

INTEGRATED HEALTH CARE APPLICATION

A PROJECT REPORT

Submitted by,

**Mr. Nikhil Krishnamurthy Rohidekar
20201CBC0026**

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING (BLOCKCHAIN)

Under the guidance of,

**Dr. Srinivasan T R
Professor, School of CSE**

at



PRESIDENCY UNIVERSITY, BENGALURU

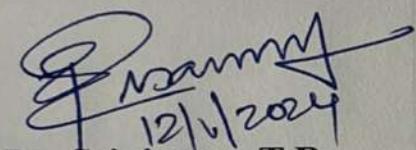
JANUARY 2024

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

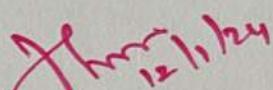
CERTIFICATE

This is to certify that the Project report "**INTEGRATED HEALTH CARE APPLICATION**" being submitted by "**NIKHIL K ROHIDEKAR**" bearing roll number "**20201CBC0026**" in partial fulfilment of requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering (Blockchain) is a bonafide work carried out under my supervision.



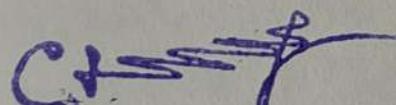
12/1/2024

Dr. Srinivasan T R
Professor
School of CSE
Presidency University

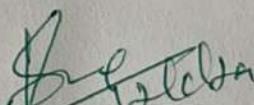


12/1/2024

Dr. S Senthil Kumar
Professor & HoD
School of CSE
Presidency University



Dr. C. KALAIARASAN
Associate Dean
School of CSE&IS
Presidency University



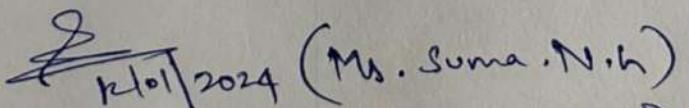
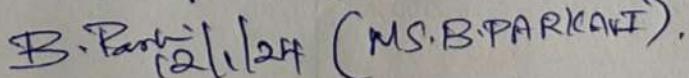
Dr. L. SHAKKEERA
Associate Dean
School of CSE&IS
Presidency University



Associate Dean

Dr. Md. SAMEERUDDIN KHAN
Dean
School of CSE&IS
Presidency University

Name and Signature of the Examiners

- 1) 
12/1/2024 (Ms. Suma N. h)
- 2) 
12/1/2024 (Ms. B. Parvathi).

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

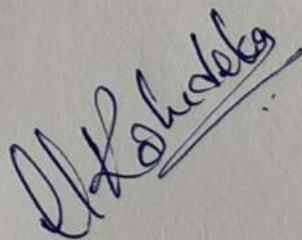
DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **INTEGRATED HEALTH CARE APPLICATION** in partial fulfilment for the award of Degree of **Bachelor of Technology in Computer Science and Engineering (Blockchain)**, is a record of our own investigations carried under the guidance of **DR. SRINIVASAN TR, Professor, School of Computer Science Engineering, Presidency University, Bengaluru.**

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

Name: - Nikhil K Rohidekar,

Roll No: -20201CBC0026



Signature of the Student

ABSTRACT

The Integrated Health Care system emerges as a boon for the ordinary individual seeking medical guidance or undergoing procedures. This platform facilitates the seamless discovery of hospitals and doctors, ensuring a comprehensive array of consultation services. Notably, it showcases healthcare providers offering diagnostic tests at regulated prices, maintaining a commitment to quality. The application, both free and customizable, serves as a crucial tool for patients and doctors alike. It enables them to effortlessly update their information, establishing a comprehensive database housing the profiles of both healthcare providers and patients. This user-friendly and adaptable Android application has undergone rigorous evaluation, affirming its ease of use and customization capabilities to meet diverse user needs. The practical implications of this innovative application are highly promising, catering to the requirements of various user categories. By providing a centralized platform for information exchange, the application contributes significantly to the enhancement of healthcare maintenance. In essence, The Integrated Health Care system stands as a testament to the potential of technology in revolutionizing and optimizing healthcare accessibility and management.

