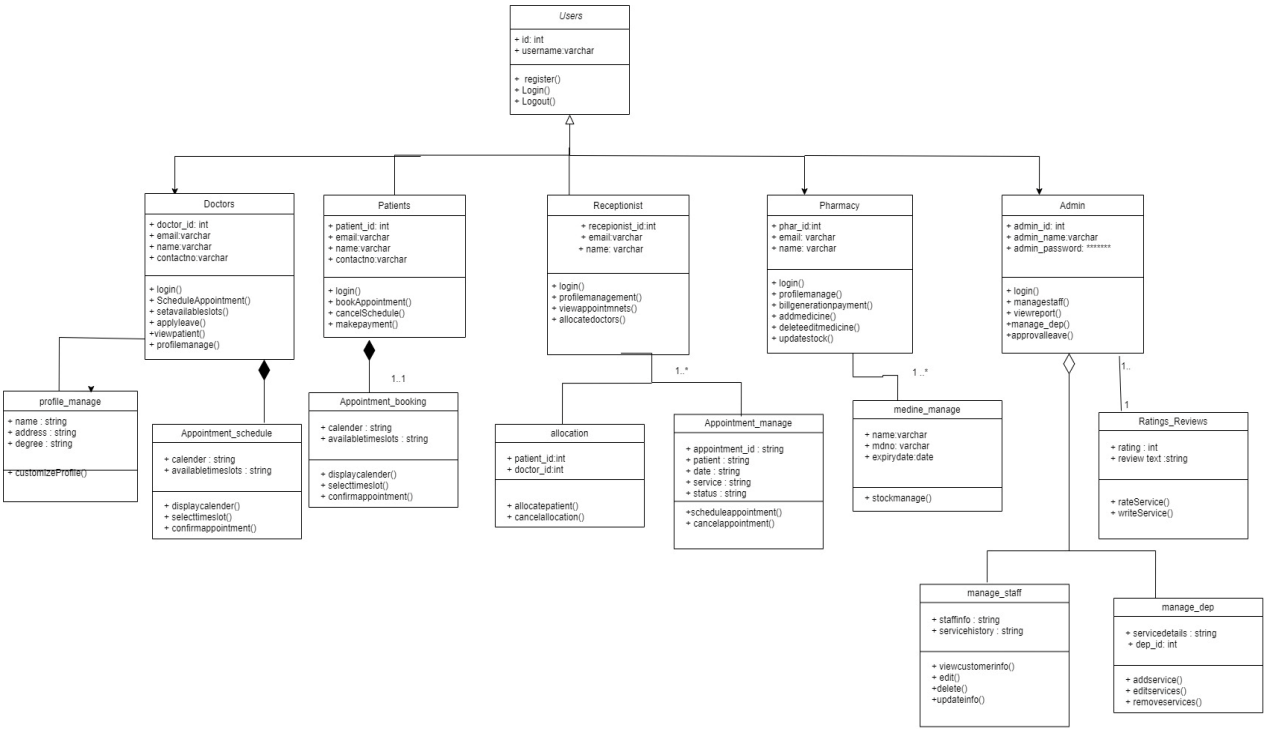


Fig 5 : Class Diagram

4.2.6 OBJECT DIAGRAM

An object diagram, a key element of Unified Modeling Language (UML), offers a specific and static snapshot of a system at a particular moment, focusing on individual objects and their relationships. These diagrams provide a detailed view of instances of classes, showcasing the actual data values and relationships between objects during runtime or at a specific point in a system's execution. Object diagrams are instrumental for verifying and understanding complex system configurations, aiding in system testing and debugging by visually representing the concrete instances and their interactions, ensuring that the system operates as intended. They serve as a powerful tool for fine-grained analysis and quality assurance, making it easier to pinpoint issues and validate that the system's runtime behavior aligns with the design.

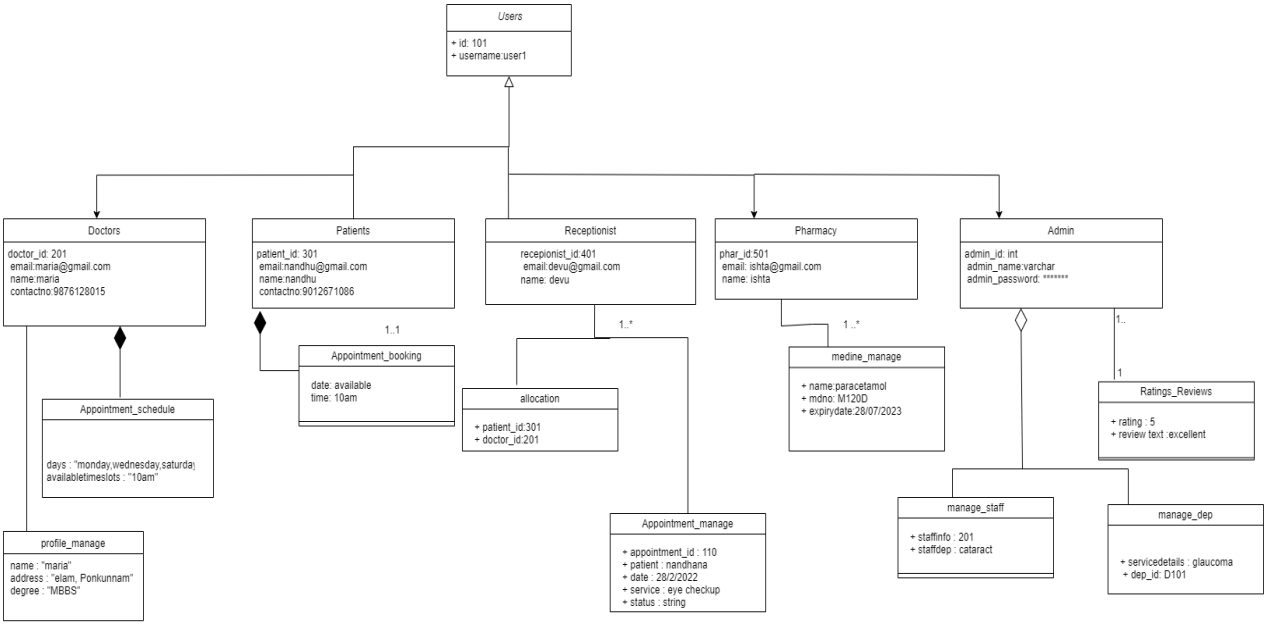


Fig 6: Object Diagram

4.2.7 COMPONENT DIAGRAM

A component diagram, a core element of Unified Modeling Language (UML), serves as a visual representation of a system's architecture, emphasizing the high-level components or modular building blocks and their interactions. These diagrams provide a detailed view of how various components, which can be classes, files, libraries, or even subsystems, are connected and work together within a system. Component diagrams are instrumental for understanding the system's overall structure, illustrating the relationships between components, and facilitating component-based software engineering and design. They offer insights into how the system's functionalities are distributed and organized, making them a crucial tool for system architects and developers. Component diagrams assist in ensuring that the system is modular, maintainable, and scalable, and that it adheres to sound architectural principles.

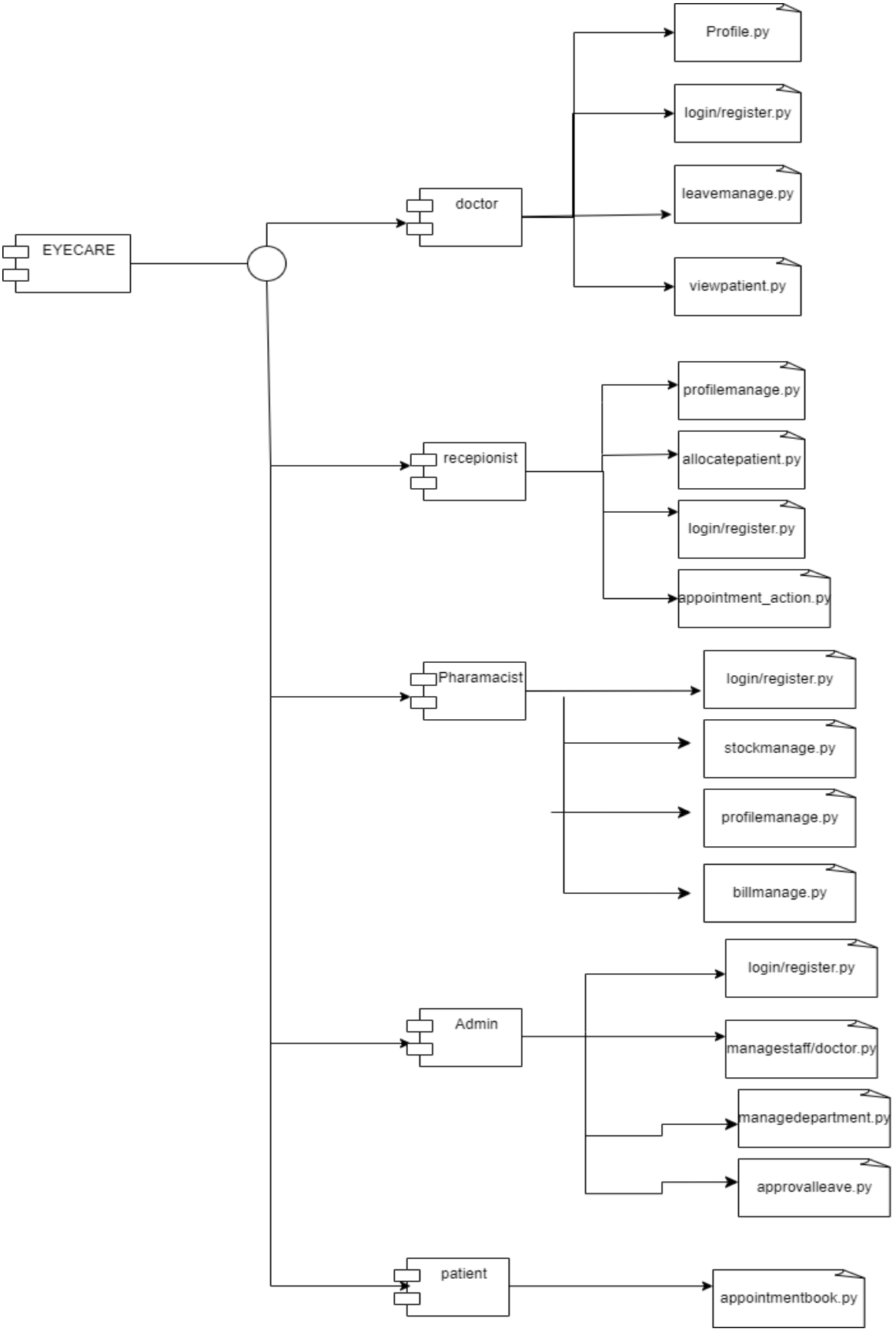


Fig 7: Component Diagram

4.2.8 DEPLOYMENT DIAGRAM

A deployment diagram, an integral aspect of Unified Modeling Language (UML), provides a precise visual representation of how a software system is physically deployed or distributed across hardware components and nodes, focusing on the actual deployment architecture. These diagrams illustrate the arrangement of nodes, which can represent hardware devices or servers, and the components or software artifacts that run on these nodes. Deployment diagrams are crucial for system architects and administrators, allowing them to design, document, and understand the physical configuration of a system, encompassing servers, databases, and other infrastructure elements. They serve as a vital tool for ensuring that a system is deployed efficiently, offering insights into scalability, redundancy, and fault tolerance, thereby contributing to the effective deployment and management of complex software systems.

