

INTEGRATED HEALTH CARE PORTAL

A PROJECT REPORT

Submitted by,

C.Syam Sundar Reddy - 20211CST0037

Baddi Ganesh Kumar -20211CST0013

D.V.S.N. Sanjay -20211CST0004

S.Mohan Kumar Reddy-20211CST0026

Under the guidance of,

Mr.Yamanappa

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

Computer Science and Technology[AI&ML]

At



PRESIDENCY UNIVERSITY

BENGALURU

DECEMBER 2024

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

CERTIFICATE

This is to certify that the Project report “**Integrated Health Care Portal**” being submitted by “C.Syam Sundar Reddy - 20211CST0037 ,Baddi Ganesh Kumar – 20211CST0013 , Desu Venkata Naga Sai Sanjay – 20211CST0004 , S.Mohan Kumar Reddy – 20211CST0026” in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Technology[AI&ML] is a bonafide work carried out under my supervision.

Mr.Yamanappa
School of CSE&IS
Presidency University

Dr.Saira Banu Atham
Professor&HoD
School of CSE&IS
Presidency University

Dr. L. SHAKKEERA
Associate Dean
School of CSE
Presidency University

Dr. MYDHILI NAIR
Associate Dean
School of CSE
Presidency University

Dr. SAMEERUDDIN KHAN
Pro-Vc School of Engineering
Dean -School of CSE&IS
Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **INTEGRATED HEALTHCARE PORTAL** in partial fulfilment for the award of Degree of **Bachelor of Technology in Computer Science and Technology**, is a record of our own investigations carried under the guidance of **Mr.Yamanappa, Assistant Professsor, School of Computer Science Engineering & Information Science, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

NAMES	ROLL NO	SIGNATURES
C.Syam Sundar Reddy	20211CST0037	
Baddi Ganesh Kumar	20211CST0013	
D.V.N.S.Sanjay	20211CST0004	
S.Mohan Kumar Reddy	20211CST0026	

ABSTRACT

The "Integrated Healthcare Portal" is an Android-based project designed to centralize and streamline the management of healthcare services. This application serves as a bridge between administrators, healthcare providers, and end-users, providing a comprehensive platform for accessing various medical services. The portal offers separate roles for administrators, users, hospitals, medical shops, labs, and ambulance services, ensuring seamless coordination. Administrators have control over core functionalities, including adding hospitals, medical shops, labs, and ambulance services to the system. Users can register and log in to the platform, enabling them to view hospitals, labs, medical shops, and ambulance services. They can send service requests to hospitals, browse available medical shops and labs, and call ambulances directly through an integrated call option. Hospitals can log in to view and manage user requests, providing timely updates to ensure efficient service delivery. Similarly, medical shops can log in to search for and fulfill requests based on unique request IDs. This portal facilitates better resource allocation, improved communication, and enhanced user convenience, making healthcare services more accessible and effective. The project emphasizes simplicity, efficiency, and scalability, addressing critical healthcare challenges with a modern, integrated solution.

ACKNOWLEDGEMENT

First of all, we indebted to the **LORD KRISHNA** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC, School of Engineering and Dean, School of Computer Science Engineering & Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Deans **Dr. Shakkeera L and Dr. Mydhili Nair**, School of Computer Science Engineering & Information Science, Presidency University, and Dr. “Saira Bhanu” Head of the Department, School of Computer Science Engineering & Information Science, Presidency University, for rendering timely help in completing this project successfully. We are greatly indebted to our guide **Mr.Yamanappa** and Reviewer **Dr.K.Haris Kumar**, School of Computer Science Engineering & Information Science, Presidency University for his inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the PIP2001 Capstone Project Coordinators **Dr. Sampath A K, Dr. Abdul Khadar A and Mr. Md Zia Ur Rahman**, department Project Coordinators **D r . Manjula H M** and Git hub coordinator **Mr. Muthuraj**.

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

C.Syam Sundar Reddy-20211CST0037
Baddi Ganesh Kumar-20211CST0013
Desu Venkata Naga Sai Sanjay-20211CST0004
S.Mohan Kumar Reddy-20211CST002

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iv
	ACKNOWLEDGEMENT	v
	LIST OF FIGURES	9
1.	INTRODUCTION	
	1.1 Motivation	10
	1.2 Problem Statement	10
	1.3 Objective of the project	10
	1.4 Scope	10
	1.5 Project Introduction	11
2.	LITERATURE REVIEW	
	2.1 Related Work	12
3.	SYSTEM ANALYSIS	
	3.1 Existing System	
	3.2 Proposed System	13
		13

4.	REQUIREMENT ANALYSIS	
	4.1 Function and non-functional requirements	14
	4.2 Hardware Requirements	15
	4.3 Software Requirements	15
5.	SYSTEM DESIGN	
	5.1 Introduction of Input design	16
	5.2 UML Diagram(class, use case, sequence, collaborative, deployment, activity, ER diagram and Component diagram)	17
	5.3 Data Flow Diagram	18
6.	IMPLEMENTATION AND RESULTS	
	6.1 Modules	29
7.	SYSTEM STUDY AND TESTING	
	7.1 Feasibility study	30
	7.2 Types of test & Test Cases	31
8.	CONCLUSION	36
9.	FUTURE ENHANCEMENT	37
10.	REFERENCES	38
11.	APPENDIX-A	39
12.	PSEUDO CODE	40
13.	SUSTAINABLE DEVELOPMENT & GOALS	57

LIST OF FIGURES

Sl. No.	Figure Name	Caption	Page No.
1	Fig 1	Class Diagram	20
2	Fig 2	Use-Case Diagram	21
3	Fig 3	Sequence Diagram	22
4	Fig 4	Collaboration Diagram	23
5	Fig 5	Activity Diagram	24
6	Fig 6	Component Diagram	25
7	Fig 7	Deployment Diagram	25
8	Fig 8	ER Diagram	26
9	Fig 9	Data-Flow-Diagram	27
10	Fig 10	Data-Flow-Diagram	28

CHAPTER-1

INTRODUCTION

1.1 Motivation

Healthcare accessibility and coordination remain significant challenges, particularly in emergencies where quick decision-making is crucial. The motivation for this project stems from the need to create a unified platform that connects users with hospitals, medical shops, labs, and ambulance services. By leveraging technology, this portal aims to simplify healthcare access and enhance service delivery, ensuring timely and effective assistance.

1.2 Problem Statement

Healthcare systems often lack integration, leading to delays and inefficiencies in service delivery. Users struggle to find nearby hospitals, labs, medical shops, or ambulances when needed, and communication between stakeholders is fragmented. This project addresses the need for a centralized platform that improves coordination and bridges gaps in healthcare services.

1.3 Objective of the Project

The primary objective is to develop an Android-based healthcare portal that facilitates seamless interaction between users, administrators, and service providers. The system aims to allow users to access and request services like hospital care, lab tests, medicines, and ambulances while enabling providers to efficiently manage and respond to requests.

1.4 Scope

The project covers the development of a user-friendly app where administrators can manage hospitals, labs, medical shops, and ambulance services. Users can register, view available services, send requests, and contact providers. The platform is scalable,

supporting future integration with additional healthcare services and ensuring broader accessibility for urban and rural populations alike.

1.5 Project Introduction

The "Integrated Healthcare Portal" is an Android-based application designed to revolutionize the way healthcare services are accessed and managed. It serves as a comprehensive platform connecting users with various healthcare providers, including hospitals, medical shops, diagnostic labs, and ambulance services. The project addresses the challenges posed by the fragmented nature of traditional healthcare systems, which often result in delays and inefficiencies, particularly during emergencies. This portal provides distinct roles for administrators, users, and service providers. Administrators can oversee and manage the addition of hospitals, labs, medical shops, and ambulance services, ensuring accurate and up-to-date information on the platform. Users can register, log in, and conveniently explore the available services. They can send requests to hospitals, browse and locate medical shops for medicines, schedule lab services, and access ambulances with a direct call feature, simplifying the process of receiving timely care. For service providers, the portal offers functionality tailored to their needs. Hospitals can log in to view and manage user requests, updating their status in real-time. Similarly, medical shops can handle user requests based on unique request IDs, and ambulance services can be easily contacted by users in critical situations. The system's primary goal is to create a unified, user-friendly platform that enhances healthcare accessibility, improves coordination among stakeholders, and optimizes service delivery. By leveraging modern technology, the Integrated Healthcare Portal bridges the gap between healthcare providers and end-users, ensuring a more efficient and reliable healthcare experience. Designed with scalability in mind, the project aims to adapt to future needs, accommodating additional healthcare services and expanding its reach to both urban and rural areas, thereby making quality healthcare accessible to all.