KEYWORDS: Mobile application, Android.

ACKNOWLEDGEMENT

First of all, we are indebted to the **GOD ALMIGHTY** 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**, School of Computer Science and Engineering & School of Information Science, Presidency University for getting us permission to undergo the project.

We record our heartfelt gratitude to our beloved Associate Deans **Dr. Kalaiarasan C** and **Dr. Shakkeera L**, School of Computer Science and Engineering & School of Information Science, Presidency University and **Dr. S Senthilkumar**, Professor & Head of the Department, School of Computer Science and Engineering, Presidency University for rendering timely help for the successful completion of this project.

We are greatly indebted to our guide **Dr. Srinivasan T R**, Professor, School of Computer Science Engineering, Presidency University for his inspirational guidance, valuable suggestions and 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 University Project-II Coordinators **Dr. Sanjeev P Kaulgud**, **Dr. Mrutyunjaya MS** and also the department Project Coordinator **Dr. Srinivasan T R**.

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

Nikhil K Rohidekar

LIST OF TABLES

Sl. No.	Table Name	Table Caption	Page No.
1	Table 1.1	Literature survey Table	4
2	Table 7.1	Gantt Chart	19
3	Table 9.1	Test Cases	48

Figure 6.1

Figure 7.1

Figure 8.1

Figure 9.1

Figure 9.2

Figure 9.3

Figure 9.4

Figure 9.5

Figure 9.6

LIST OF FIGURES

Sl. No.	Figure Name	Caption	Page No.
1	Figure 4.1	DFD Diagram	11
2	Figure 4.2	UML Diagram Component	12
3	Figure 4.3	UML Diagram Deployment	12
4	Figure 4.4	ER Diagram	12
5	Figure 7.1	Gantt Chart	19
6	Figure 8.1	Android Architecture	20
7	Figure 9.1	Outcome Architecture	36
7	Figure 9.2	Block Diagram	36
7	Figure 9.2	Use Case Diagram	37
8	Figure 9.3	Class Diagram	38
9	Figure 9.4	Sequence Diagram	39
10	Figure 9.5	Collaboration Diagram	40
11	Figure 9.6	Activity Diagram	41

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iv
	ACKNOWLEDGMENT	v
	LIST OF TABLES	vi
	LIST OF FIGURES	vii
	TABLE OF CONTENTS	viii
1.	INTRODUCTION	1
	1.1 Problem Statement	1
	1.2 Why is the Particular Topic Chosen?	2
	1.3 Motivation	3
	1.4 Scope	3
2.	LITERATURE SURVEY	4
	2.1 Related Work	4
	2.2 Proposed System	5
	2.3. Functionality	6
	2.3.1 Admin Functionality	6
	2.3.2 User Functionality	6
	2.3.3 Hospital Functionality	7
	2.3.4 Doctor Functionality	7
3	RESEARCH GAPS OF EXISTING METHODS	8
4	PROPOSED MOTHODOLOGY	11
5	OBJECTIVES	13
6	SYSTEM DESIGN & IMPLEMENTATION	14
	6.1 Software Installation for Android Project	14
	6.1.2. Android Studio Ide and Sdk Installation	15
	6.1.3 Setting Up Android Developer Options:	17
7	TIMELINE FOR EXECUTION OF PROJECT (GANTT CHART)	19
8	SOFTWARE ENVIRONMENT	20
	8.1 Architecture	30
9	OUTCOMES	36

10	RESULTS AND DISCUSSIONS	42
	9.1 Feasibility Study	42
	9.2 Importance of Healthcare Informatics	43
11	SYSTEM STUDY AND TESTING	46
12	CONCLUSION	49
13	FUTURE ENHANCEMENT	50
	REFRENCES	51
	APPENDIX-A - SCREENSHOTS	52
	APPENDIX-B - ENCLOSURES	58