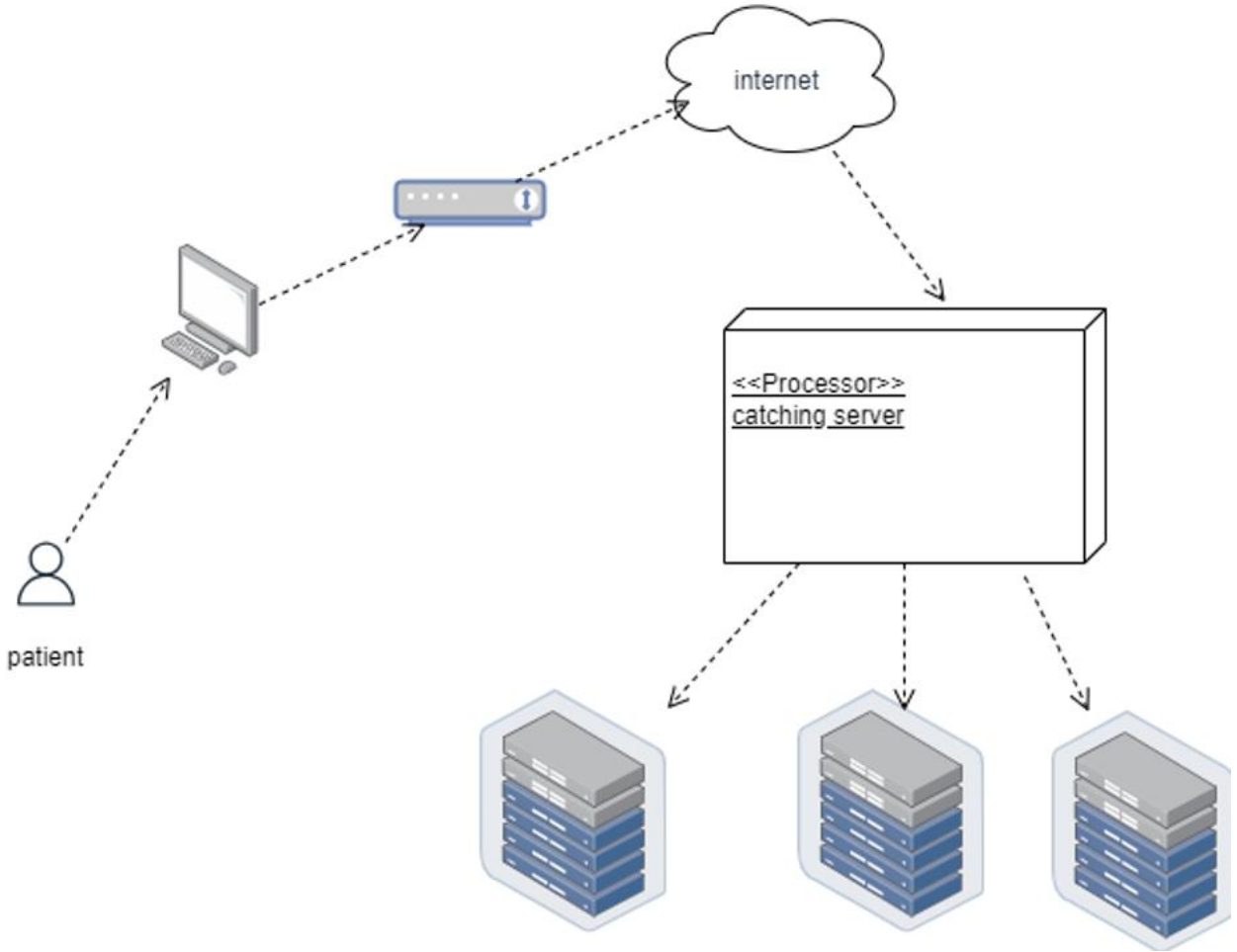
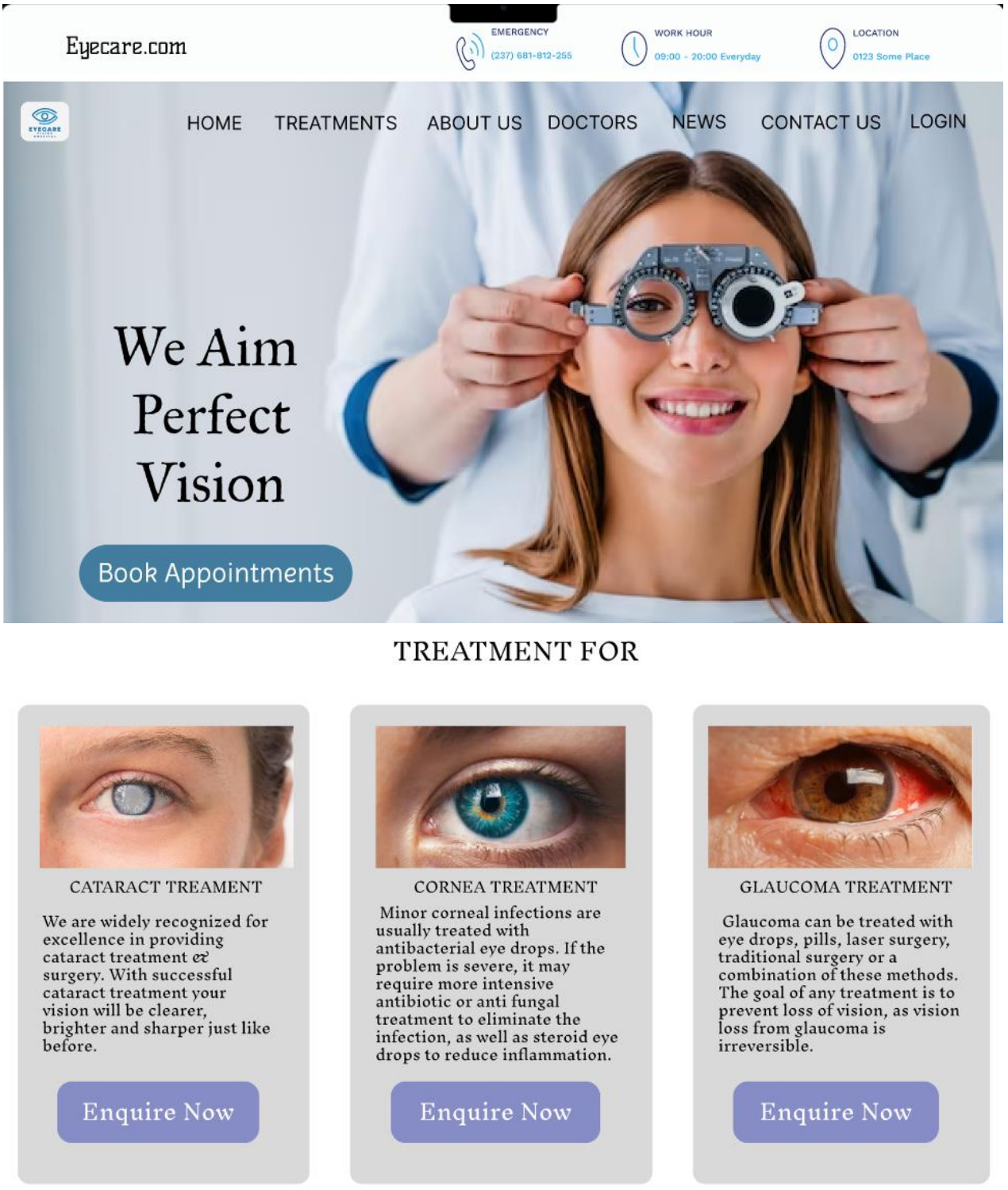


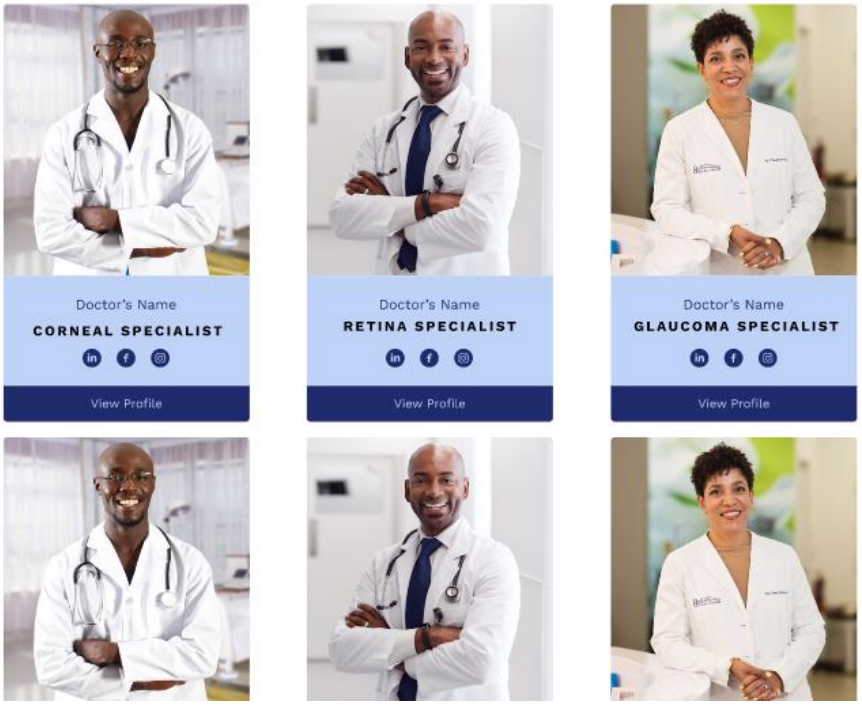
Fig 8: Deployment Diagram

4.3 USER INTERFACE DESIGN USING FIGMA

Form Name: web page



OUR DOCTORS



ABOUT US

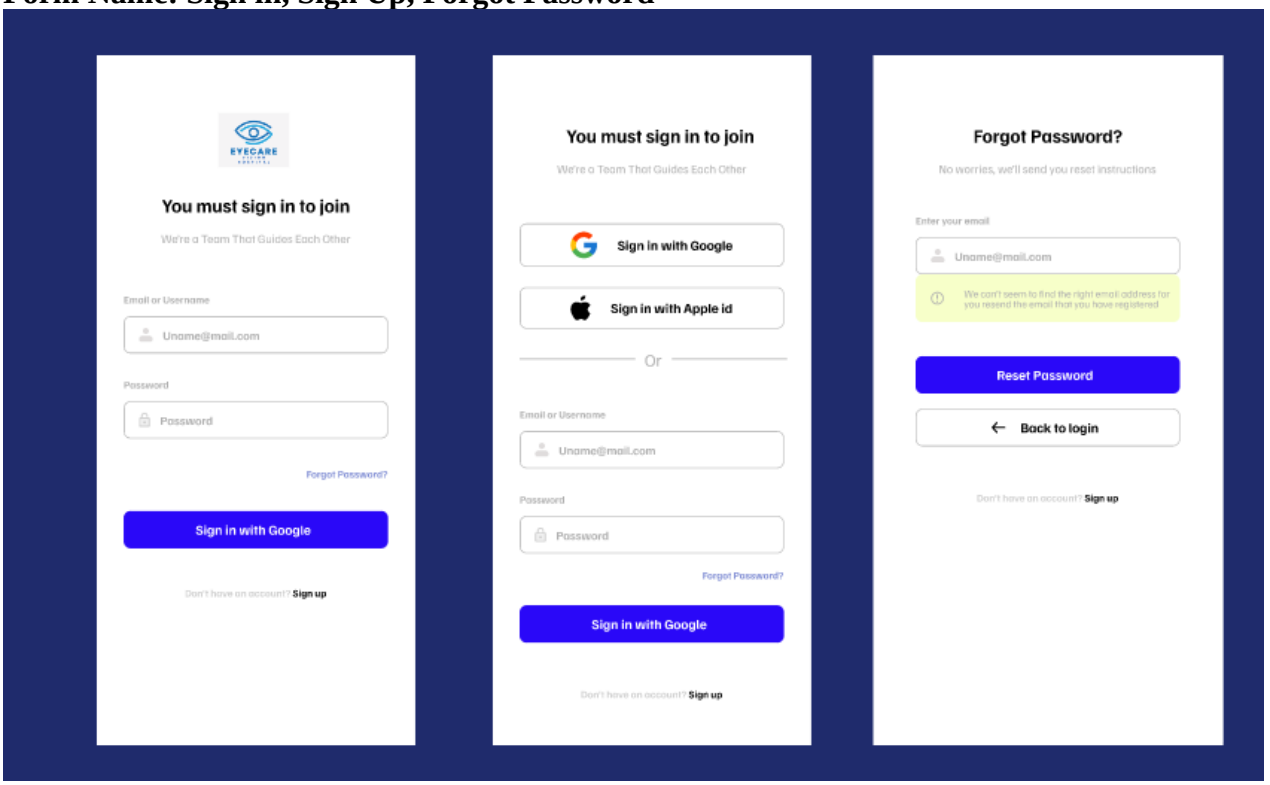
WELCOME TO HOSPITAL NAME

Best Care for Your Good Health

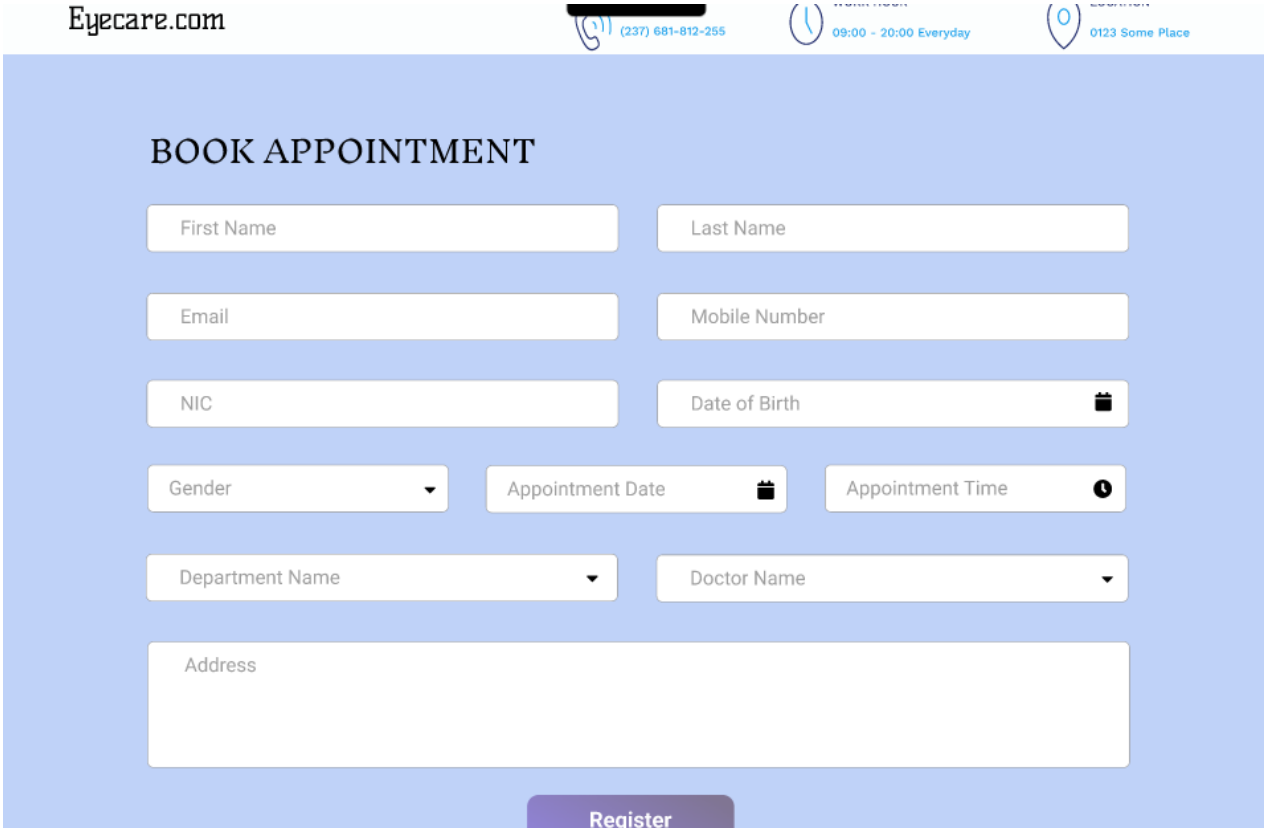
- A Passion for Healing
- 5-Star Care
- All our best
- Believe in Us
- Always Caring
- A Legacy of Excellence