CHAPTER-2

LITERATURE SURVEY

2.1 Related Work

The integration of digital platforms in healthcare has become a critical focus in addressing modern challenges of accessibility, efficiency, and resource optimization. Johnston and Smith (2023) highlight the transformative potential of integrated healthcare platforms, emphasizing their ability to centralize service delivery and improve coordination among stakeholders. Their work explores the core functionalities and design principles needed to modernize healthcare delivery systems effectively. Brown and Lee (2022) examine how digital health solutions can bridge the gap between healthcare providers and patients, particularly in underserved areas. Their research sheds light on the impact of mobile and web-based applications in reducing barriers to accessing medical services, paving the way for equitable healthcare. Nguyen and Chen (2023) delve into the use of artificial intelligence in optimizing healthcare portals, focusing on predictive analytics and resource allocation. Their study underscores the role of AI in enhancing the operational efficiency of integrated systems, particularly in emergency response and inventory management. Kumar and Sharma (2022) explore the design and implementation of mobile applications tailored for healthcare services. They provide insights into user-centered design practices that cater to diverse demographics, ensuring these applications remain accessible and functional across different regions. Wilson and Garcia (2023) discuss the role of mobile apps in streamlining healthcare management. Their work emphasizes the need for scalability and interoperability, allowing seamless integration with existing healthcare systems while ensuring robust security and user privacy. These references collectively underscore the importance of leveraging technology to transform healthcare delivery, each offering unique perspectives on design, implementation, and optimization for digital health solutions.

CHAPTER-3

SYSTEM ANALYSIS

3.1 Existing System:

In the existing healthcare systems are fragmented, with users needing to individually contact hospitals, labs, medical shops, or ambulance services. There is no centralized platform for accessing multiple services, leading to delays and inefficiencies. Communication between users and providers is often manual and uncoordinated.

Disadvantages

- Users must navigate separate systems or contacts for different services.
- Inefficient communication leads to slower response times in critical situations.
- Rural and underserved areas often lack streamlined access to healthcare services.

3.2 Proposed System

The proposed system is an Android-based healthcare portal that integrates hospitals, medical shops, labs, and ambulance services into a unified platform. Administrators can manage services, while users can register, log in, and request or access these services seamlessly. Hospitals can handle user requests, medical shops can fulfill medicine needs, and ambulance services are a call away. The system ensures real-time updates and improves coordination among stakeholders for efficient healthcare delivery.

Advantages

- Provides users with a single platform to find and request multiple healthcare services.
- Reduces delays in accessing medical care, especially during emergencies.
- Streamlines interaction between users and healthcare providers, ensuring prompt responses.
- Supports the addition of new healthcare services and expands to underserved areas..

CHAPTER-4

REQUIREMENT ANALYSIS

4.1 Function and non-functional requirements

Requirement's analysis is very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types: Functional and non-functional requirements. **Functional Requirements:** These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

Examples of functional requirements:

- 1) Authentication of user whenever he/she logs into the system
- 2) System shutdown in case of a cyber-attack

Non-functional requirements: These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.

They basically deal with issues like:

- Portability
- Security
- Maintainability
- Reliability
- Scalability
- Performance
- Reusability

Examples of non-functional requirements:

- 1) Emails should be sent with a latency of no greater than 12 hours from such an activity.
- 2) The processing of each request should be done within 10 seconds
- 3) The site should load in 3 seconds whenever of simultaneous users are > 1000

4.2 Hardware Requirements

- Processor - I3/Intel Processor
- RAM - 8 GB
- Hard Disk - 1TB

4.3 Software Requirements

- Operating System - Windows 10
- JDK - java
- Plugin -Kotlin
- SDK - Android
- IDE -Android studio
- Database` - MY SQL, PHP

CHAPTER-5

SYSTEM DESIGN

5.1 Introduction of Input design

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
3. When the data is entered it will check for its validity. Data can be entered with the help

of screens. Appropriate messages are provided as when needed so that the user will

not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

5.2 UML Diagram

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The

standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: A Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

The Unified Modelling Language (UML) serves as a standardized, general-purpose modelling language within the realm of object-oriented software engineering, overseen and created by the Object Management Group (OMG). Its primary objective is to establish a universal language for modelling object-oriented computer software, aiming to provide a common ground for software developers to communicate and collaborate effectively. UML consists of two main components: a Meta-model, which defines the structure and semantics of UML itself, and a notation, which encompasses the graphical symbols and diagrams used to represent various aspects of software systems. While currently focused on these components, UML may incorporate additional methods or processes in the future. As a standard language, UML facilitates the specification, visualization, construction, and documentation of software artifacts, along with applications in business modelling and other non-software domains. It encapsulates a collection of best engineering practices proven effective in modelling large and intricate systems. In the software development process, UML plays a pivotal role by enabling developers to express the design of software projects using graphical notations. Its

adoption promotes clarity, consistency, and efficiency in communication, aiding in the development of robust and scalable object-oriented software systems. Thus, UML stands as a cornerstone in the development of object-oriented software and the broader software engineering process.

GOALS:

The Primary goals in the design of the UML are as follows:

- Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models.
- Provide extendibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development process.
- Provide a formal basis for understanding the modelling language.
- Encourage the growth of OO tools market.
- Support higher level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices.

CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

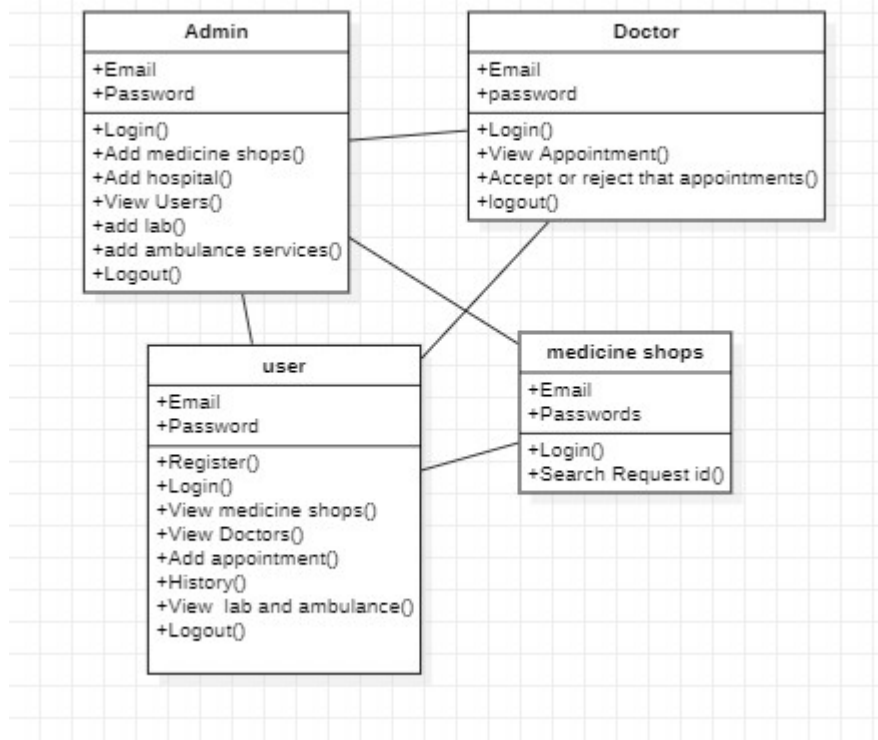


FIG-1

USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

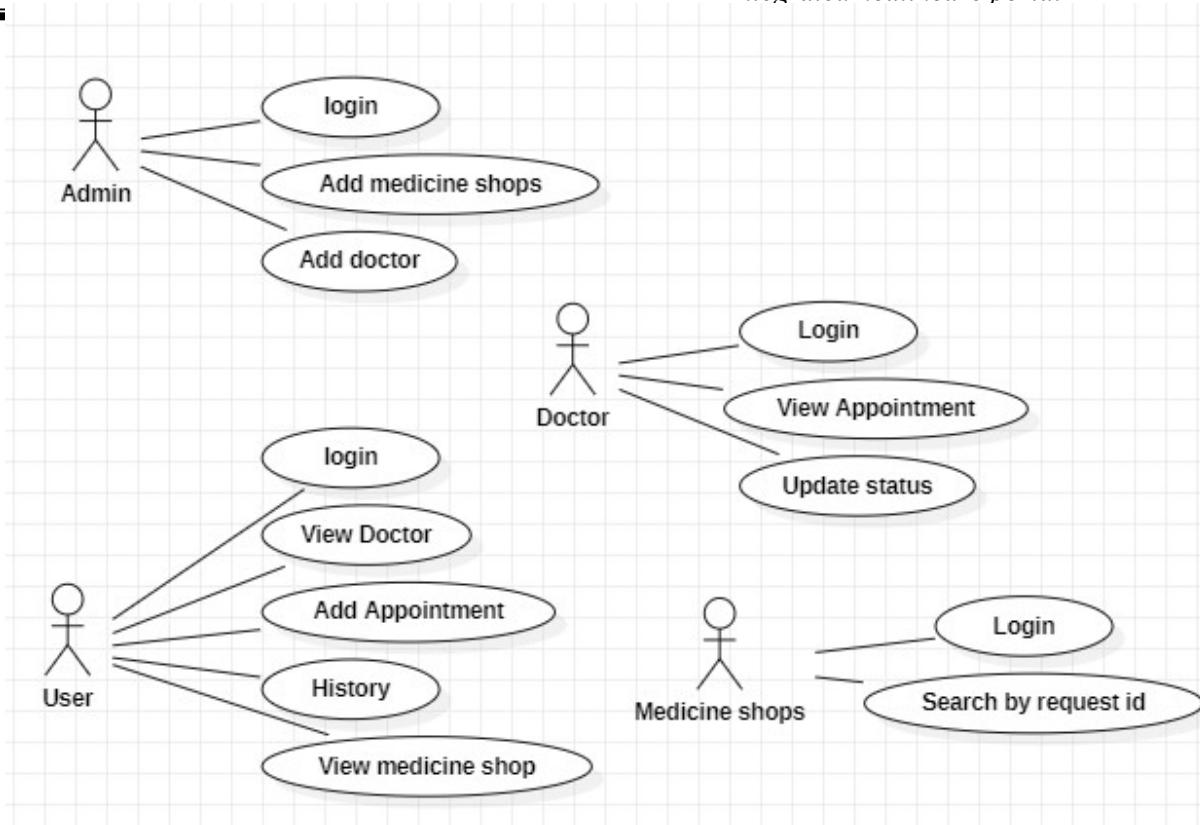


FIG-2

SEQUENCE DIAGRAM:

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

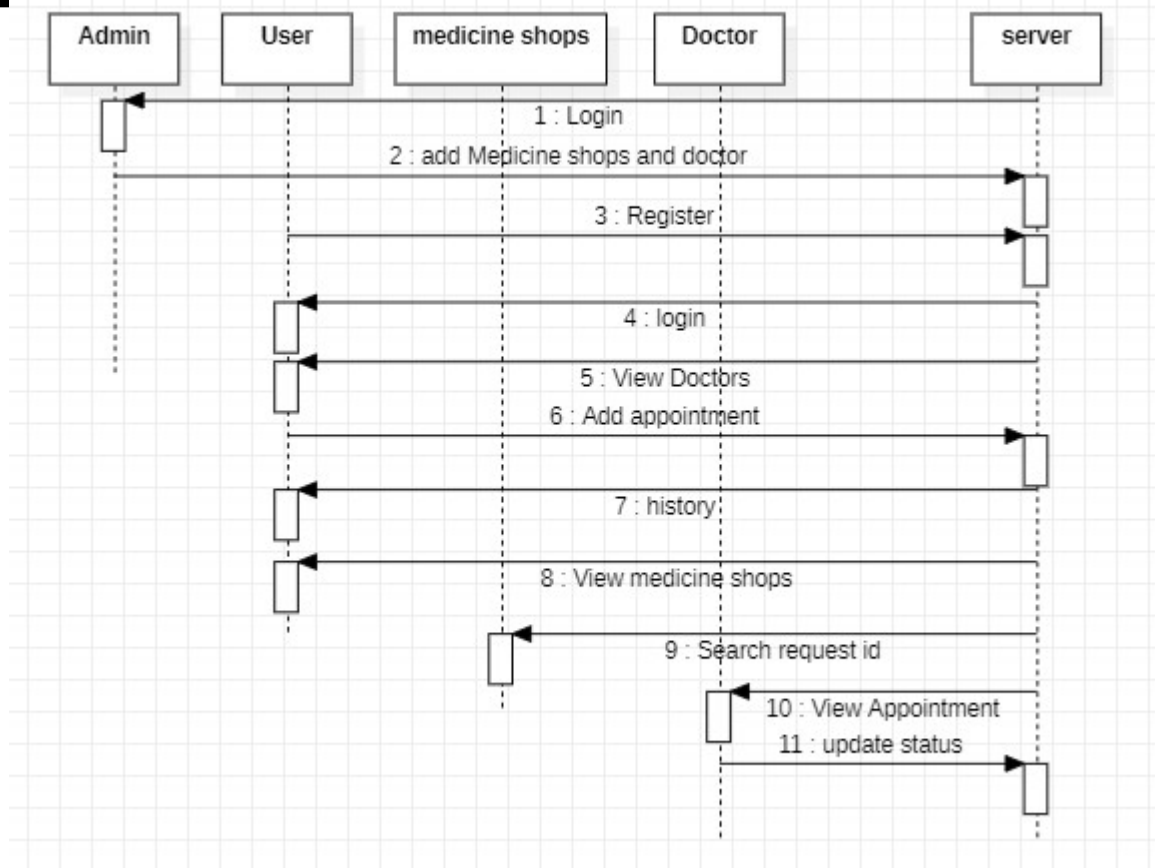


FIG-3

COLLABORATION DIAGRAM:

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.

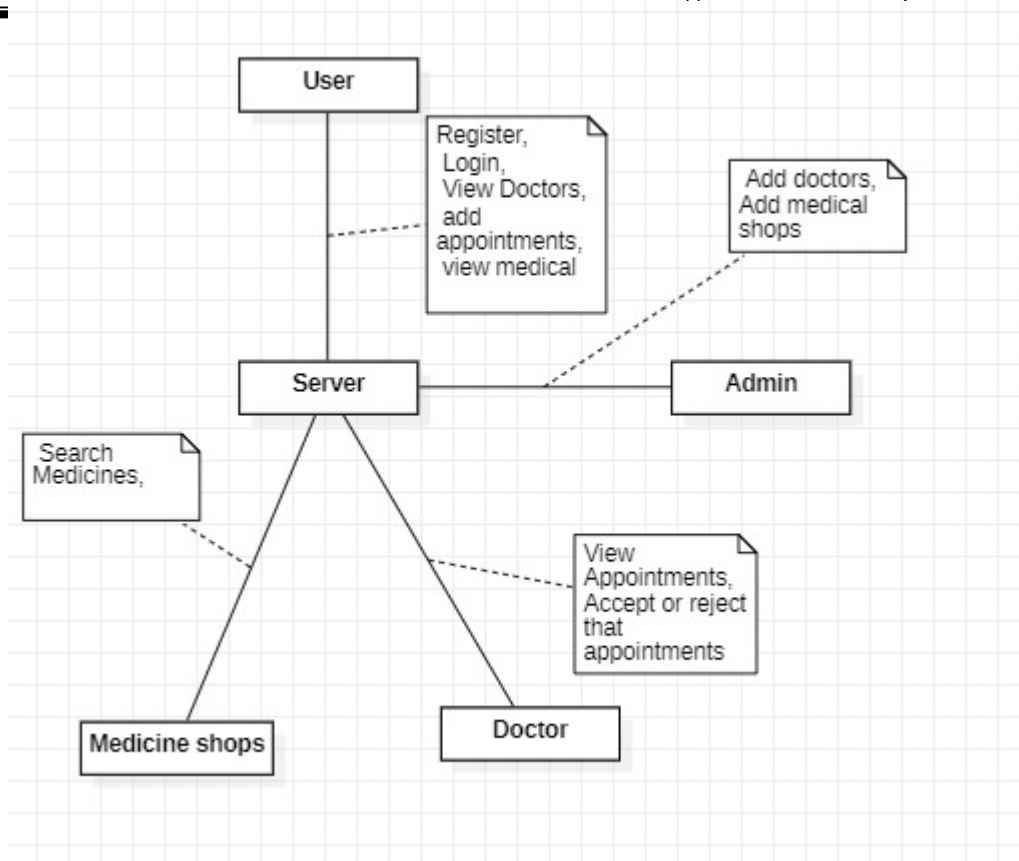


FIG-4

ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

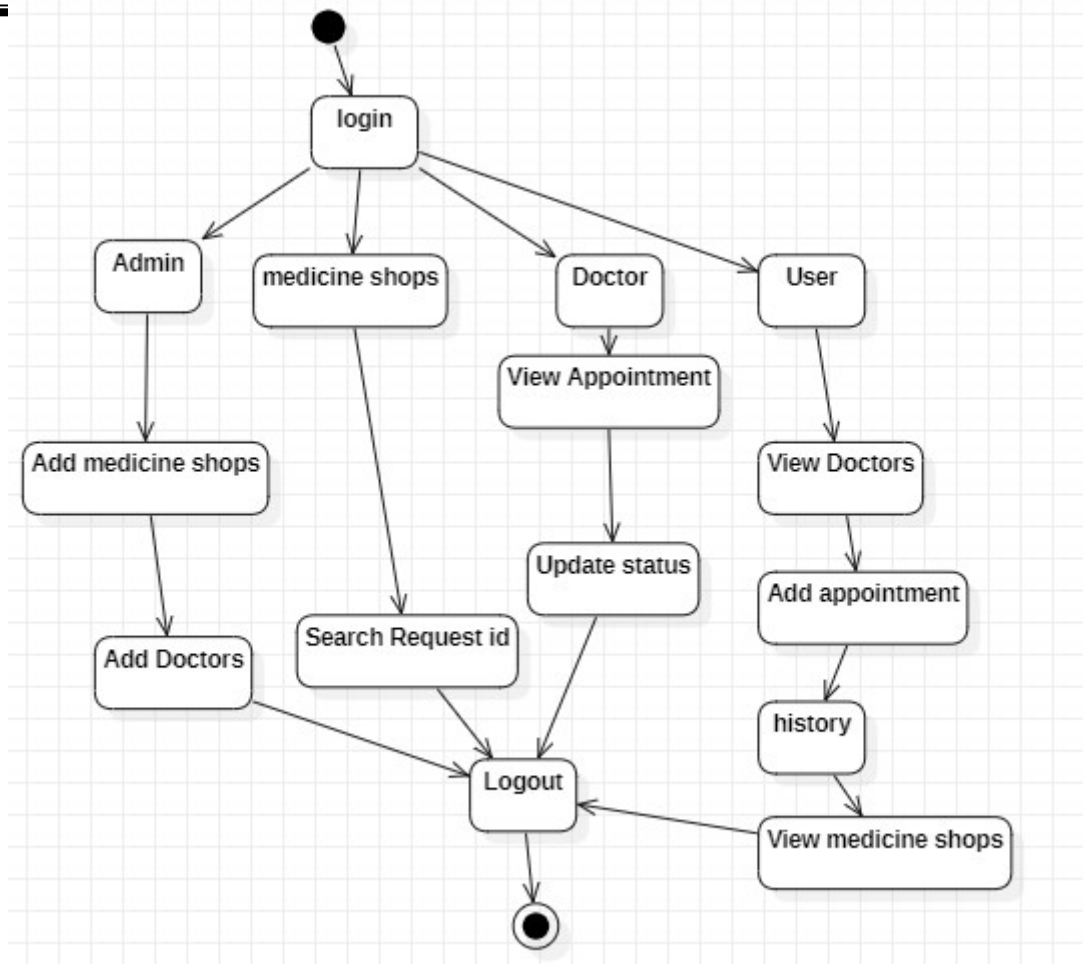


FIG-5

COMPONENT DIAGRAM:

A component diagram in software engineering illustrates the components of a system and their relationships. Components represent modular units of functionality, such as classes, modules, or libraries, and are depicted as rectangles with the component's name inside. Relationships between components are shown with lines connecting them, indicating dependencies, associations, or interfaces. Component diagrams help visualize the architecture of a system, including how components interact and communicate with each other. They are useful for understanding the structure of a software system and for

communicating design decisions to stakeholders.