In the dynamic landscape of healthcare, it is imperative to stay ahead of the curve and ensure that our system remains efficient, effective, and accessible. This study aims to address the challenges faced by the healthcare industry, particularly in terms of efficiency and accessibility. By leveraging the latest technologies and methodologies, we can create a platform that not only enhances the quality of care provided but also improves the overall user experience. The proposed system will be a comprehensive solution that addresses the needs of both healthcare providers and patients, ultimately leading to better outcomes and improved patient satisfaction. As we continue to evolve and refine our approach, we are confident that our system will become a valuable asset to the healthcare industry, providing a much-needed solution to the challenges faced by the sector.

CHAPTER-1

INTRODUCTION

1.1 Problem Statement-MHRD – IIC

Develop the concept of an integrated Health Care portal with App that helps common man to look for a hospital or a doctor for any consultation needs or undergoing medical procedure. A platform that offers service providers approved for offering diagnostic tests at regulated price and quality. This portal could be linked to IRDA approved healthcare insurance companies with clear mention of what is included and what is not, Deductibles, etc.

In the dynamic landscape of healthcare, the pursuit of innovative solutions is imperative to meet the evolving needs of both patients and healthcare providers. The motivation behind our project stems from the increasing demand for a more seamless, user-centric approach to healthcare services. As technology continues to redefine various facets of our lives, it is crucial to leverage its potential to enhance the efficiency and accessibility of healthcare. The problem at hand revolves around the limitations inherent in existing healthcare systems. Despite significant advancements in medical technology, the interfaces connecting patients and providers often fall short in terms of user-friendliness. This gap impedes effective communication and engagement between patients and healthcare professionals, impacting the overall quality of healthcare delivery. Navigating through appointments, accessing pertinent information, and fostering a collaborative environment between patients and providers are aspects that demand attention and improvement. Our project is poised to address this critical problem by developing a comprehensive healthcare platform that prioritizes usability, accessibility, and user engagement. The scope of our endeavor encompasses the creation of an intuitive and robust system that not only simplifies the process of booking appointments but also facilitates effective

knowledge sharing between healthcare providers and patients. We aim to bridge the existing gaps in communication, ensuring that the healthcare experience becomes not only more efficient but also more personalized and user-friendly. The proposed platform seeks to redefine the conventional patient-provider interaction model by introducing features that empower users to navigate their healthcare journey with ease. From streamlined appointment booking to a centralized database for storing patient and provider profiles, our project envisions a holistic approach to healthcare management. By embracing the latest technological advancements, we aspire to cultivate an environment where healthcare is not just a service but an experience tailored to the unique needs of each individual. In essence, this project endeavors to contribute to the ongoing transformation of the healthcare landscape, envisioning a future where technology serves as a catalyst for creating a more patient-centric, efficient, and accessible healthcare ecosystem.

This healthcare portal is more than just an app, it is a comprehensive platform designed to connect individuals with healthcare providers, making the search for hospitals and doctors a seamless and informed process. Our mission is to ensure that the common man has easy access to reliable healthcare services, while also offering a directory of service providers approved to conduct diagnostic tests at regulated prices and with uncompromising quality. The app will not only simplify the process of finding the right healthcare facilities and professionals but also provide a transparent and user-friendly interface.

1.2 WHY IS THE PARTICULAR TOPIC CHOSEN?

The topic of the Integrated Health Care Android Application was chosen due to the pressing need for a comprehensive and accessible healthcare solution. Many individuals struggle to find reliable healthcare providers and diagnostic services at reasonable prices. This project aims to bridge that gap by offering a user-friendly, customizable app that connects patients with doctors and hospitals

while ensuring affordability and quality. It also addresses the need for a centralized database for healthcare profiles, making it easier for both patients and doctors to manage their healthcare information efficiently. Ultimately, this project seeks to improve healthcare accessibility and overall well-being for all users.

1.3 MOTIVATION

The escalating demand for seamless healthcare necessitates innovative solutions. Bridging the gap in current healthcare systems motivates the exploration of streamlined, user-centric platforms.