We, EyeCare.com, situated at Anikad, Kottayam, Kerala are a super speciality eye hospital providing a wide range of complete eye treatments under one roof. Our motto is to provide best, comprehensive and affordable eye care with the latest comprehensive technologies. We are committed to achieve and maintain excellence in eye care. Our dedicated team of specialist doctors cater to all eye diseases from corneal pole to retinal pole of the eyeball and take care of paediatric to geriatric eye services. A full array of state of the art equipment and a high safety environment enables our doctors to diagnose and treat eye problems accurately, safely and effectively. With a focus to provide best eye care treatment, our eye hospital has always been on preventive ophthalmology and hence our activities are focused at information, education and informed consent at every step. We stand for a healthy vision of life. We are always abreast with the latest and newest in technology wherever required with patient satisfaction and quality of services being our priority.

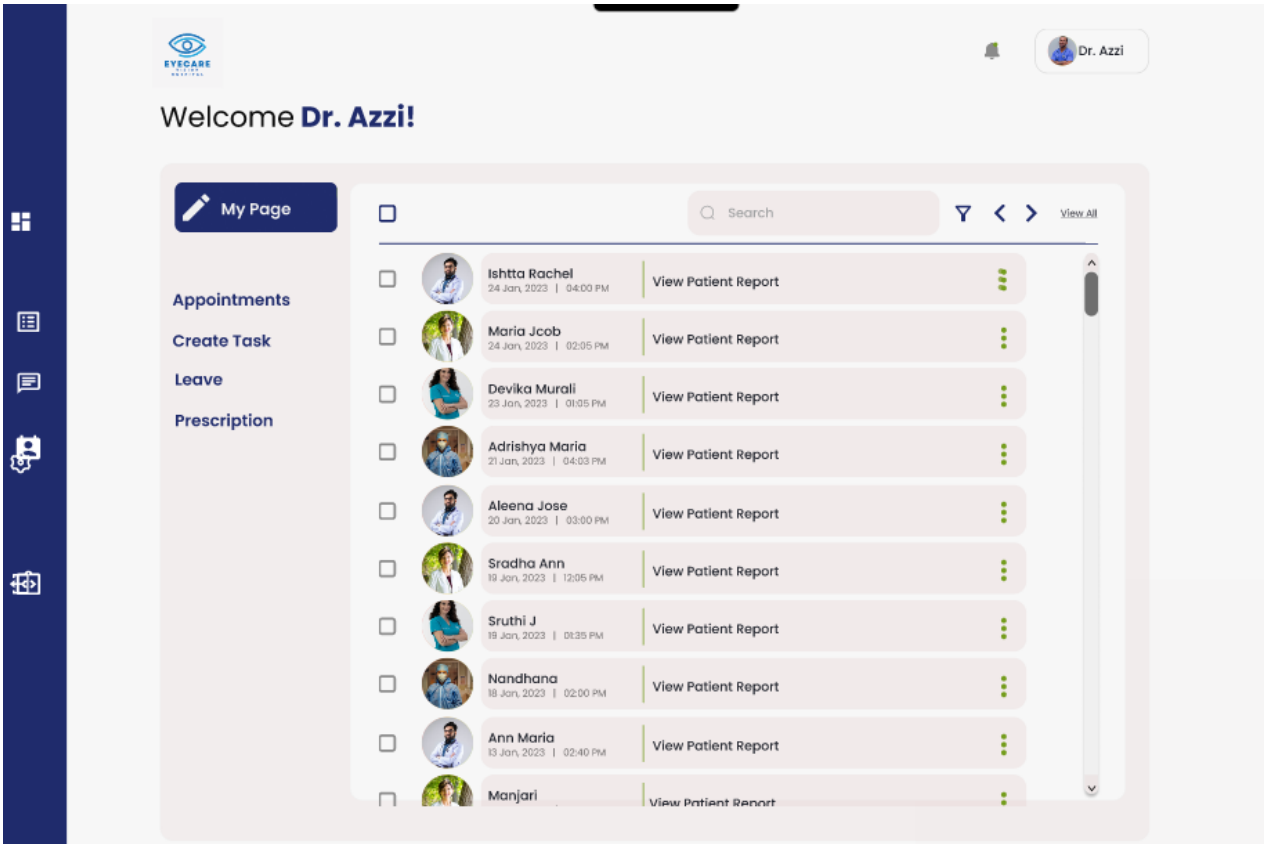
Form Name: Sign in, Sign Up, Forgot Password