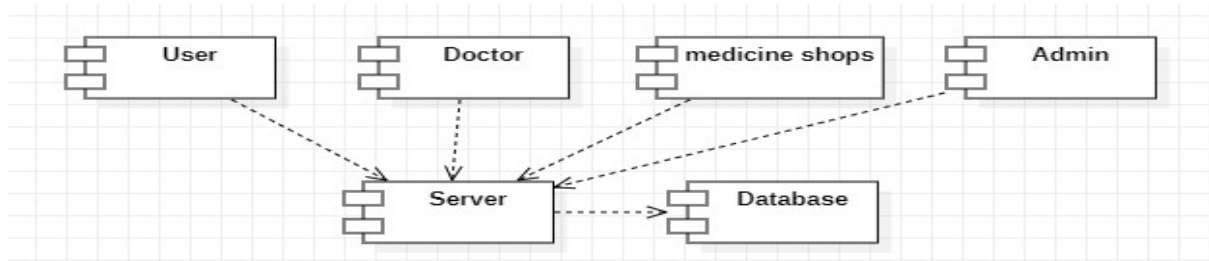


FIG-6

DEPLOYMENT DIAGRAM:

A deployment diagram in software engineering visualizes the physical deployment of software components onto hardware nodes in a distributed system. Nodes represent hardware devices, such as servers, computers, or mobile devices, depicted as rectangles with the node's name inside. Components, represented by rectangles with the component's name inside, are deployed onto nodes, showing how software elements are distributed across the hardware infrastructure. Deployment diagrams illustrate the configuration and deployment topology of a system, including the relationships between software components and the hardware resources they utilize. They aid in understanding system deployment and resource allocation in distributed environments.

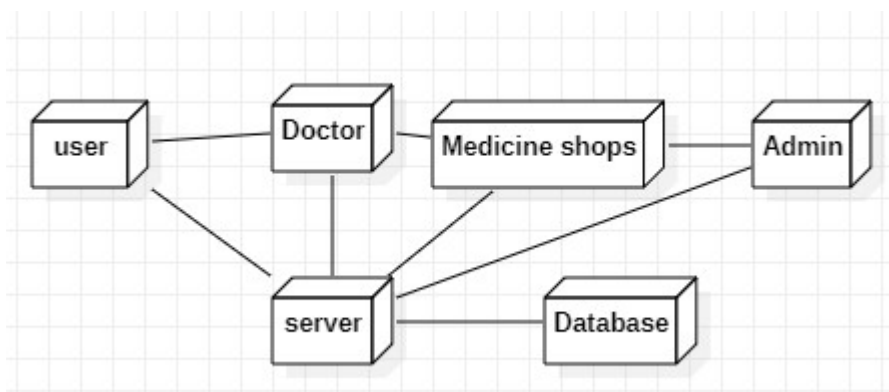


FIG-7

ER Diagram:

An Entity-Relationship (ER) diagram in database design illustrates the relationships between entities within a database schema. Entities represent real-world objects or

concepts, such as customers, orders, or products, depicted as rectangles with the entity's name inside. Relationships between entities are shown with lines connecting them, indicating associations or dependencies. Cardinality and participation constraints may also be included to specify the nature of the relationships. ER diagrams help visualize the structure of a database schema, including the entities, attributes, and relationships between them. They serve as a blueprint for designing and implementing relational databases effectively.