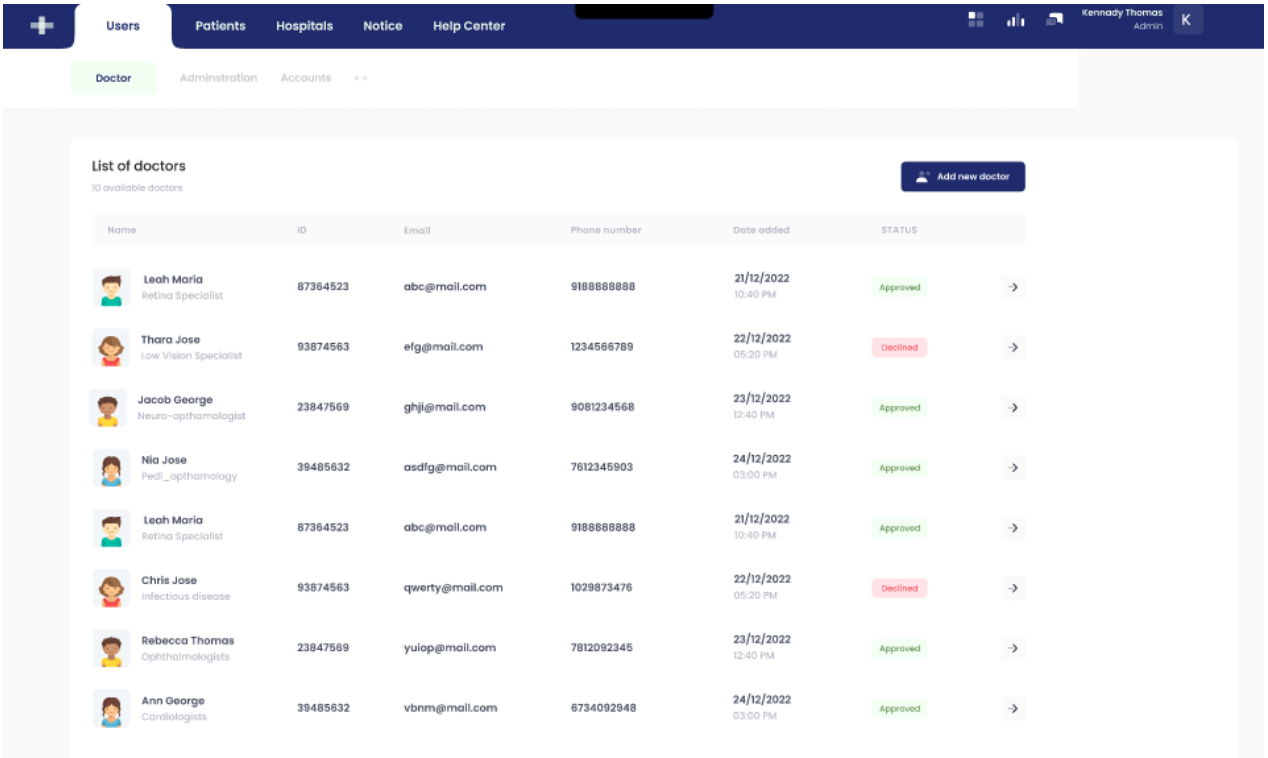
Form Name: Appointment Form

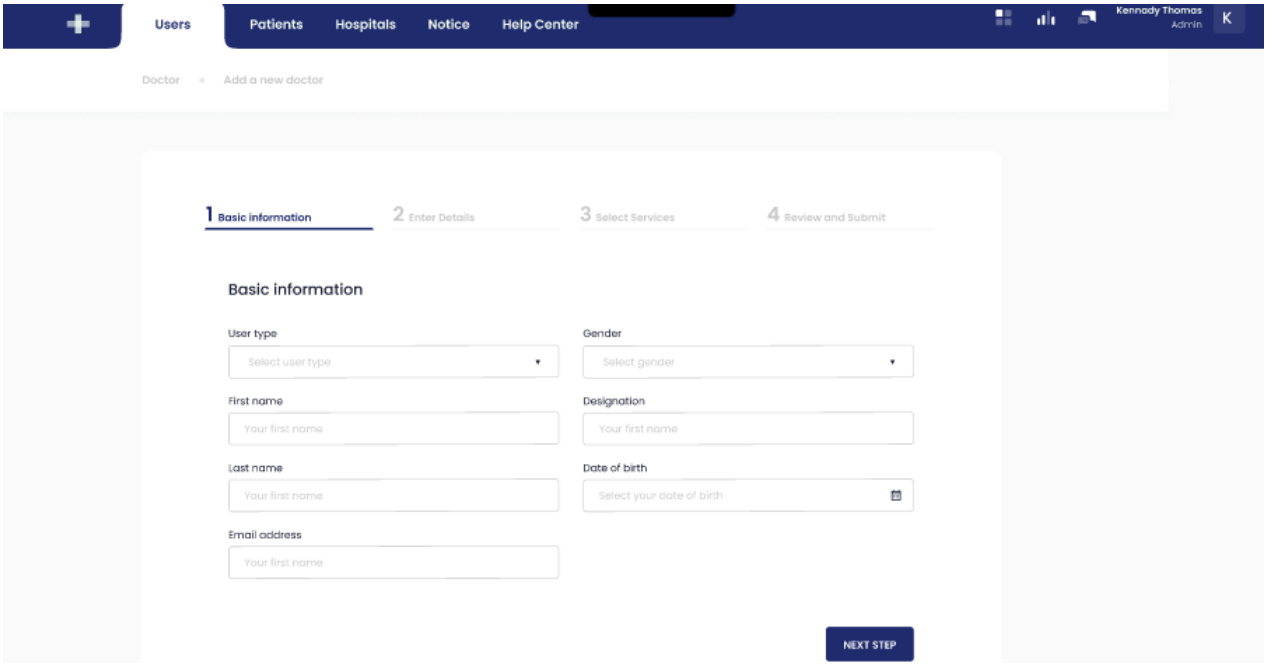


Form Name: Doctors Dashboard

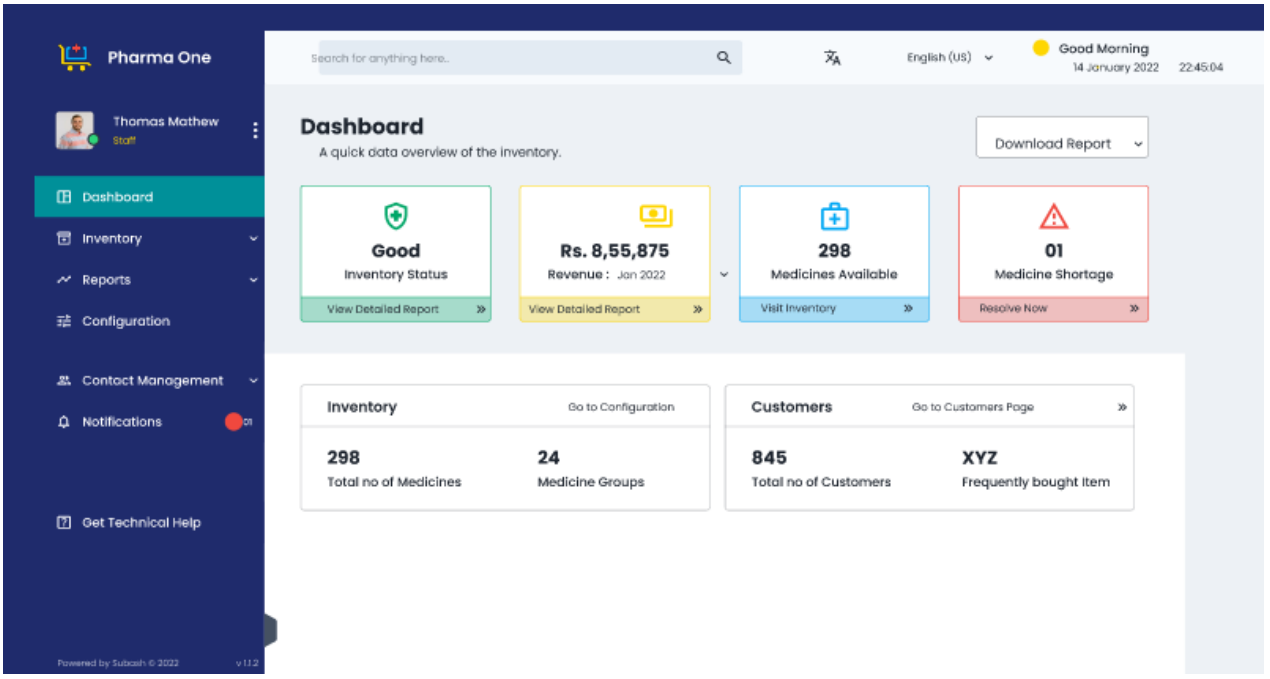


Form Name: Admin Dashboard





Form Name: Pharmacy Dashboard



4.4 DATABASE DESIGN

4.4.1 Relational Database Management System (RDBMS)

A Relational Database Management System (RDBMS) is a database management system that stores and manages data in structured tabular form, comprising rows and columns, with each table representing a distinct entity. RDBMS systems, like MySQL, PostgreSQL, and Oracle, use the structured query language (SQL) to manage data. They excel at maintaining the integrity of data relationships through the use of keys and constraints, allowing for efficient querying and retrieval of information. RDBMS is essential for data-intensive applications, where structured data storage, transaction management, and data integrity are paramount, making it a foundational technology for a wide array of applications, from business systems to web applications.

4.4.2 Normalization

Normalization is a database design process used to minimize data redundancy and maintain data integrity in a relational database. It involves organizing data into smaller, related tables and establishing relationships to ensure that each piece of data is stored in one place and one place only. The key aim of normalization is to eliminate or reduce data anomalies such as update, insert, or delete anomalies, which can lead to inconsistencies in the database. It does this through a series of progressive steps or "normal forms," ensuring that data is structured in a way that minimizes redundancy while maintaining the accuracy and consistency of the information. This structured approach to database design is essential for data management, enabling efficient querying, reducing errors, and optimizing storage efficiency.

4.4.3 Sanitization

Sanitization is a process of cleansing, validating, and securing data to mitigate security risks and prevent potential vulnerabilities. It involves removing or escaping potentially harmful characters or code from input data to ensure that it is safe to use in applications, particularly web applications that interact with user inputs. The primary goal of sanitization is to protect against security threats, such as cross-site scripting (XSS) and SQL injection, by filtering out or neutralizing malicious data before it can be processed or displayed. Sanitization is an essential practice in cybersecurity and web development, as it safeguards applications and data from exploitation, helping maintain the confidentiality and integrity of information.

4.4.4 Indexing

Indexing is a database optimization technique that involves creating data structures, known

as indexes, to accelerate data retrieval and query performance. Indexes are like organized reference guides that point to the location of specific data within a database table. By pre-sorting and storing key columns in an index, databases can quickly locate and retrieve data, reducing the need for full table scans. This process significantly enhances the speed of data retrieval, especially when dealing with large datasets, making it a fundamental practice in database management. However, it comes with trade-offs, such as increased storage requirements and potentially slower data insertion and update operations, so effective indexing strategies involve careful consideration of query patterns and database requirements.

4.5 TABLE DESIGN

1. Table Tbl_login

Primary key: **loginid**

No	Field Name	Data Type	Key Constraints	Description of the field
1	Login_id	INT	Primary Key	Login id of users
2	Username	varchar	Null	Username of users
3	Password	varchar	null	Password of users
4	Status	int	null	status

Table 1: Table Tbl_login

2. Table Tbl_register

Primary key : Reg_id

Foreign Key: Login_id references **Table Tbl_login**

No	Field Name	Data Type	Key Constraints	Description of the field
1	Reg_id	int	Primary key	Register id of users
2	Login_id	int	Foreign Key	Login id of users
3	Email	varchar	null	Email of users
4	Status	int	null	status

Table 2: Table Tbl_register

3. Table Tbl_appointment

Primary key: app_id

Foreign key: reg_id references **Table Tbl_login**,
drslot_id references **Table Tbl_drslot**

No	Field Name	Data Type	Key Constraints	Description of the field
1	App_id	int	Primary key	Appointment id of user
2	Reg_id	int	foreign key	Register id of user
3	Status	int	null	status
4	Name	Varchar	Null	Name of user
5	Address	varchar	Null	Address of user
6	Place	varchar	null	Place of user
7	Dob	date	null	Date of birth of user
8	Mobile number	Varchar	null	Mobile number of user
9	allergy	Varchar	null	Allergy
10	reason	Varchar	null	Reason of consulation
11	date	Date	null	Date
12	Time	Time	null	Time
13	Drslot_id	Int	Foreign Key	Doctor slot id

Table 3: Table Tbl_appointment

4. **Table Tbl_doctor**

Primary key: D_id

Foreign key: login_id references **Table Tbl_login**,
spec_id references **Table Tbl_specialization**

NO	Field Name	Data Type	Key Constraints	Description of field
1	D_id	int	Primary key	Doctor id
2	Login_id	int	Foreign key	Login id
3	Status	int	null	status
4	Name	varchar	null	Name of doctor
6	Dob	varchar	null	Date of birth

7	Address	varchar	null	address
8	Mob no	varchar	null	Mobile number
9	Email	varchar	null	Email of doctor
10	Spec_id	int	Foreign key	Specialization id
11	Ug	varchar	null	Undergraduate education
12	Ug_year	varchar	null	Ug year
13	Pg	varchar	null	Postgraduate
14	Pg_year	varchar	null	Pg year
15	Comments	varchar	null	comments

Table 4: **Table Tbl_doctor**

5. **Table Tbl_drslot**
Primary key: drslot_id
Foreign key: login_id references **Table Tbl_login**

No	Field Name	Data Type	Key Constraints	Description of field
1	Drslot_id	Int	Primary key	Doctor slot id
2	Available	Date	null	Available dates
3	Slot1	Int	null	Slots numbers
4	Slot2	int	null	Slot numbers
5	Slot3	Int	null	Slot numbers
6	Login_id	Int	Foreign key	Login id
7	Status	Int	null	status

Table 5: **Table Tbl_drslot**

6**. Table Tbl_specialization**
Primary key : spec_id
Foreign key: login_id references **Table Tbl_login**

No	Field name	Data type	Key constraints	Description of the key
1	Spec_id	Int	Primary key	Specialization id
2	Status	Int	null	status
3	Spec_name	Varchar	null	Specialization name
4	Login_id	Int	Foreign key	Login id

Table 6: **Table Tbl_specialization**

7. **Table Tbl_recepionist**
Primary key: recep_id
foreign key: login_id references **Table Tbl_login**,
app_id references **Table Tbl_appointment**

No	Field name	Data type	Key constraints	Description of the key
1	Recep_id	Int	Primary key	Receptionist id
2	Status	Int	null	status
3	Login_id	int	Foreign key	Login id
4	Mob no	varchar	null	Mobile numberd
5	Degree	varchar	null	Degree of staff
6	Name	varchar	null	Name of staff
7	Email	varchar	null	Email of staff
8	Year_exp	int	null	Year of experience
9	App_id	int	Foreign key	Appointment id

Table 7: Table Tbl_recepionist

8.**Table Tbl_pharmacy**
Primary key: phar_id
Foreign key:login_id references **Table Tbl_login**

No	Field name	Data type	Key constraints	Description of key
1	Phar_id	Int	Primary key	Pharmacy id
2	Status	Int	null	Status
3	Login_id	Int	Foreign key	Login id
4	Mob no	varchar	Null	Mobile number
5	Degree	varchar	Null	Degree
6	Name	varchar	Null	Name
7	Email	varchar	Null	Email
8	Year_exp	int	Null	Year of experience

Table 8: Table Tbl_pharmacy

9.Table Tbl_Medicine_Category
Primary key: phar_id

No	Field name	Data type	Key constraints	Description of key
1	Medcat_id	Int	Primary key	Medicine Category id
2	Status	Int	null	Status
3	Medcat_name	Int	null	Medicine Category name
4	Medcat_desc	varchar	Null	Medicine Category description

Table 9: Table Tbl_Medicine_Category

10.Table Tbl_Medicine

Primary key: Medcat_id references **Table Tbl_Medicine_Category**

No	Field name	Data type	Key constraints	Description of key
1	Med_id	Int	Primary key	Medicine id
2	Status	Int	null	Status
3	Medcat_id	Int	Foreign Key	Medicine Category references
4	Med_name	varchar	Null	Medicine Category description
5	Med_contains	varchar	null	Medicine contains
6	Med_dos	varchar	null	Medicine dosage informations
7	Med_com	Varchar	Null	Medicine company informations
8	Med_exp	date	Null	Medicine expiry date informations
9	Med_details	Varchar	Null	Medicine informations

Table 10: Table Tbl_Medicine

CHAPTER 5
SYSTEM TESTING

5.1 INTRODUCTION

Software testing is a fundamental and systematic process integral to software development and quality assurance. It encompasses a broad spectrum of activities meticulously designed to confirm that a software application aligns with its intended requirements and functions correctly. The core objective of software testing is to pinpoint and report defects, errors, or inconsistencies within the software's behavior, spanning from functional issues to performance bottlenecks, security vulnerabilities, and usability problems. This methodical process involves the formulation of test cases, which encompass inputs, anticipated outputs, and specific conditions. These cases are executed to assess the software's actual performance, scrutinizing it against expected outcomes. When discrepancies arise, they are categorized as defects, promptly addressed, and corrected to ensure the software's reliability and robustness.

Software testing operates as a multifaceted endeavor woven throughout the software development life cycle. It commences with unit testing, scrutinizing individual components or functions in isolation, and then advances to integration testing, confirming that these components interact seamlessly when integrated. System testing extends the evaluation to the entire system's functionality and often includes stress testing to gauge performance under extreme conditions. Finally, user acceptance testing concludes the process, verifying that the software aligns with user expectations and requirements. This systematic approach to testing holds immense significance. It primarily helps identify and rectify issues early in the development process, preventing defects from propagating to the production environment where resolution can be considerably more expensive. Moreover, it guarantees the software functions reliably, fostering a positive user experience. Software testing is indispensable in meeting regulatory compliance standards, which vary across industries. Overall, this continuous and multifaceted process, ranging from automated techniques to manual inspections by seasoned QA professionals, underpins the development of dependable, robust, and user-safe software applications. It serves as a cornerstone in the pursuit of high-quality software delivery amid a dynamic and competitive technological landscape.

5.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related

data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- Unit testing
- Integration Testing
- Data validation Testing
- Output Testing

5.2.1 Unit Testing

Unit testing is a focused and fundamental aspect of software testing, wherein individual components or units of a software application are rigorously examined in isolation to ensure they function as intended. These units may be functions, methods, or classes, and testing them in isolation helps identify any functional issues or discrepancies. Unit tests are created to assess specific functionalities and behaviors of these components by providing inputs and evaluating their corresponding outputs. The primary objective of unit testing is to pinpoint defects, validate the correctness of each unit's behavior, and ensure that they perform their designated tasks accurately. By addressing issues at this granular level, unit testing contributes to the development of reliable and maintainable software, providing a strong foundation for the broader testing process and reducing the cost and effort required to identify and rectify defects in later stages of development.

5.2.2 Integration Testing

Integration testing is a crucial phase of software testing that focuses on assessing the interactions and collaborations between individual components or modules of a software application. It aims to verify that these components work cohesively and seamlessly when combined. Integration tests identify issues such as communication problems, data flow between modules, and the correct functioning of the application's interfaces. Unlike unit testing, which evaluates components in isolation, integration testing examines how these components interconnect and interact with one another. It plays a pivotal role in uncovering potential integration-related defects, ensuring that the software functions as a unified whole,

and that data and control flow correctly between the integrated parts. Integration testing is instrumental in achieving a robust and dependable software system that meets its intended functionality while maintaining data consistency and reliability across the entire application.

5.2.3 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as BlackBox testing or System tests. Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

5.2.4 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- Input Screen Designs
- Output Screen Designs

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

5.2.5 Automation Testing

Automation testing is a software testing approach that utilizes automated tools and scripts to execute test cases and verify the functionality of an application. It replaces the manual execution of repetitive and time-consuming test scenarios, increasing testing efficiency and accuracy. Automation testing involves the creation of test scripts that mimic user interactions and system behaviors, allowing for repeated and consistent testing of software. It is particularly useful for regression testing, where changes in code are verified against existing functionalities, ensuring that new updates do not introduce defects. Automation testing offers benefits like faster test execution, broader test coverage, and the ability to simulate various

user scenarios. However, it requires an initial investment in script creation and maintenance, making it most advantageous for long-term or frequently updated projects, where it significantly reduces testing time and enhances overall software quality.