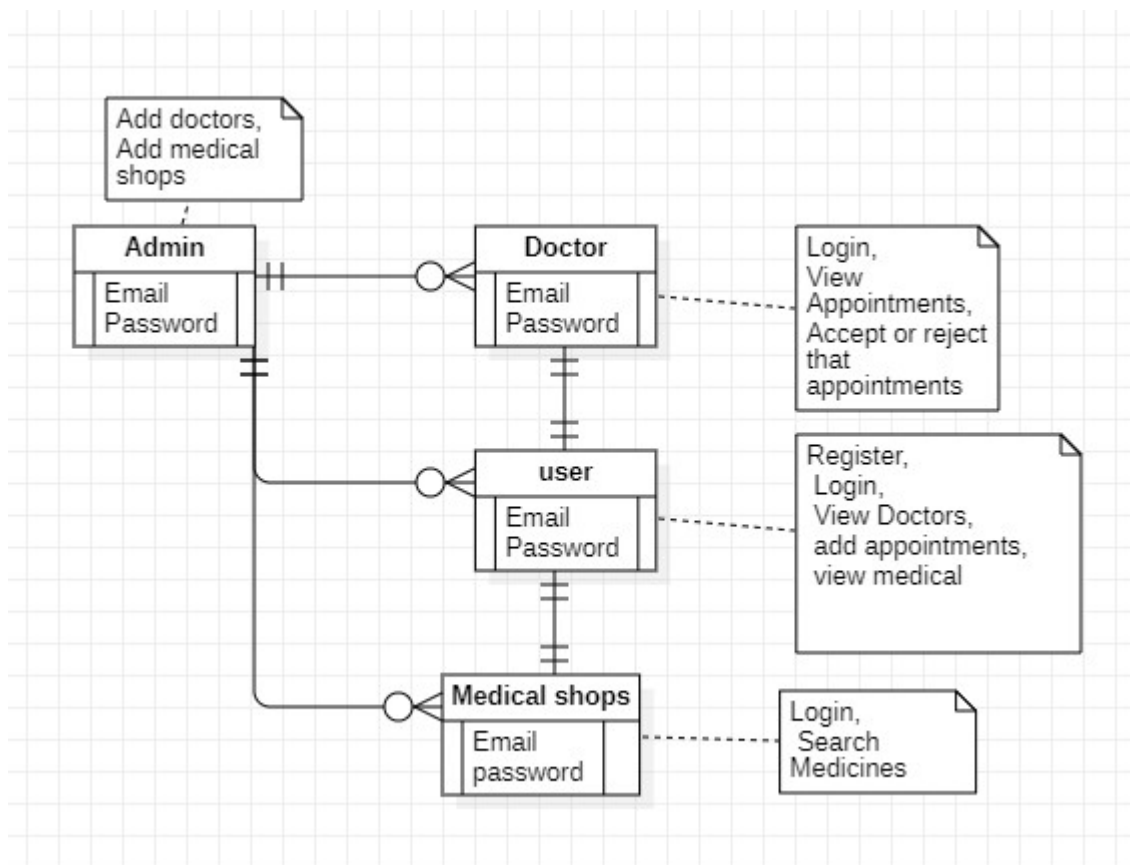
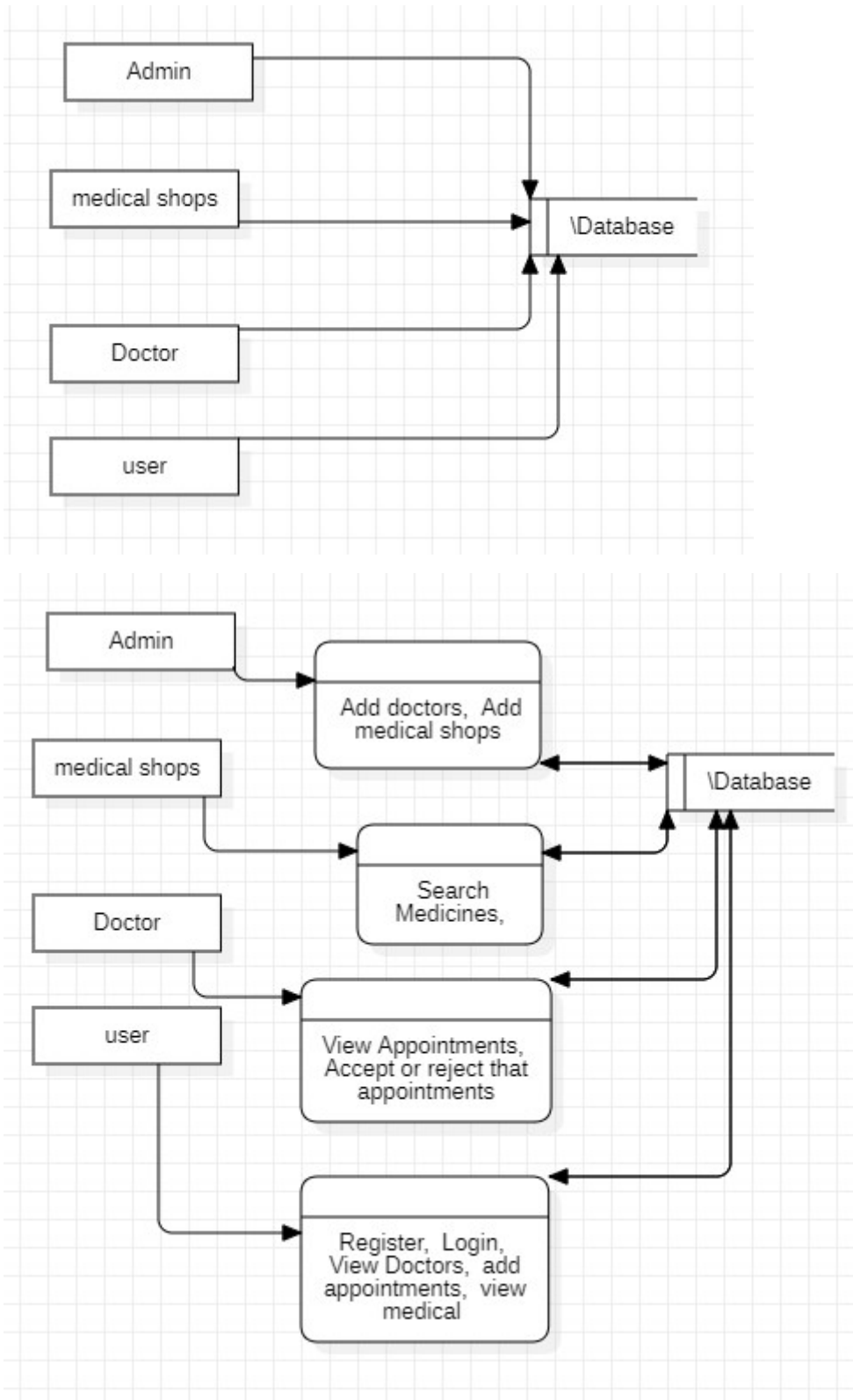
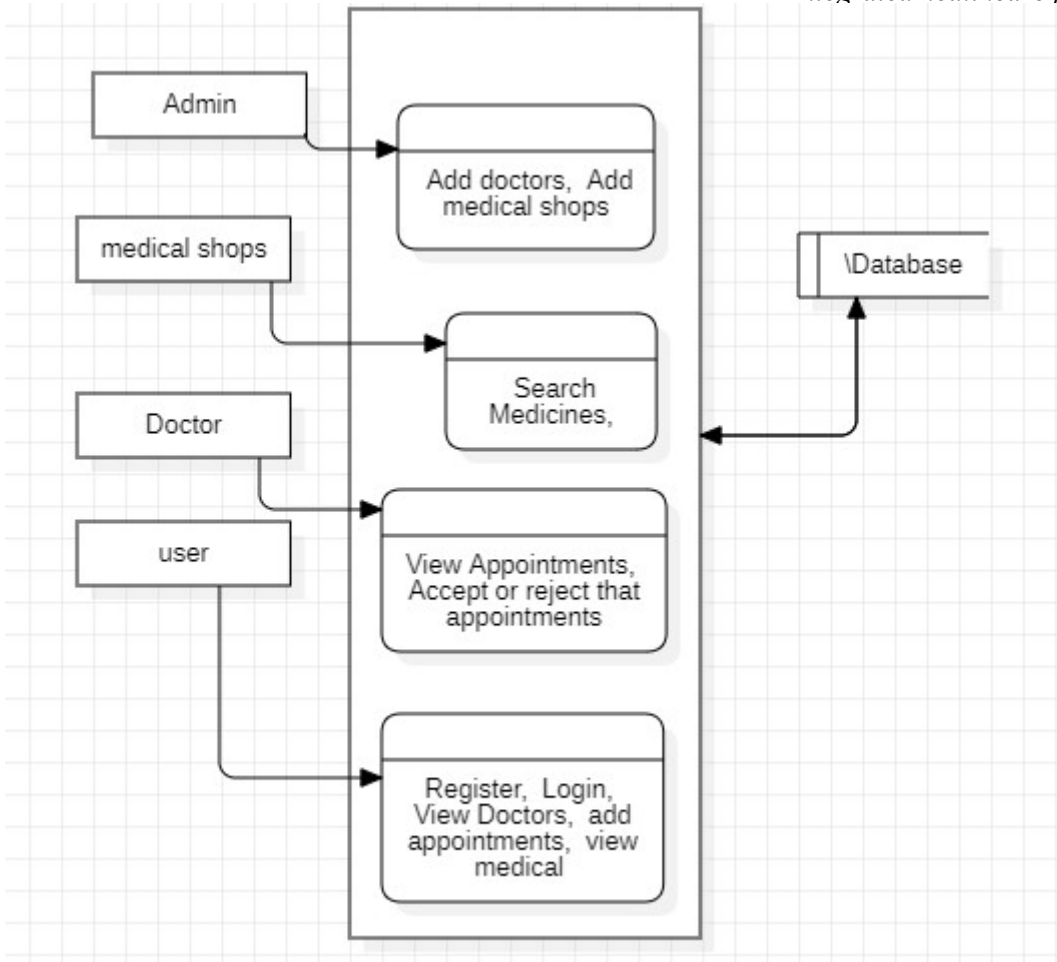


FIG-8

5.3 Data Flow Diagram:

FIG-9



**FIG-10**

CHAPTER-6

IMPLEMENTATION AND RESULTS

6.1 Modules

Admin Module: The administrator can log in to the portal and manage the database by adding hospitals, medical shops, labs, and ambulance services. This ensures that all service providers are validated and up-to-date.

User Module: Users can register, log in, and access services such as viewing hospitals, medical shops, labs, and ambulance services. Users can send requests to hospitals, search for medical shop inventories by request ID, and directly contact ambulance services with a call option.

Hospital Module: Hospitals can log in, view user requests, and update the status of those requests to maintain transparency and responsiveness.

Medical Shops Module: These providers can log in to search for specific requests by request ID and title, ensuring they can promptly address user needs.

CHAPTER-7

SYSTEM STUDY AND TESTING

7.1 Feasibility study

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- **ECONOMICAL FEASIBILITY**
- **TECHNICAL FEASIBILITY**
- **SOCIAL FEASIBILITY**

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His

level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

7.2 Types of test & Test Cases

UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they

actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centred on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked. Organization and preparation

of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

SYSTEM TEST

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

WHITE BOX TESTING

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

UNIT TESTING:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed

- All links should take the user to the correct page.

INTEGRATION TESTINGSS

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g., components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test case id	Test Scenario	Test Steps	Prerequisites	Test Data	Expected result	Actual result	Test status
#CVD001	To authenticate a successful signup with user data	<ul style="list-style-type: none"> • User navigate the signup page • Enter the valid user data • Click on signup button 	User data	Username Password Mobile Email location	When the user submits the user data, data should be store in database successfully	As Expected,	Pass
#CVD002	To authenticate a successful login with user data	<ul style="list-style-type: none"> • User navigate the login page • Enter the valid username, password • Click on login button 	Username, password	Username, password	When the user submits the user data, data should be authenticate successfully	As Expected,	Pass

TESTING CASES

CHAPTER - 8

CONCLUSION

In conclusion, the "Integrated Healthcare Portal" represents a significant advancement in the centralization and management of healthcare services, addressing the critical need for streamlined communication and coordination among healthcare stakeholders. By integrating administrators, users, hospitals, medical shops, labs, and ambulance services into a unified platform, the application enhances resource allocation, improves service accessibility, and fosters user convenience. Its user-friendly interface and robust functionality ensure efficiency and reliability, empowering users to seamlessly connect with essential healthcare services.

CHAPTER - 9

FUTURE ENHANCEMENT

Future enhancements to the portal could include integrating AI-driven tools for predictive analytics, enabling proactive healthcare management and better resource forecasting. Expanding telemedicine capabilities would allow users to consult with healthcare providers remotely, increasing accessibility for underserved areas. Incorporating secure payment gateways and electronic health records (EHRs) could further improve service convenience and data management. Additionally, multilingual support and voice-activated navigation could enhance usability for diverse user groups. Advanced notification systems, real-time tracking for ambulances, and partnerships with government health programs could further scale the platform's impact. These enhancements would strengthen the portal's role as a transformative healthcare solution, ensuring adaptability to evolving healthcare challenges and user needs.