5.2.6 Selenium Testing

Selenium testing is a widely used open-source toolset for automating web browser interactions and validating web applications. It offers a comprehensive suite of tools and libraries to create automated test scripts that simulate user interactions with a web application. Selenium allows testers and developers to write test cases in various programming languages and execute them across different web browsers and operating systems. Its versatility and extensibility make it a popular choice for functional and regression testing of web applications. Selenium plays a critical role in automating repetitive testing tasks, ensuring consistent test execution, and accelerating the software testing process, especially in the context of web-based applications and websites.

Example: Test Case 1

Code

```
package LoginJava;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import io.cucumber.java.en.And;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
public class Loginclass {
 WebDriver driver=null;

        @Given("browser is open")
        public void browser_is_open() {
           System.out.println("inside step-browser is open");

System.setProperty("webdriver.gecko.mariomette","D:\\cucumber_selenium\\newart\\src\\test\\res
ources\\Drivers\\geckodriver.exe");
           driver= new FirefoxDriver();
           driver.manage().window().maximize();
        }

        @And("user is on login page")
        public void user_is_on_login_page() throws Exception {
          driver.navigate().to("http://127.0.0.1:8000/dd");
          Thread.sleep(2000);
        }
```

```
@When("user enters username and password")
public void user_enters_username_and_password() throws Exception {
    driver.findElement(By.id("email")).sendKeys("sradha123@gmail.com");
    driver.findElement(By.id("password")).sendKeys("Sradha@123");

}


@When("User click on login")
public void user_click_on_login() {
   // Write code here that turns the phrase above into concrete actions
        driver.findElement(By.id("login-button")).click();
   throw new io.cucumber.java.PendingException();
}


@Then("user is navigated to the home page")
public void user_is_navigated_to_the_home_page() throws Exception {

        driver.findElement(By.id("navbarmain")).isDisplayed();
        Thread.sleep(5000);
        driver.quit();
        driver.close();


}

}
```

Eg.Screenshot

```
Scenario: Check login is successfull with valid credential # src/test/resources/LoginFeature/Loginfile.fe
inside step-browser is open
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
1698192422302   geckodriver     INFO    Listening on 127.0.0.1:41733
1698192422580   mozrunner::runner       INFO    Running command: "C:\\Program Files\\Mozilla Firefox\\fir
console.warn: services.settings: Ignoring preference override of remote settings server
console.warn: services.settings: Allow by setting MOZ_REMOTE_SETTINGS_DEVTOOLS=1 in the environment
1698192423084   Marionette      INFO    Marionette enabled
Dynamically enable window occlusion 0
1698192423218   Marionette      INFO    Listening on port 57272
WebDriver BiDi listening on ws://127.0.0.1:39786
Read port: 57272
1698192423421   RemoteAgent     WARN    TLS certificate errors will be ignored for this session
DevTools listening on ws://127.0.0.1:39786/devtools/browser/8a261131-9b7e-43ed-b073-fc47a59314a8
  Given browser is open                                 # LoginJava.Loginclass.browser_is_open()
  And user is on login page                             # LoginJava.Loginclass.user_is_on_login_page()
  When user enters username and password                # LoginJava.Loginclass.user_enters_username_an
console.warn: LoginRecipes: "Falling back to a synchronous message for: http://127.0.0.1:8000."
JavaScript error: http://127.0.0.1:8000/dd, line 175: ReferenceError: isUsernameValid is not defined
console.warn: LoginRecipes: "Falling back to a synchronous message for: http://127.0.0.1:8000."
JavaScript error: http://127.0.0.1:8000/static/plugins/counterup/jquery.easing.js, line 39: TypeError: jQ
JavaScript error: http://127.0.0.1:8000/static/plugins/shuffle/shuffle.min.js, line 1: TypeError: Shuffle
JavaScript error: https://maps.googleapis.com/maps/api/js?key=AIzaSyAkeLMlsiwzp6b3Gnaxd86lvakimwGA6UA&cal
  And User click on login                               # LoginJava.Loginclass.user_click_on_login()
      io.cucumber.java.PendingException: TODO: implement me
        at LoginJava.Loginclass.user_click_on_login(Loginclass.java:46)
        at *.User click on login(file:///D:/cucumber_selenium/newart/src/test/resources/LoginFeature/Logi

  Then user is navigated to the home page               # LoginJava.Loginclass.user_is_navigated_to_th
```

Test Report

Test Case 1

Project Name: Eyecare Eye Hospital Management System					
Login Test Case					
Test Case ID: Test_1			**Test Designed By: Liya K Joseph**		
Test Priority(Low/Medium/High):High			**Test Designed Date: 10-10-2023**		
Module Name: Login Screen			**Test Executed By: Sona Maria Sebastin**		
Test Title: verify login with valid email and password			**Test Execution Date: 10-10-2023**		
Description: Test the login page					
Pre-Condition: User has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/ Fail)**
1	Navigation to Login Page		Login Page should be displayed	Login page displayed	Pass
2	Provide Valid username	Provide Valid username sradha123@gmail.com	User should able to login	User should navigate to hospital website	Pass
3	Provide valid password	Password Sradha@123			
4	Click on Sign In button	Submit details			
Post-Condition: Patient successfully logined to hospital website					

Test Case 2:

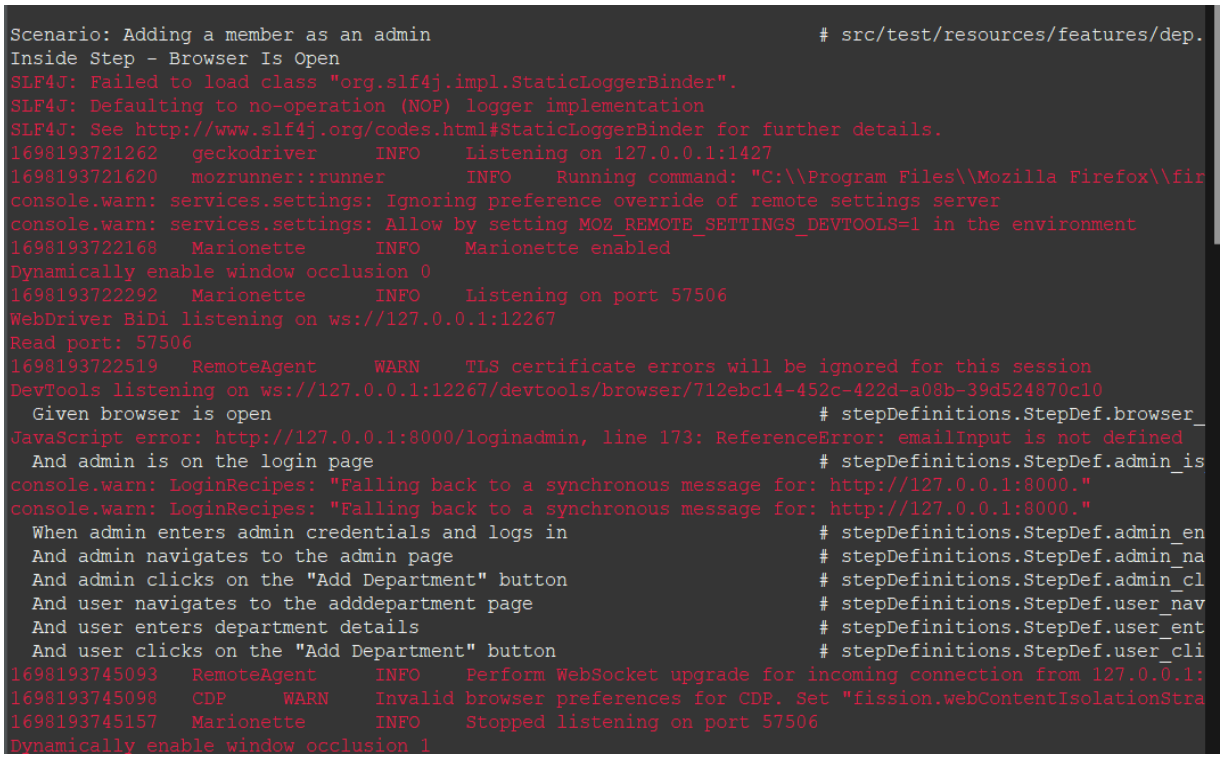
Code

```
package stepDefinitions;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.When;
import io.cucumber.java.en.And;
import io.cucumber.java.en.Then;
import org.openqa.selenium.firefox.FirefoxDriver;
public class StepDef {
private WebDriver driver; // Initialize your WebDriver instance
```

```
@Given("browser is open")
public void browser_is_open() {
System.out.println("Inside Step - Browser Is Open");
System.setProperty("webdriver.gecko.marionette",
"D:\\cucumber_selenium\\adddepartment\\src\\test\\resources\\drivers\\geckodr
iver.exe");
driver = new FirefoxDriver();
driver.manage().window().maximize();
}
@And("admin is on the login page")
public void admin_is_on_the_login_page() throws Exception{
driver.navigate().to("http://127.0.0.1:8000/loginadmin");
Thread.sleep(3000);
}
@When("admin enters admin credentials and logs in")
public void admin_enters_admin_credentials_and_logs_in()throws Throwable
{
driver.findElement(By.name("email")).sendKeys("admin3@gmail.com");
driver.findElement(By.name("password")).sendKeys("admin3");
driver.findElement(By.id("login")).click();
Thread.sleep(3000);
}
@And("admin navigates to the admin page")
public void admin_navigates_to_the_admin_page()throws Exception
{

Thread.sleep(3000);
driver.findElement(By.id("dep")).click();
}

@And("admin clicks on the \"Add Department\" button")
public void admin_clicks_on_the_add_department_button() throws
InterruptedException {
// Click on the "Manage" button
driver.findElement(By.id("add_dep")).click();
Thread.sleep(3000);
}
@And("user navigates to the adddepartment page")
public void user_navigates_to_the_adddepartment_page() throws
InterruptedException {
Thread.sleep(3000);
}
@And("user enters department details")
public void user_enters_department_details() throws InterruptedException {
// Enter category details
driver.findElement(By.name("depp")).sendKeys("Oculoplasty");
driver.findElement(By.name("desc")).sendKeys("plastic surgery of the eye and
adjacent parts (such as the tear ducts or eyelids)");
}
@And("user clicks on the \"Add Department\" button")
public void user_clicks_on_the_add_department_button(){
```

```
driver.findElement(By.id("create_dep")).click();
}
@Then("department should be added and displayed on the department page")
public void
department_should_be_added_and_displayed_on_the_department_page()
throws InterruptedException {       // Implement code to verify that the added
category is displayed on the member page
driver.findElement(By.id("deplist")).isDisplayed();
Thread.sleep(2000);
driver.close();
driver.quit();
}
}
```

Screenshot



Test Report

Test Case 2

Project Name: Eyecare Eye Hospital Management System	
Department Test Case	
Test Case ID: Test_2	**Test Designed By: Liya K Joseph**
Test Priority(Low/Medium/High):High	**Test Designed Date: 10-10-2023**
Module Name: Admin Dashboard	**Test Executed By : Sona Maria Sebastin**
Test Title : verify login with valid email and password and add new department	**Test Execution Date: 10-10-2023**

Description: Test the login Page and add department					
Pre-Condition: User has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/ Fail)**
1	Navigation to Login Page		Login Page should be displayed	Login page displayed	Pass
2	Provide Valid username	Provide Valid username admin3@gmail.com	Usershould able to login	User should navigated to admin dashboard	Pass
3	Provide valid password	Password admin3			
4	Click on Sign In button	Enter details			
5	Admin adds department name and its description	Department name: Oculoplasty descriptions: plastic surgery of the eye	Display the details	Department details should be displayed	pass
Post-Condition: A new department added by admin					

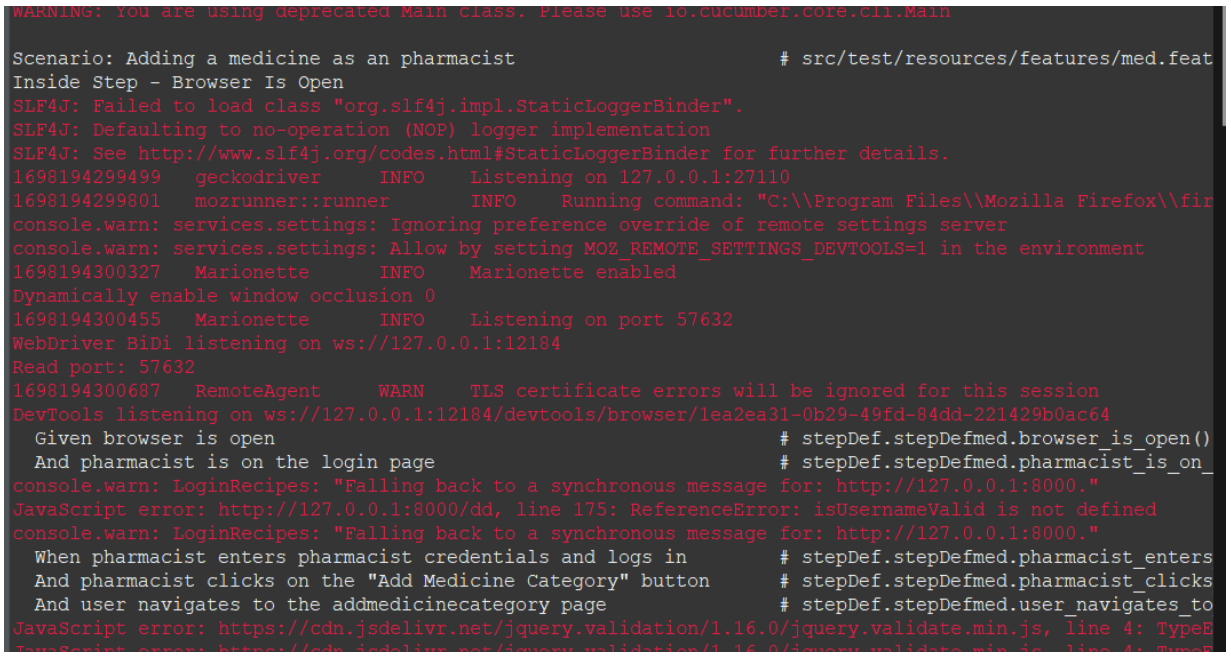
Test Case 3:

Code

```
package stepDef;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import io.cucumber.java.en.And;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
public class stepDefmed {
private WebDriver driver; // Initialize your WebDriver instance
        @Given("browser is open")
        public void browser_is_open() {
        System.out.println("Inside Step - Browser Is Open");
        System.setProperty("webdriver.gecko.marionette",
        "D:\\cucumber_selenium\\medicine\\src\\test\\resources\\drivers\\geckodriver.e
        xe");
        driver = new FirefoxDriver();
        driver.manage().window().maximize();
        }
        @And("pharmacist is on the login page")
```

```
public void pharmacist_is_on_the_login_page() throws Exception{
driver.navigate().to("http://127.0.0.1:8000/dd");
Thread.sleep(3000);
}
@When("pharmacist enters pharmacist credentials and logs in")
public void pharmacist_enters_admin_credentials_and_logs_in()throws
Throwable {
driver.findElement(By.name("email")).sendKeys("aleenatom2024@gmail.com
");
driver.findElement(By.name("password")).sendKeys("Aleenatom@12345");
driver.findElement(By.id("login-button")).click();
Thread.sleep(3000);
}

@And("pharmacist clicks on the \"Add Medicine Category\" button")
public void pharmacist_clicks_on_the_add_category_button() throws
InterruptedException {
// Click on the "Manage" button
driver.findElement(By.id("add_medcat")).click();
Thread.sleep(3000);
}
@And("user navigates to the addmedicinecategory page")
public void user_navigates_to_the_addmedicinecategory_page() throws
InterruptedException {
Thread.sleep(3000);
}
@And("user enters medicinecategory details")
public void user_enters_medicinecategory_details() throws
InterruptedException {
// Enter category details
driver.findElement(By.name("category_name")).sendKeys("Eye Pain Relief");
driver.findElement(By.name("description")).sendKeys("Eye pain relief refers
to the alleviation of discomfort or pain experienced in or around the eyes.");
}
@And("user clicks on the \"Add Category\" button")
public void user_clicks_on_the_add_category_button(){
driver.findElement(By.id("create")).click();
}
@Then("category should be added and displayed on the category page")
public void
category_should_be_added_and_displayed_on_the_category_page() throws
InterruptedException {       // Implement code to verify that the added category
is displayed on the member page
driver.findElement(By.id("list")).isDisplayed();
Thread.sleep(2000);

driver.close();

driver.quit();
}
}
```

Screenshot



Test report

Test Case 3

Project Name: Eyecare Eye Hospital Management System					
Medicine Category Test Case					
Test Case ID: Test_3			**Test Designed By: Liya K Joseph**		
Test Priority(Low/Medium/High):High			**Test Designed Date: 10-10-2023**		
Module Name: Pharmacist Dashboard			**Test Executed By: Sona Maria Sebastin**		
Test Title: verify login with valid email and password and add new medicine category			**Test Execution Date: 10-10-2023**		
Description: Test the login Page and add category					
Pre-Condition: User has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**ActualResult**	**Status (Pass/ Fail)**
1	Navigation to Login Page		Login Page should be displayed	Login page displayed	Pass
2	Provide Valid username	Provide Valid username aleenatom2024 @gmail.com	User should able to	User should navigated to pharmacist dashboard	Pass

3	Provide valid password	Password Aleenatom@12345	login		
4	Click on Sign In button	Enter details			
5	Pharmacist adds new medicine category name and its description	Department name: Eye Pain Relief descriptions: its details	Display the details	Medicine Category name and its details should be displayed	pass
Post-Condition: Anew medicine category added by pharmacists staff					

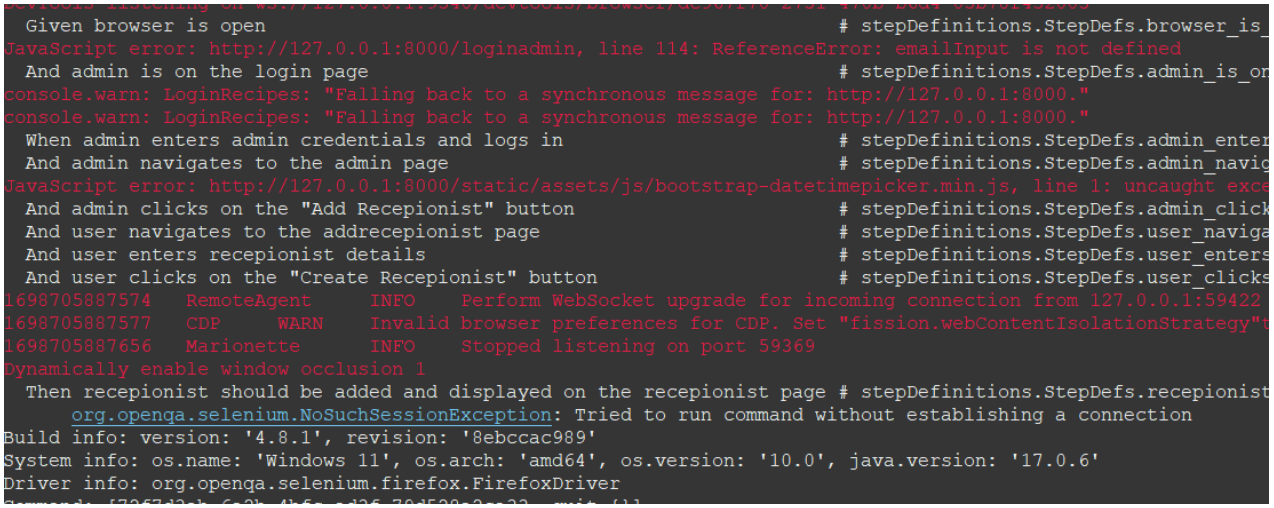
Test Case 4

Code

```
package stepDefinitions;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import io.cucumber.java.en.And;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
public class StepDefs {
private WebDriver driver; // Initialize your WebDriver instance

        @Given("browser is open")
        public void browser_is_open() {
        System.out.println("Inside Step - Browser Is Open");
        System.setProperty("webdriver.gecko.marionette",
        "D:\\cucumber_selenium\\staff\\src\\test\\resources\\drivers\\geckodriver.exe");
        driver = new FirefoxDriver();
        driver.manage().window().maximize();
        }
        @And("admin is on the login page")
        public void admin_is_on_the_login_page() throws Exception{
        driver.navigate().to("http://127.0.0.1:8000/loginadmin");
        Thread.sleep(3000);
        }
        @When("admin enters admin credentials and logs in")
        public void admin_enters_admin_credentials_and_logs_in()throws Throwable {
        driver.findElement(By.name("email")).sendKeys("admin3@gmail.com");
        driver.findElement(By.name("password")).sendKeys("admin3");
        driver.findElement(By.id("login")).click();
        Thread.sleep(3000);
        }
```

```
@And("admin navigates to the admin page")
public void admin_navigates_to_the_admin_page()throws Exception
{

Thread.sleep(3000);
driver.findElement(By.id("rep")).click();
}

@And("admin clicks on the \"Add Recepionist\" button")
public void admin_clicks_on_the_add_recepionist_button() throws InterruptedException {
// Click on the "Manage" button
driver.findElement(By.id("add_rep")).click();
Thread.sleep(3000);
}
@And("user navigates to the addrecepionist page")
public void user_navigates_to_the_addrecepionist_page() throws InterruptedException {
Thread.sleep(3000);
}
@And("user enters recepionist details")
public void user_enters_recepionist_details() throws InterruptedException {
// Enter category details
driver.findElement(By.name("Name")).sendKeys("Leah Thomas");
driver.findElement(By.name("email")).sendKeys("leahthomas2@gmail.com");
driver.findElement(By.name("password")).sendKeys("Leah@12345");
driver.findElement(By.name("phn")).sendKeys("9087652807");
}
@And("user clicks on the \"Create Recepionist\" button")
public void user_clicks_on_the_create_recepionist_button(){
driver.findElement(By.id("create_rep")).click();
}
@Then("recepionist should be added and displayed on the recepionist page")
public void recepionist_should_be_added_and_displayed_on_the_recepionist_page()
throws InterruptedException {       // Implement code to verify that the added category is
displayed on the member page
driver.findElement(By.id("replist")).isDisplayed();
Thread.sleep(2000);

driver.close();

driver.quit();
}
}
```

Screenshot



Test Case 4

Project Name: Eyecare Eye Hospital Management System					
Add Recepionists Test Case					
Test Case ID: Test_4			**Test Designed By: Liya K Joseph**		
Test Priority (Low/Medium/High): High			**Test Designed Date: 10-10-2023**		
Module Name: Admin Dashboard			**Test Executed By: Sona Maria Sebastin**		
Test Title: verify login with valid email and password and add new receptionist staff details			**Test Execution Date: 10-10-2023**		
Description: Test the login Page and add staff					
Pre-Condition: User has valid username and password					
Step	**Test Step**	**Test Data**	**Expecte dResult**	**Actual Result**	**Status (Pass/ Fail)**
1	Navigation to Login Page		Login Page should be displayed	Login page displayed	Pass
2	Provide Valid username	Provide Valid username admin3@gmail.com	User should able to login	User should navigated to admin dashboard	Pass
3	Provide valid password	Password admin3			
4	Click on Sign In button	Details submitted			
5	Admin adds new receptionists	Name: Leah Thomas mail: leahthomas2@gmail.co	Display the details	Staff details added and displayed	pass

	staff details	m password: Leah@12345 phno: 9087652807			
Post-Condition: A new receptionist staff added by admin.					

CHAPTER 6
IMPLEMENTATION

6.1 INTRODUCTION

The project's implementation phase is where the conceptual design is transformed into a functional system. Gaining users' confidence that the new system will function as intended and be useful might be seen as the most important stage in creating a successful new system both efficient and precise. User documentation and training are its main concerns. Usually, conversion happens either during or after the user's training. Implementation is the simple act of putting a new system design into use, and it involves the creating an operational version of a newly changed system design. At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion. Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

➢ Careful planning.

➢ Investigation of system and constraints.

➢ Design of methods to achieve the changeover.

These are the tasks included in the implementation.

6.2 IMPLEMENTATION PROCEDURES

Software implementation involves the final deployment of the software package in its actual working environment, ensuring it meets the requirements and expectations of its intended users while operating seamlessly. In many organizations, the responsibility for commissioning the software development project falls to individuals who will not be directly using the software. During the initial stages, doubts may arise about the software,

but it's essential to prevent resistance from building up. This entails ensuring that:

- Active users understand the benefits of the new system.
- Confidence in the software is established.
- Adequate user guidance is provided to ensure comfort with using the application.

Before evaluating the system, users must be informed that the server software must be active on the server to access the results. The actual process will not proceed if the server object is not functioning on the server.

6.2.1 User Training

User training serves the crucial purpose of preparing individuals to effectively test and make necessary modifications to the system. It's essential that participants have confidence in their roles within the new system to realize the expected benefits of a computer-based solution. The complexity of systems often necessitates more comprehensive training. During this training, users gain proficiency in various tasks, including data entry, handling error notifications, database queries, report generation, and other essential functions.

6.2.2 Training on the Application Software

User training commences with a foundational computer awareness program, followed by instruction on the newly introduced application software. This training elucidates the fundamental principles of using the system, covering screen navigation, available help resources, common data entry errors, validation procedures, and methods for modifying entered data. Subsequently, specialized program training tailors the content to the specific needs of user groups or individuals, considering factors such as hierarchy and roles within the organization. Training content may vary based on user group and the level of expertise required.

6.2.3 System Maintenance

The enigma of software development lies in its maintenance phase. In this stage, a software product is actively operational. Properly maintaining a system post-implementation is an integral component of the software development life cycle. Maintenance is essential to ensure that a system remains adaptable to changes in its environment. Software maintenance involves more than just error correction; it encompasses the ongoing health and optimization of the system.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

My project is only a humble venture to satisfy the needs to manage the eye hospital management works. The proposed system provides all functionalities of the existing system along with new features. The patient user can select a slot and doctor as per the doctor's availability and patient can view their previous records. The doctor can set their schedules for the next day and they can accept or reject the online appointments it will notify the corresponding patient with email. The staff can view the available doctors and their approved appointments they can add record on them.