CHAPTER-10

REFERENCES

1. M. Johnston and L. Smith, "Integrated Healthcare Platforms: Modernizing Service Delivery," *Journal of Healthcare Informatics*, , 2023.
 2. T. Brown and A. Lee, "Digital Health Solutions: Transforming Access to Medical Services," *International Journal of Digital Health*, 2022.
 3. P. Nguyen and Y. Chen, "AI-Driven Healthcare Portals for Resource Optimization," in *Proceedings of the Global Health Tech Conference*, 2023,
 4. R. Kumar and P. Sharma, "Mobile Applications for Integrated Healthcare Services," *Journal of Mobile Health Innovation*, 2022.
 5. H. Wilson and M. Garcia, "Streamlining Healthcare Management Through Mobile Apps," *Advances in Healthcare Technology*, 2023.
 6. S. Patel and R. Johnson, "Scalable Architectures for Healthcare Applications," *IEEE Transactions on Health Informatics*, 2023.
 7. Z. Ahmed and T. Walker, "User-Centric Design in Healthcare Portals: A Case Study," *International Journal of User Experience in Healthcare*, 2023.
 8. J. Kim and H. Park, "The Role of Telemedicine in Modern Healthcare Portals," *Telehealth and Medicine Today*, 2022.
 9. M. Singh and K. Gupta, "Integrated Healthcare Platforms for Urban and Rural Communities," *Journal of Community Health Systems*, 2023.
- C. Taylor and D. Evans, "Enhancing Communication in Healthcare with Digital Tools," *Journal of Health Communication Research*, 2023.

APPENDIX-A

ADMIN PAGE :

11:04 7.00 KB/s 5G 84

LABORATORY 

NAME *mohan lab*
EMAIL *mohan@gmail.com*
NUMBER *7569155706*
CITY *Bengaluru*

AVAILABLE

MORNING TO EVENING

ALL TESTS ARE PROVIDED

Delete

NAME *lab*
EMAIL *lab@gmail.com*
NUMBER *4555555555*
CITY *Tirupati*

AVAILABLE

MORNING 8 TO 7PM

ALL TESTS WE WILL PERFORM

Delete  success

11:05 4.00 KB/s 5G 33

HOSPITAL 

NAME *Sriram bieling*
EMAIL *sriram@gmail.com*
NUMBER *6361738871*
CITY *Tamilnadu*

AVAILABLE

MORNING TO EVENING

MULTI-SPECIALITY

Delete

NAME *raksha hospital*
EMAIL *rakshahospital@gmail.com*
NUMBER *9638525896*
CITY *Rajankunte*

AVAILABLE

24 HOURS

WE ARE COMMITTED TO PROVIDING
CONVENIENT, EXPERTS AND LOCAL
HEALTH CARE AT AFFORDABLE PRICES

 success
Delete

11:04 17.0 KB/s 4G 53%

AMBULANCE



NAME rah
EMAIL raj@gmail.com
NUMBER 5775787878
CITY Tirupati

AVAILABLE

24 HOURS

Delete



success

11:05 1.00 KB/s 5G 53%

USER

NAME sriram
EMAIL Choudhary@gmail.com
NUMBER 6361738871
CITY Tamilnadu

NAME Ganesh Kumar
EMAIL baddiganesh143@gmail.com
NUMBER 9346743367
CITY Visakhapatnam

NAME Tanveer Shaik
EMAIL tanveer442004@gmail.com
NUMBER 9193984502
CITY Hindupur

NAME uday
EMAIL uday@gmail.com
NUMBER 9676852398
CITY Bengaluru

NAME jashu
EMAIL jas@gmail.com
NUMBER 9346743367

success[user/id=35, name=sriram, num=6361738871, email=Choudhary]

NAME abhi
EMAIL abhi@gmail.com
NUMBER 6300238596

11:06 2.02 KB/s 5G 53%

MEDICAL SHOPS



NAME medical
EMAIL medical@gmail.com
NUMBER 5855555555
CITY Tirupati

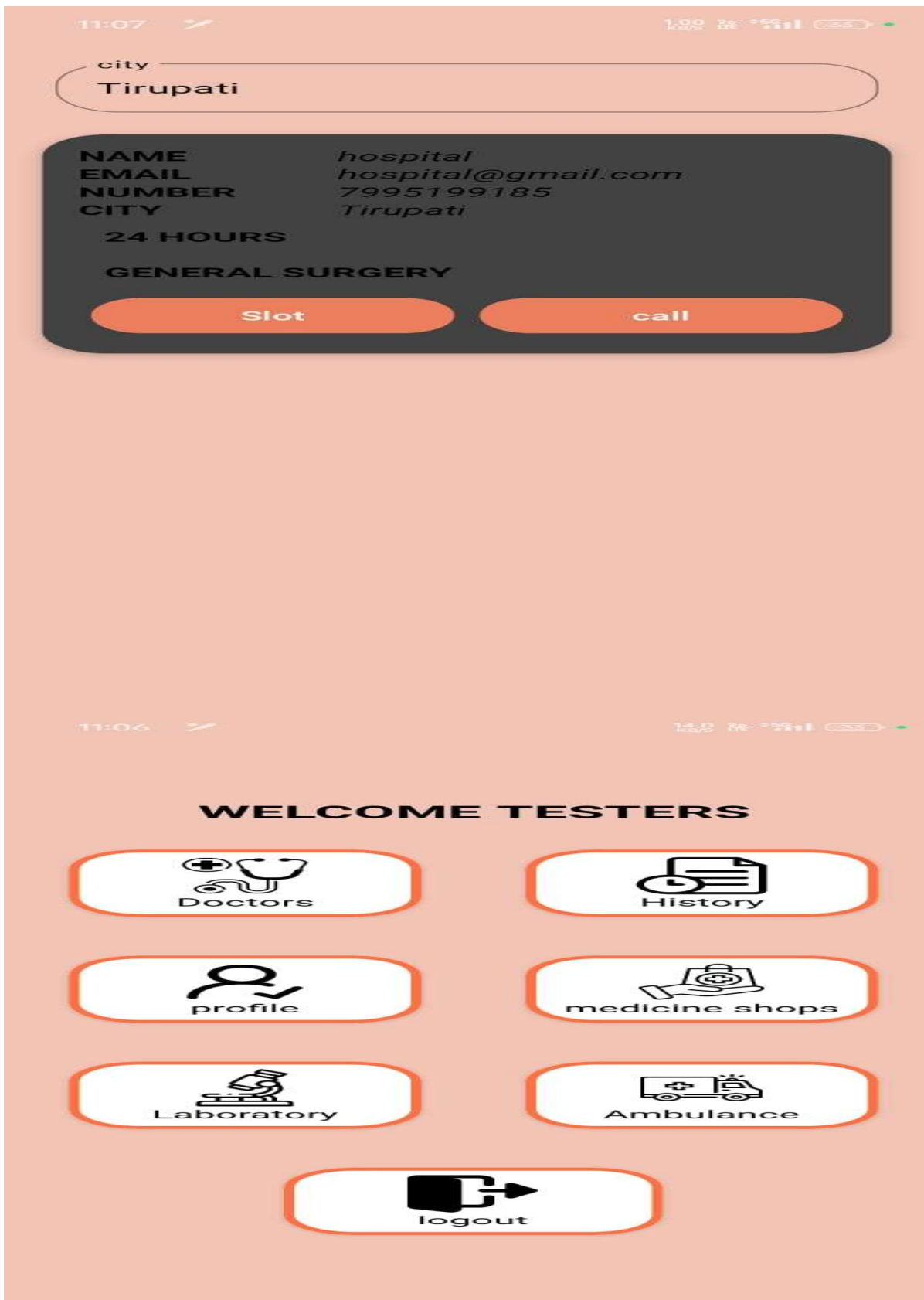
AVAILABLE

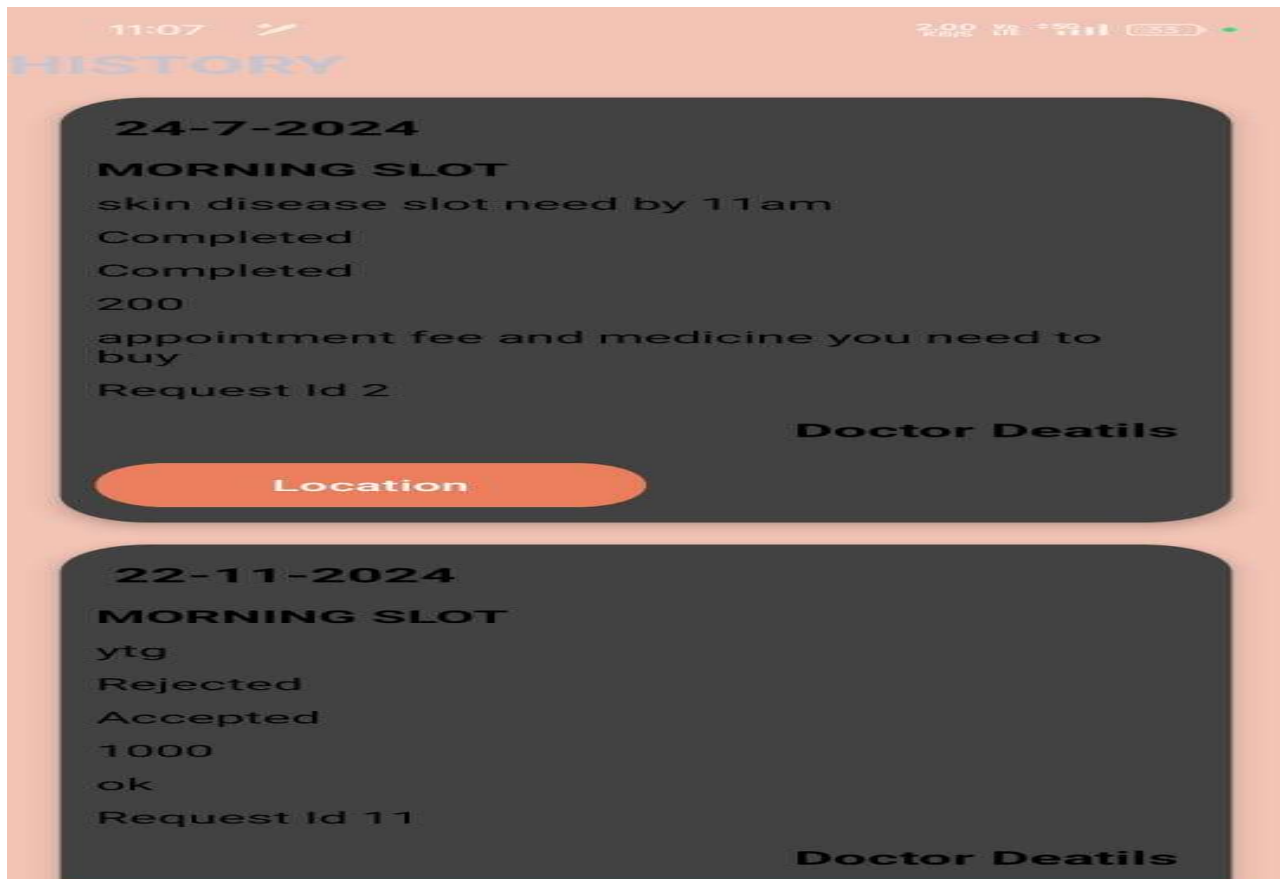
Delete



success

USER PAGE:





call

call



11:08

2:00 5G 32

city
Tirupati

NAME rah
EMAIL raj@gmail.com
NUMBER 5775787878
CITY Tirupati

call



Ambulance

HOSPITAL PAGE:

11:10

m

21.2 W 5G

100%

WELCOME MEDICAL

Request id

2

24-7-2024

MORNING SLOT

skin disease slot need by 11am

Completed

Completed

200

appointment fee and medicine you need to buy

Request Id 2

Doctor Deatils

Location

11:10

m

21.2 W 5G

100%

WELCOME HOSPITAL

24-7-2024

MORNING SLOT

skin disease slot need by 11am

Completed

Completed

200

appointment fee and medicine you need to buy

User Deatils

Upload

nice support

22-11-2024

[requestda(requestId=2, statusPoint=Completed, usstatus=Com...

Rejected

Accepted

1000

PSEUDO CODE

```
package com.example.integratedhealthcareportal

import android.annotation.SuppressLint
import android.content.Context
import android.content.Intent
import android.os.Bundle
import android.telephony.gsm.SmsManager
import android.telephony.gsm.SmsManager.getDefault
import android.text.TextUtils
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.Toast
import android.os.Handler
import android.text.InputFilter
import android.widget.ArrayAdapter
import android.widget.PopupMenu
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.app.AppCompatActivity
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import com.example.grihagully.model.RetrofitClient
import
com.example.integratedhealthcareportal.databinding.ActivityHospitalDashbroadBinding
import com.example.integratedhealthcareportal.databinding.CarddoccostBinding
import com.example.integratedhealthcareportal.databinding.CardhoshostoryBinding
import com.example.integratedhealthcareportal.databinding.CardprofileBinding
import com.example.medease.model.logout
import com.example.skinsmart.model.requestda
import com.example.skinsmart.model.requestresponse
import com.google.android.material.bottomsheet.BottomSheetDialog
import com.ymts0579.model.model.DefaultResponse
import kotlinx.coroutines.CoroutineScope
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.launch
import retrofit2.Call
import retrofit2.Callback
import retrofit2.Response

class DoctorDashboard : AppCompatActivity() {
```

```
private val b by lazy {
    ActivityHospitalDashbroadBinding.inflate(layoutInflater)
}
private val bind by lazy {
    CardprofileBinding.inflate(layoutInflater)
}
var id=0
var name=""
var num=""
var email=""
var address=""
var city=""
var pass=""
var type=""
var status=""
var descrip=""
var timing=""
private lateinit var progressDialog: AlertDialog
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)

    setContentView(b.root)

    getSharedPreferences("user", AppCompatActivity.MODE_PRIVATE).apply {
        id=getInt("id",0)
        name=getString("name","").toString()
        num=getString("num","").toString()
        email=getString("email","").toString()
        address=getString("address","").toString()
        city=getString("city","").toString()
        pass=getString("pass","").toString()
        type=getString("type","").toString()
        status=getString("status","").toString()
        descrip=getString("descrip","").toString()
        timing=getString("timing","").toString()
    }
    readvieworders(email)

    bind.etname.setText(name)
    bind.etnum.setText(num)
```

```
bind.icemail.setText(email)
```

```
bind.etaddress.setText(address)
    bind.etcity.setText(city)
    bind.etpassword.setText(pass)
    bind.ettiming.setText(timing)
    bind.etdescri.setText(descrip)
```

```
    bind.etcity.setFilters(arrayOf<InputFilter>(InputFilter { source, start, end, dest,
dstart, dend ->
    if (source.length > 0 && dstart == 0) {
        val v = CharArray(source.length)
        TextUtils.getChars(source, 0, source.length, v, 0)
        v[0] = v[0].uppercaseChar()
        return@InputFilter String(v)
    }
    null
}
))
```

```
val k=arrayOf("choose your choice","Available","Not Available")
```

```
ArrayAdapter(this@DoctorDashboard,
    android.R.layout.simple_list_item_checked, k).apply {
    bind.spinstatus.adapter=this
}
k.forEachIndexed { index, s ->
    if(s==status){
        bind.spinstatus.setSelection(index,true)
    }
}
```

```
b.tvname.text="Welcome $name"
b.imageView4.setOnClickListener {
    val popupMenu: PopupMenu = PopupMenu(this,b.imageView4)
    popupMenu.menuInflater.inflate(R.menu.menu,popupMenu.menu)
```

```
popupMenu.setOnMenuItemClickListener(PopupMenu.OnMenuItemClickListener {
item ->
    when(item.itemId) {
        R.id.action_profile ->{
            popupMenu.dismiss()
            BottomSheetDialog(this).apply {
```

```
setContentView(bind.root)
```

```
/// profile
```

```
bind.btnupdateprofile.setOnClickListener {  
    val name1=bind.ename.text.toString().trim()  
    val num1=bind.etnum.text.toString().trim()  
    val email1=bind.etable.text.toString().trim()  
    val address1=bind.etaddress.text.toString().trim()  
    val city1=bind.etcity.text.toString().trim()  
    val password1=bind.etpassword.text.toString().trim()  
    val timing1=bind.ettiming.text.toString().trim()  
    val descrip1=bind.etchedescr.text.toString().trim()  
    val status1=bind.spinstatus.selectedItem.toString()  
  
    if(name1.isEmpty()){  
        bind.ename.error="Enter your Name"  
    }else if(num1.isEmpty()){  
        bind.etnum.error="Enter your Number"  
    }else if(email1.isEmpty()){  
        bind.etable.error="Enter your Email"  
    }else if(address1.isEmpty()){  
        bind.etaddress.error="Enter your Address"  
    }else if(city1.isEmpty()){  
        bind.etcity.error="Enter your city"  
    }else if(password1.isEmpty()){  
        bind.etpassword.error="Enter your Password"  
    }else if(timing1.isEmpty()){  
        bind.ettiming.error="Enter your timings"  
    }else if(descrip1.isEmpty()){  
        bind.etnum.error="Enter your Description"  
    } else if(num1.count() != 10){  
        bind.etnum.error="Enter your Number properly"  
    }else if(status1=="choose your choice"){  
        Toast.makeText(this@DoctorDashboard, "choose your choice",  
Toast.LENGTH_SHORT).show()  
    }else{  
        CoroutineScope(Dispatchers.IO).launch {  
  
RetrofitClient.instance.updateprofile(id,name1,num1,email1,address1,city1,password1,t  
ype,status1,descrip1,timing1,"update")  
            .enqueue(object: Callback<DefaultResponse> {  
                override fun onFailure(call: Call<DefaultResponse>, t:
```



```
R.id.action_Logout -> {logout()
    popupMenu.dismiss()}
```

```
    }

    }
    // data view adaptr

    inner class hospitalorderAdapter(var context: Context, var listdata:
    ArrayList<requestda>):
        RecyclerView.Adapter<hospitalorderAdapter.DataViewHolder>(){

            inner class DataViewHolder(val view: CardhoshostoryBinding) :
            RecyclerView.ViewHolder(view.root)


            override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
            DataViewHolder {
                return DataViewHolder(
                    CardhoshostoryBinding.inflate(
                        LayoutInflater.from(context),parent,
                        false))
            }

            override fun onBindViewHolder(holder: DataViewHolder,
            @SuppressWarnings("RecyclerView") position:Int) {
                with(holder.view){

                    listdata[position].apply {
                        tvdate.text=date
                        tvslot.text=slot
                        tvusedescr.text=udescr
                        tvstatus.text=usstatus
                        tvdstatus.text=statusPoint
                        tvcost.text=cost
                        tvddescription.text=ddescrip
                        tvdoctorinfo.text="Name $name\n"+"Number $num"
                        tvdoctorinfo.visibility= View.GONE
                        linearfeed.visibility=View.GONE
                        tvfeedback.text=feedback

                        var floaft=0.0f
                        rating!!.forEach {
                            if(it!=' '&&it.isDigit()||it=='.'){
```

```
floatf=it.toFloat()
    }
}
ratingbar.isIndeterminate=true
ratingbar.rating=floatf

if(statusPoint=="Pending"&& status=="Pending"){

    tvcost.visibility= View.GONE
    tvustatus.visibility= View.GONE
    tvcost.visibility= View.GONE
    tvddescription.visibility= View.GONE
    btncompleted.visibility=View.GONE
    btnAccept.visibility=View.VISIBLE
} else if(statusPoint=="Rejected"||status=="Rejected"){
    tvcost.visibility= View.GONE
    tvustatus.visibility= View.GONE
    tvcost.visibility= View.GONE
    tvddescription.visibility= View.GONE
    btncompleted.visibility=View.GONE
    btnAccept.visibility=View.GONE
    btnupload.visibility=View.GONE
}

if(statusPoint=="Accepted"){
    btnAccept.visibility=View.GONE
}

if(statusPoint=="Completed"){
    btncompleted.visibility=View.GONE
    btnAccept.visibility=View.GONE
}

if(feedback==""){
    linearfeed.visibility=View.GONE
} else {

    linearfeed.visibility=View.VISIBLE
}
```

```
tvdoctordeatils.setOnClickListener {
    tvdoctorinfo.visibility= View.VISIBLE
    Handler().postDelayed({

tvdoctorinfo.visibility= View.GONE

    },3500)

    }

    if(path==""){
        btnupload.visibility=View.VISIBLE
    }

    btnupload.setOnClickListener {
        context.startActivity(Intent(context,DoctorUpload::class.java).apply {
            putExtra("rid",requestId)

        })
    }

    btnAccept.setOnClickListener {

        val alertDialog= AlertDialog.Builder(context)
        alertDialog.setTitle("Accept or Reject ")
        alertDialog.setIcon(R.drawable.logo)
        alertDialog.setCancelable(false)
        alertDialog.setMessage("Do you Want to Accept or Reject the Slot?")
        alertDialog.setPositiveButton("Yes"){ alertDialog, which->
            readtoaccept(requestId,"Accepted",demail.toString(),num.toString())
            alertDialog.dismiss()
        }
        alertDialog.setNegativeButton("No"){ alertDialog,which->
            readtoaccept(requestId,"Rejected",demail.toString(),num.toString())
            alertDialog.dismiss()
        }
        alertDialog.show()
    }
}
```

```

    }

    btncompleted.setOnClickListener {
        readtoaccept(requestId,"Completed",detail.toString(),num.toString())
    }

    }

    }

    }

    private val bind by lazy {
        CarddoccostBinding.inflate(layoutInflater)
    }
    private fun readtoaccept(requestId: String?, status: String,
detail:String,unum:String) {
        if(status=="Accepted"){
            BottomSheetDialog(context).apply {
                setContentView(bind.root)

                bind.btnadcost.setOnClickListener {
                    val cost=bind.etcost.text.toString().trim()
                    val descri=bind.etedescription.text.toString().trim()

                    if(cost.isEmpty()){
                        bind.etcost.error="Enter your Cost"
                    }else if(descri.isEmpty()){
                        bind.etedescription.error="Enter Your Description"
                    }else{
                        CoroutineScope(Dispatchers.IO).launch {

RetrofitClient.instance.updatestatusdoc(status,cost,descri,requestId!!.toInt(),"updatestatus
doc")

                            .enqueue(object: Callback<DefaultResponse> {
                                override fun onFailure(call: Call<DefaultResponse>, t:
Throwable) {