7.2 FUTURE SCOPE

A modern health technology system can greatly improve patient care and convenience. This could include a patient portal for accessing medical records and billing online. Telemedicine allows for remote consultations through video chat or other methods. Medication tracking and refill reminders help patients adhere to treatment plans. Blogs and online resources can provide patient education. A mobile app makes accessing these services easier when on the go. Behind the scenes, a referral management system streamlines the process of connecting patients to specialists. Automated inventory management saves time and reduces waste. Supporting multiple languages ensures these solutions are accessible to all patients. Implementing these technologies thoughtfully can enhance healthcare delivery and the patient experience.

.

CHAPTER 8
BIBLIOGRAPHY

REFERENCES:

- https://prezi.com/hmtyiwvfjm1d/eye-clinic-management-system/
- https://www.freeprojectz.com/premium-synopsis/synopsis-eye-care-appointment-system
- https://github.com/sumitkumar1503/hospitalmanagement
- https://eyecarecomkottayam.myomni.in/
- Grady Booch, James Rumbaugh, Ivar Jacobson, "Unified Modeling Language User
- Guide"Alistair Cockburn, "Writing Effective Use Cases"

WEBSITES:

- https://www.djangoproject.com/
- https://www.w3schools.com/django/
- https://www.codecademy.com/learn/learn-css
- https://chat.openai.com/chat
- www.jquery.com

CHAPTER 9

APPENDIX

9.1 SAMPLE CODE

Admin add doctor

```
<!DOCTYPE html>
{% load static %}
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-
scalable=0">
<link rel="shortcut icon" type="image/x-icon" href="{% static
'assets/img/favicon.ico'%}">
<title>eye</title>
<link rel="stylesheet" type="text/css" href="{% static
'assets/css/bootstrap.min.css'%}">
<link rel="stylesheet" type="text/css" href="{% static 'assets/css/font-
awesome.min.css'%}">
<link rel="stylesheet" type="text/css" href="{% static
'assets/css/select2.min.css'%}">
<link rel="stylesheet" type="text/css" href="{% static 'assets/css/bootstrap-
datetimepicker.min.css'%}">
<link rel="stylesheet" type="text/css" href="{% static 'assets/css/style.css'%}">
</head>
<body>
<div class="main-wrapper" style="padding-left:8%;">
<div class="sidebar" id="sidebar">
<div class="sidebar-inner slimscroll">
<div id="sidebar-menu" class="sidebar-menu">
<ul>
<li class="menu-title">Main</li>
<li>
<a href="index-2.html"><i class="fa fa-dashboard"></i> <span>Dashboard</span></a>
</li>
<li class="active">
<a href=" {% url 'admin_doctors' %}"><i class="fa fa-user-md"></i>
<span>Doctors</span></a>
</li><li>
<a href="{% url 'departments' %}"><i class="fa fa-hospital-o"></i>
<span>Departments</span></a>
</li>
<li>
<a href="{% url 'rep' %}"><i class="fa fa-hospital-o"></i>
<span>Recepionists</span></a>
</li>
</div></div></div>
<div class="page-wrapper">
<div class="content">
<div class="row">
<div class="col-lg-8 offset-lg-2">
```

```
<h4 class="page-title">Add Doctor</h4>
</div>
</div>
<div class="row">
<div class="col-lg-8 offset-lg-2">
<form method='post' action="{% url 'admin_adddoctor' %}" id="adddoc" role="form"
onsubmit="return validateForm()" nonvalidate="nonvalidate" class="bv-form">
{% csrf_token %}
<div class="row">
<div class="col-sm-6">
<div class="form-group">
<label> Name <span class="text-danger">*</span></label>
<input class="form-control" type="text" name="Name" id="fname">
</div>
</div>
<div class="col-sm-6">
<div class="form-group">
<label>Email <span class="text-danger">*</span></label>
<input class="form-control" type="email" name="email" id="email">
</div>
</div>
<div class="col-sm-6">
<div class="form-group">
<label>Password</label>
<input class="form-control" type="password" name="password" >
</div>
</div>
<div class="col-sm-6 col-md-6 col-lg-3">
<div class="form-group">
<label>Departments</label>
<select class="form-control select" name="depp">
{% for department in depts %}
<option value={{department.Dep_id}}>{{ department.Dep_name }}</option>
{% endfor %}
</select>
</div>
</div>
<div class="col-sm-6">
<div class="form-group">
<label>Phone </label>
<input class="form-control" type="text" name="phn" id="phn">
</div>
</div>
<div class="m-t-20 text-center">
<button type="submit" class="btn btn-primary submit-btn">Create Doctor</button>
</div>
</form></div>
</div></div>
</div></div>
```

```
<div class="sidebar-overlay" data-reff=""></div>
<script src="{% static 'assets/js/jquery-3.2.1.min.js'%}"></script>
<script src="{% static 'assets/static/assets/js/popper.min.js'%}"></script>
<script src="{% static 'assets/static/assets/js/bootstrap.min.js'%}"></script>
<script src="{% static 'assets/static/assets/js/jquery.slimscroll.js'%}"></script>
<script src="{% static 'assets/static/assets/js/select2.min.js'%}"></script>
<script src="{% static 'assets/static/assets/js/moment.min.js'%}"></script>
<script src="{% static 'assets/js/bootstrap-datetimepicker.min.js' %} "></script>
<script src="{% static 'assets/static/assets/js/app.js'%}"></script>
<script>
function validateForm() {
// Get form elements
var nameField = document.getElementById("fname");
var emailField = document.getElementById("email");
var phoneField = document.getElementById("phn");
// Regular expressions for validation
var namePattern = /^[A-Za-z\s]+$/;
var emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
var phonePattern = /^[0-9]+$/;
// Validation logic
if (!namePattern.test(nameField.value)) {
alert("Name should only contain characters and spaces.");
nameField.focus();
return false;
}
if (!emailPattern.test(emailField.value)) {
alert("Email is not valid.");
emailField.focus();
return false;
}
if (!phonePattern.test(phoneField.value)) {
alert("Phone should only contain numbers.");
phoneField.focus();
return false;
}
// If all validations pass, the form will be submitted
return true;
}
</script>
</body>
</html>
```

Appointment form

```
<!DOCTYPE html>
<html>
<head>
<title>Book an Appointment</title>
<!-- Include Bootstrap CSS -->
```

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
alpha1/dist/css/bootstrap.min.css">
<style>
body {
background-image: url(' https://www.eye-care-hospital.com/images/slide-2.jpg'); /*
Replace with your background image URL */
background-size: cover;
background-repeat: no-repeat;
background-attachment: fixed;
color: #fff; /* Text color */
}
.container {
background-color: rgba(0, 0, 0, 0.7); /* Background color with transparency */
padding: 20px;
border-radius: 10px;
margin-top: 20px;
}
.form-label {
font-weight: bold;
}
.form-control {
background-color: rgba(255, 255, 255, 0.1); /* Input field background color with
transparency */
color: #fff; /* Input text color */
}
.btn-primary {
background-color: #007bff; /* Primary button color */
border-color: #007bff;
}
.btn-primary:hover {
background-color: #0056b3; /* Button color on hover */
}
</style>
</head>
<body>
<div class="container">
<h1 class="mt-5">Book an Appointment</h1>
<form method="POST" >
{% csrf_token %}
<div class="mb-3">
<label for="name" class="form-label">Name:</label>
<input type="text" class="form-control" id="name" name="name" required>
</div>
<div class="mb-3">
<label for="address" class="form-label">Address:</label>
<input type="text" class="form-control" id="address" name="address" required>
</div>
<div class="mb-3">
<label for="place" class="form-label">Place:</label>
```

```
<input type="text" class="form-control" id="place" name="place" required>
</div>
<div class="mb-3">
<label for="dob" class="form-label">Date of Birth:</label>
<input type="date" class="form-control" id="dob" name="dob" required>
</div>
<div class="mb-3">
<label for="gender" class="form-label">Gender:</label>
<select class="form-select" id="gender" name="gender">
<option value="Male" {% if user_profile.gender == 'Male' %} selected {% endif
%}>Male</option>
<option value="Female" {% if user_profile.gender == 'Female' %} selected {% endif
%}>Female</option>
<option value="Others" {% if user_profile.gender == 'Other' %} selected {% endif
%}>Others</option>
</select>
</div>
<div class="mb-3">
<label for="mobile" class="form-label">Mobile Number:</label>
<input type="text" class="form-control" id="mobile" name="mobile" required>
</div><div class="mb-3">
<label for="allergy" class="form-label">Do you have allergies?</label>
<div class="form-check">
<input class="form-check-input" type="radio" name="allergy" id="allergy_yes"
value="Yes" required>
<label class="form-check-label" for="allergy_yes">Yes</label>
</div>
<div class="form-check">
<input class="form-check-input" type="radio" name="allergy" id="allergy_no" value="No"
required>
<label class="form-check-label" for="allergy_no">No</label>
</div>
</div>
<div class="mb-3">
<label for="reason" class="form-label">Reason for Consultation:</label>
<textarea class="form-control" id="reason" name="reason" required></textarea>
</div>
<div class="mb-3">
<label for="doctor" class="form-label">Select a Doctor:</label>
<select class="form-select" id="doctor" name="doctor" required>
<option value="" disabled>Select a Doctor</option>
{% for doctor in doctors %}
<option value="{{ doctor.id }}" {% if selected_doctor == doctor.id %} selected {%
endif %}>{{ doctor.Name }}</option>
{% endfor %}
</select>
</div>
<div class="mb-3">
<label for="date" class="form-label">Select a Date:</label>
```

```
<select class="form-select" id="date" name="date" required>
<option value="" >Select a Date</option>
{% for date in date_options %}
<option value="{{ date.date }}">{{ date.date }}</option>
{% endfor %}
</select>
</div>
<div class="mb-3">
<label for="time" class="form-label">Select a Time Slot:</label>
<select class="form-select" id="time" name="time" required>
<option value="" disabled selected>Select a Time Slot</option>
{% for option in time_options %}
<option value="{{ option.id }}">{{ option.text }}</option>
{% endfor %}
</select>
</div>
<button type="submit" class="btn btn-primary">Book Appointment</button>
</form>
<!-- Display error messages if needed -->
{% if error_message %}
<div class="alert alert-danger mt-3">{{ error_message }}</div>
{% endif %}
<!-- Display success messages if needed -->
{% if success_message %}
<div class="alert alert-success mt-3">{{ success_message }}</div>
{% endif %}
</div>
<!-- Include Bootstrap JS (if needed) -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
alpha1/dist/js/bootstrap.min.js"></script>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<!-- Include jQuery library -->
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<!-- Add this script after including jQuery -->
<!-- Include Bootstrap JS (if needed) -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
alpha1/dist/js/bootstrap.min.js"></script>
<!-- Include jQuery library -->
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<!-- Add this script after including jQuery -->
<script>
// Function to update the date dropdown based on the selected doctor
$("#doctor").change(function() {
var selectedDoctor = $(this).val();
if (selectedDoctor) {
// Send an AJAX request to fetch dates for the selected doctor
$.ajax({
url: '/get_dates/' + selectedDoctor + '/',  // Replace with your URL
type: 'GET',
```

```
success: function(data) {
// Update the date dropdown with fetched data
$("#date").html(data.date_options);
}
});
} else {
// If no doctor is selected, reset the date dropdown
$("#date").html('<option value="" disabled selected>Select a Date</option>');
}
});
$("#date").change(function() {
var selectedDate = $(this).val();
var selectedDoctor = $("#doctor").val();
if (selectedDate && selectedDoctor) {
// Send an AJAX request to fetch time slots for the selected date and doctor
$.ajax({
url: `/get_times/${selectedDoctor}/${selectedDate}/`,  // Use template literals to
construct the URL
type: 'GET',
success: function(data) {
// Update the time dropdown with fetched data
var timeDropdown = $("#time");
timeDropdown.empty();  // Clear existing options
timeDropdown.append('<option value="" disabled selected>Select a Time Slot</option>');
$.each(data.time_options, function(index, option) {
timeDropdown.append($('<option>', {
value: option.id,
text: option.text
}));
});
},
error: function() {
// Handle any AJAX errors here
console.log("Error fetching time slots.");
}
});
} else {
// If no date or doctor is selected, reset the time dropdown
$("#time").html('<option value="" disabled selected>Select a Time Slot</option>');
}
});
</script>
</body>
</html>
```

Delete doctor

```
{% load static %}
<!DOCTYPE html>
```

```
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Delete Doctor</title>
<!-- Add Bootstrap CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<!-- Add your custom CSS -->
<style>
body {
padding: 20px;
}
h1 {
font-size: 24px;
}
.container {
max-width: 400px;
}
.btn-danger {
background-color: #dc3545;
border-color: #dc3545;
}
.btn-danger:hover {
background-color: #c82333;
border-color: #bd2130;
}
</style>
</head>
<body>
<div class="container">
<h1 class="mt-4">Delete Doctor</h1>
<p class="mt-4">Are you sure you want to delete the following doctor?</p>
<div class="mt-4">
<p><strong>Doctor Name:</strong> {{ doctor.Name }}</p>
</div>
<form method="post" class="mt-4">
{% csrf_token %}
<button type="submit" class="btn btn-danger">Delete</button>
<a href="{% url 'admin_doctors' %}" class="btn btn-secondary ml-2">Cancel</a>
</form>
</div>
<!-- Add Bootstrap JS (optional) -->
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></scri
pt>
```

```
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>
```

View medicine category

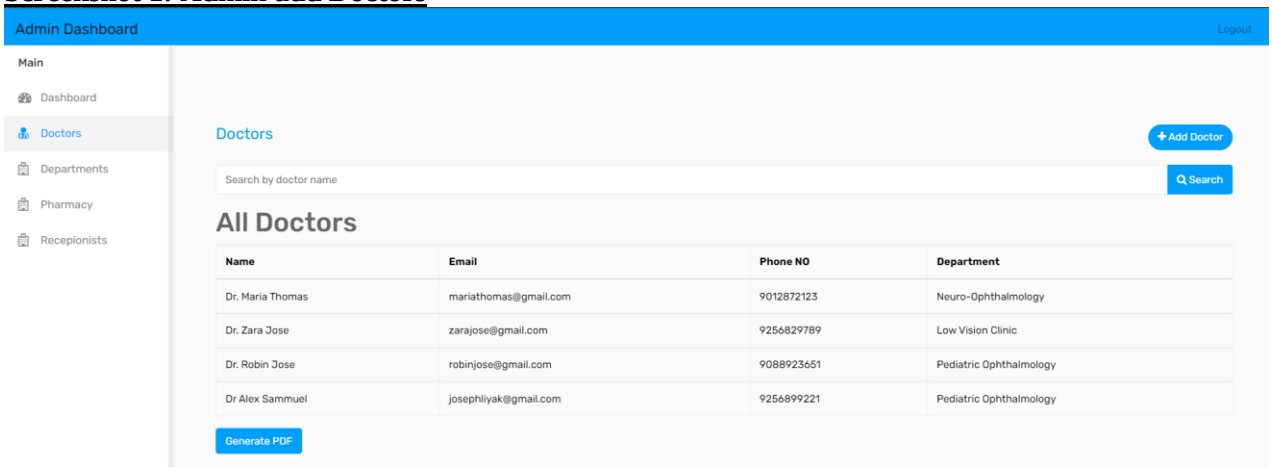
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Medicine Categories</title>
<!-- Include Bootstrap CSS -->
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap/dist/css/bootstrap.min.css">
<!-- Custom CSS for styling -->
<style>
body {
background-image:
url('https://i.pinimg.com/736x/92/70/c1/9270c19eb7d37366c20da05e28844248.jpg'); /*
Replace with your image path */
background-size: cover;
background-repeat: no-repeat;
background-attachment: fixed;
}
.container {
background-color: rgba(255, 255, 255, 0.8); /* Semi-transparent white background for
the form */
padding: 20px;
border-radius: 10px;
margin-top: 20px;
}
.navbar {
background-color: black;
}
table {
width: 100%;
border-collapse: collapse;
}
th, td {
border: 1px solid black;
padding: 8px;
text-align: left;
}

th {
background-color: #f2f2f2;
}
</style>
```

```
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-dark bg-black">
<div class="container-fluid">
<a class="navbar-brand" href="#">Medicine Categories</a>
<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav"
aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarNav">
<ul class="navbar-nav ml-auto">
<li class="nav-item">
<a class="nav-link" href="{% url 'phar_staff_page' %}">Home</a>
</li>
<li class="nav-item">
<a class="nav-link" href="{% url 'add_medicine_category' %}">Add Category</a>
</li>
<li class="nav-item">
<a class="nav-link" href="{% url 'view_medicine_category' %}">View Categories</a>
</li>
</ul>
</div>
</div>
</nav>
<div class="container">
<h2 class="text-center" id="list">Medicine Categories</h2>
<form id="search-form" method="GET" action="{% url 'search_medicine_category' %}">
<div class="input-group mb-3">
<input type="text" class="form-control" name="category_name" placeholder="Search by
medicine category name">
<div class="input-group-append">
<button class="btn btn-primary" type="submit"><i class="fa fa-search"></i>
Search</button>
</div>
</div>
</form>
<table class="table">
<thead>
<tr>
<th scope="col">Category Name</th>
<th scope="col">Description</th>
<th scope="col">Actions</th>
</tr>
</thead>
<tbody>
{% for i in categories %}
<tr>
<td>{{ i.category_name }}</td>
```

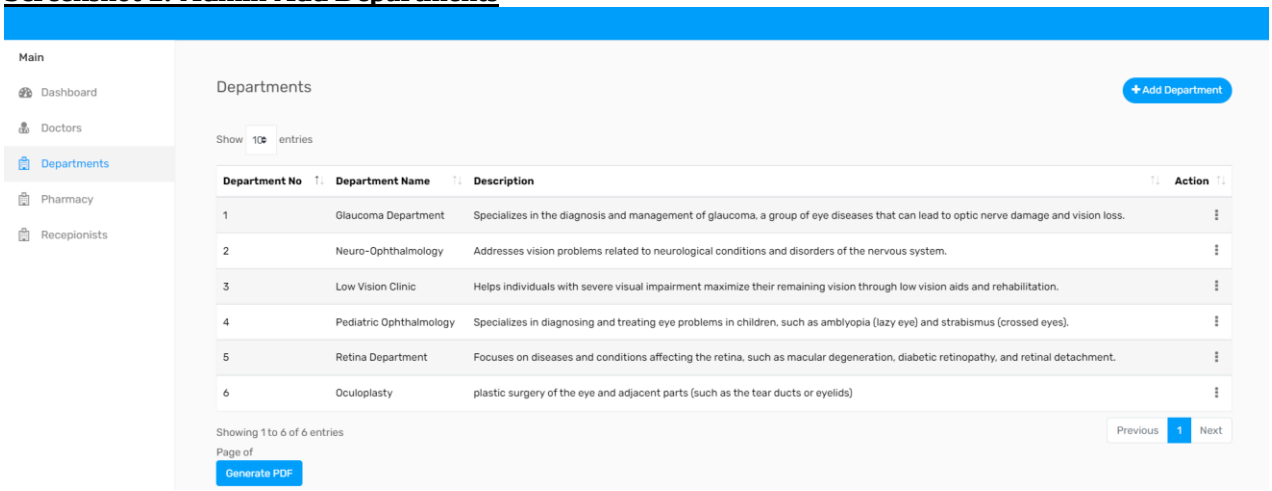
```
<td>{{ i.description }}</td>
<td>
<button class="btn btn-primary">Edit</button>
<a class="dropdown-item" href="{% url 'delete_medicine_category' i.MedCatId %}"><i
class="fa fa-pencil m-r-5"></i> Delete</a>
</td>
</tr>
{% empty %}
<tr>
<td colspan="5">No medicine category found.</td>
</tr>
{% endfor %}
</tbody>
</table>
<button class="btn btn-primary" onclick="generatePDF()">Generate PDF</button>
</div>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap/dist/js/bootstrap.min.js"></script>
<script>
function generatePDF() {
fetch('{% url 'generate_pdf' %}', {
method: 'GET',
})
.then(response => {
if (response.ok) {
return response.blob();
} else {
throw new Error('PDF generation failed');
}
})
.then(blob => {
// Create a link element to download the PDF
const url = window.URL.createObjectURL(blob);
const a = document.createElement('a');
a.href = url;
a.download = 'category_report.pdf';
document.body.appendChild(a);
a.click();
window.URL.revokeObjectURL(url);
})
.catch(error => {
console.error('Error:', error);
});
}
</script>
</body>
</html>
```

9.1 SCREEN SHOTS

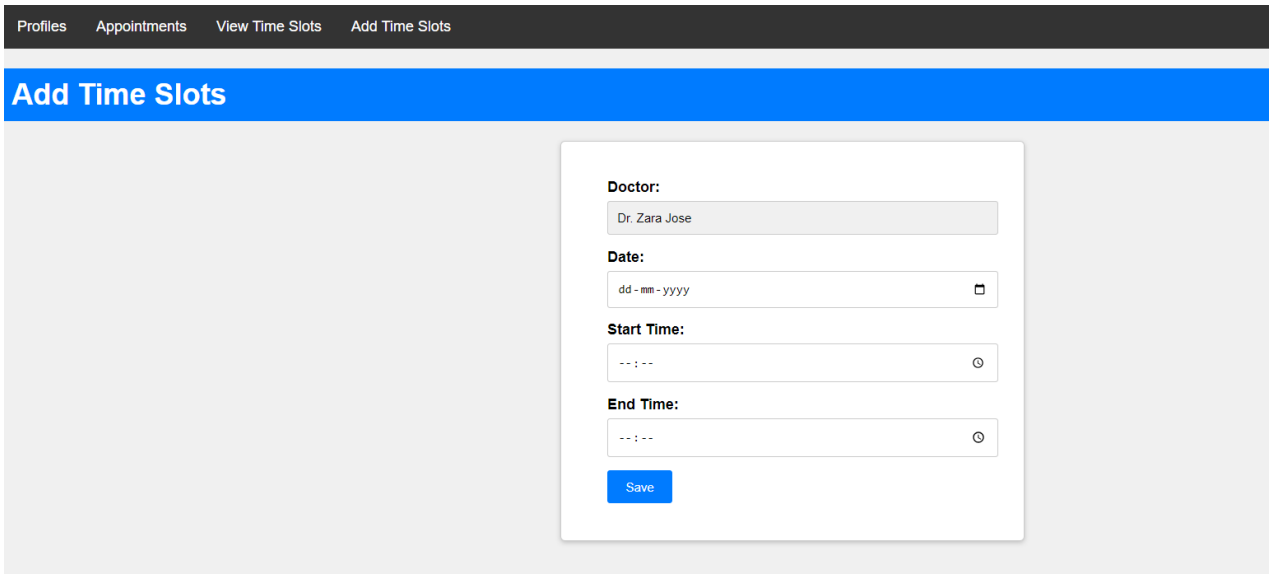
Screenshot 1: Admin add Doctors



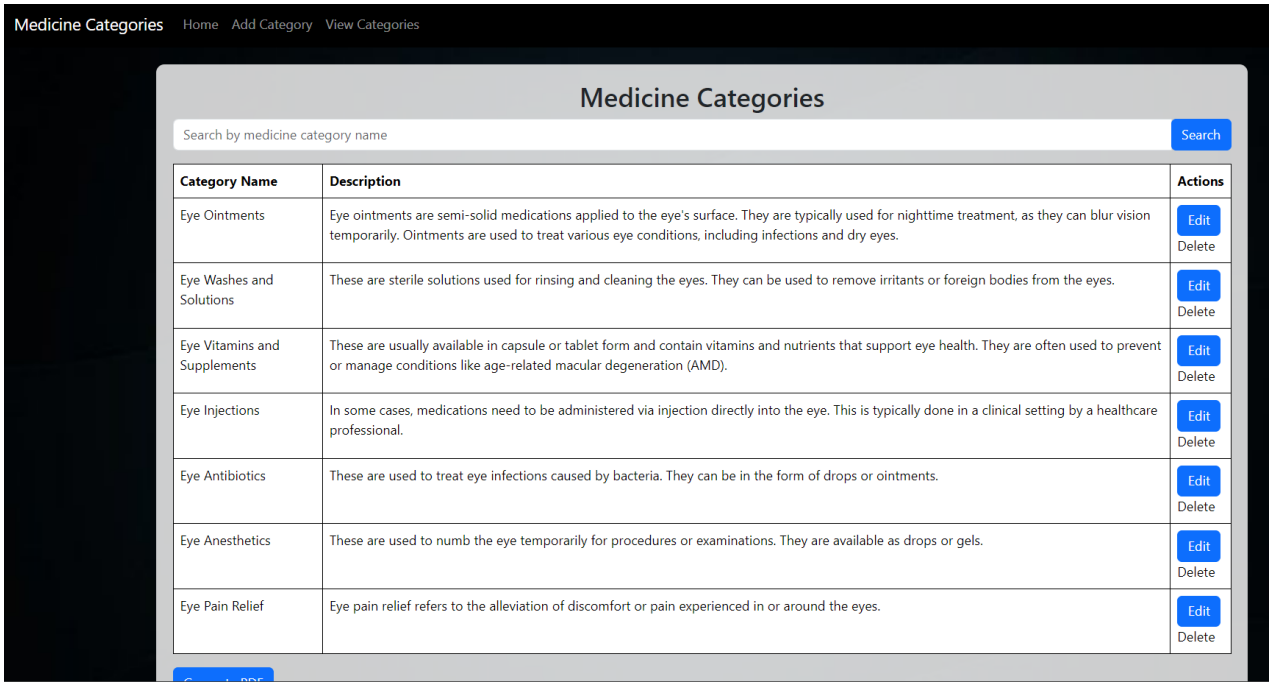
Screenshot 2: Admin Add Departments



Screenshot 3: Doctors Add TimeSlots



Screenshot 4: Pharmacists view Medicine



Screenshot 5: Appointment Form