```

School of Computer Science engineering & Information Science, Presidency University.

```
}}
```

```
}
```

```
}
```

```
}
```

```
override fun getItemCount() = listdata.size  
}
```

```
// get the part
```

```
private fun readvieworders(email: String) {  
    val builder = AlertDialog.Builder(this,R.style.TransparentDialog)  
    val inflater = this.layoutInflater  
    builder.setView(inflater.inflate(R.layout.progressdialog, null))  
    builder.setCancelable(false)
```

```
    progressDialog = builder.create()  
    progressDialog.show()
```

```
    CoroutineScope(Dispatchers.IO).launch {  
        RetrofitClient.instance.getdoctororders(email,"vieworders")  
            .enqueue(object : Callback<requestresponse> {  
                override fun onResponse(call: Call<requestresponse>, response:  
Response<requestresponse>) {
```

```
b.listorder.adapter=hospitalorderAdapter(this@DoctorDashboard,response.body()!!.user  
)
```

```
        b.listorder.layoutManager=  
LinearLayoutManager(this@DoctorDashboard)  
        Toast.makeText(this@DoctorDashboard,  
"$ {response.body()!!.user},$email", Toast.LENGTH_SHORT).show()  
        progressDialog.dismiss()  
    }  
    override fun onFailure(call: Call<requestresponse>, t: Throwable) {  
        Toast.makeText(this@DoctorDashboard, "$ {t.message}",  
    }  
}
```

Sustainable Development Goals



Goal 3: Health and Well-Being

The health care platform plays its part in bringing health and well-being by discussing health services from a single platform where users can access them easily, and therefore get medical aids when needed. In this way, it improves the communication process among users, hospitals, laboratories, medical shops, and ambulance services, increasing efficiencies in service delivery.

Goal 9: Industry, Innovation and Infrastructure

From setting up of the infrastructure in healthcare to innovatively empowering it with enhanced infrastructure using modern technologies such as Django, React.js, and Progressive Web Apps (PWAs), the project combines several healthcare service providers and strikes a balanced ground between urban and rural areas.

Goal 10: Reduced Inequalities

The portal connects the urban and rural societies by providing equitable healthcare services. It therefore ensures that healthcare and medical service provision reach underserved communities, thus solving their shortcomings in accessing healthcare services.

Yamanappa_Group-21_Report

ORIGINALITY REPORT

7%

SIMILARITY INDEX

5%

INTERNET SOURCES

4%

PUBLICATIONS

4%

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to Presidency University

Student Paper

2%

2

Submitted to M S Ramaiah University of Applied Sciences

Student Paper

<1%

3

Kutub Thakur, Helen G. Barker, Al-Sakib Khan Pathan. "Artificial Intelligence and Large Language Models - An Introduction to the Technological Future", CRC Press, 2024

Publication

<1%

4

Mark W. Anderson, Michael G. Fox, Nicholas C. Nacey. "MRI of the Hand", Elsevier BV, 2025

Publication

<1%

5

Ton Duc Thang University

Publication

<1%

6

V. Sharmila, S. Kannadhasan, A. Rajiv Kannan, P. Sivakumar, V. Vennila. "Challenges in Information, Communication and Computing Technology", CRC Press, 2024

Publication

<1%

7	ceowebkly.com Internet Source	<1 %
8	www.apmonitor.com Internet Source	<1 %
9	Nagesh Bhat, Bhavna Jha Kukreja, Pankaj Kukreja, Hiroj Bagde. "chapter 13 Efficient Treatment Planning on Optimization Techniques Empowered by AI in Dentistry", IGI Global, 2024 Publication	<1 %
10	Submitted to Hofstra University Student Paper	<1 %
11	Oleksandr Kuznetsov, Emanuele Frontoni, Natalia Kryvinska, Sarychev Volodymyr, Tetiana Smirnova. "Harnessing edge computing for real-time cybersecurity in healthcare systems", Institution of Engineering and Technology (IET), 2024 Publication	<1 %
12	Submitted to BITS, Pilani-Dubai Student Paper	<1 %
13	aitoptools.com Internet Source	<1 %
14	www.coursehero.com Internet Source	<1 %
www.safaribooksonline.com		

15	Internet Source	<1 %
16	Submitted to Canadian University of Dubai Student Paper	<1 %
17	Submitted to Hankuk University of Foreign Studies Student Paper	<1 %
18	Submitted to South Dakota Board of Regents Student Paper	<1 %
19	Submitted to University of Ulster Student Paper	<1 %
20	bmcmedinformdecismak.biomedcentral.com Internet Source	<1 %
21	fdocuments.us Internet Source	<1 %
22	linnk.ai Internet Source	<1 %
23	mmcalumni.ca Internet Source	<1 %
24	pmc.ncbi.nlm.nih.gov Internet Source	<1 %
25	utpedia.utp.edu.my Internet Source	<1 %

Acceptance for Presentation Inbox x



International Conference <InternationalConference@ifim.edu.in>
to me, A ▾

Wed, Jan 15, 11:23 AM (5 days ago) ☆ 😊 ↶ ⋮

Dear Prof Ganesh Kumar Baddi,

We are delighted to inform you that your abstract (INTEGRATED HEALTH CARE PORTAL) has been accepted for presentation at the **International Conference on Managing and Disrupting Businesses in the Era of Artificial Intelligence (ICMBAI 2025)**.

As a selected participant, you are now invited to register your paper. We encourage you to complete the registration process promptly to secure your spot.

We are pleased to share that your paper has the potential to be published in the **IEEE Xplore** (Scopus-indexed), offering a unique opportunity to showcase your research to a global audience.

To proceed with your registration, please visit the official registration link: <https://rzp.io/rzp/2AQjleit>

We are also excited to announce that we are in discussions with IEEE Xplore Publishing to include your paper in the conference proceedings.

Should you have any queries or require assistance, please do not hesitate to reach out to us.

We look forward to your valuable contribution to ICMBAI 2025 and to welcoming you to this esteemed conference.

Thank you.

Best regards,