1.4 SCOPE

This project aims to develop a comprehensive healthcare platform, addressing usability issues and enhancing patient-provider engagement. The scope encompasses creating an intuitive system that revolutionizes the healthcare experience for both users and providers.

2. RELATED WORK

A Sustainable Patient Application for Integrated Health Care

Md. Rashedur R. Islam, Md. S. Md. Shafiqul Islam, F. Farhanuzzaman, Md. Alim, Robia Roy

Affordability, mobility, and accessibility are three major requirements for providing services to the smart city community. Considering the impact of the

CHAPTER-2

LITERATURE SURVEY

Year	Author	Title	Algorithm	Pros	Cons
2019	Md. Nasfikur R. Khan; A K E H Mashuk; Whomaira F Durdana; Mehdi Alam; Robin Roy	A Customizable Android Application for Integrated Health Care	Random Forest Algorithm	High Accuracy, Robust to Overfitting	Complexity, Computational Cost
2016	Christopher Kulanga; Eric Saforo; Steve Ollis; Marc Mitchell	Integrating mHealth application	Combination of SVM and KNN Algorithm	Versatility, Regularization, Non-parametric, No Training Phase.	Computational Cost, Sensitive to Outliers.
2018	Mohammed Alkhawlan; Wesam Ali Husien; Saba Noori Alhamdany	Facilitating Patient Registrations Using an Integrating Healthcare Management System	Support Vector Machine Algorithm	Effective in High-Dimensional Spaces.	Computational Intensity, Choosing the Right Kernel

Table 1.1 Literature survey Table

2.1 RELATED WORK

A Customizable Android Application for Integrated Health Care

Md. Nasfikur R. Khan; A K E H Mashuk; Whomaira F Durdana; Mehdi Alam; Robin Roy

Affordable mobile technologies are introducing newer communication possibilities to the smart phone users every day. Considering the impact of this

incredibly popular mobile technology on the users, an android application named “Doctor who?” has been designed and developed, which makes an easy and effective communications with the users to the doctors and hospitals.

Integrating mHealth application

Christopher Kulanga; Eric Saforo; Steve Ollis; Marc Mitchell

mHealth has revolutionized the health care service delivery at all levels of health care system in both developing and developed world. Both at patient level and service provider level, mHealth has become very useful in areas of decision support, data storage, data processing into useful information, information flow and information use just to mention a few. For service providers with multiple service deliveries, there is a challenge in managing well the various mHealth applications across the various program deliveries.

Facilitating Patient Registrations Using an Integrating Healthcare Management System

Mohammed Alkhawlani; Wesam Ali Husien; Saba Noori Alhamdany

In addition, this computer application gives the patient the opportunity to use the system online and confirm their own appointments by themselves a matter that saves the time and effort of both the patient and the nurse. Moreover, by this computer application patient will be able to check the status of their appointment and doctor availability online.

2.2 PROPOSED SYSTEM

The app will show the doctor and the hospitals the user will fix the appoints with the doctor. The user can give the feedback of the hospitals. The user can view the cost of the appoint before booking. The doctors will view the books and give the response the user. The user pays the payments in the app itself.

Advantages:

1. Improved Efficiency:

- The application streamlines healthcare processes, reducing administrative burdens and enhancing overall efficiency.

2. Reduces Scope of Error:

- By automating appointment scheduling and payment processes, the system minimizes the likelihood of errors in data entry and transaction processing.

3. Increased Data Security & Retrieve-ability:

- Robust security measures are implemented to safeguard sensitive healthcare data, ensuring data integrity and accessibility.

4. Improved Patient Care:

- The direct interaction between users and doctors, coupled with efficient appointment management, contributes to an elevated standard of patient care.

2.3 Functionality

2.3.1 Admin Functionality:

Administrators are central to the effective functioning of healthcare management applications. Their responsibilities encompass logging in, adding hospitals, viewing hospital information, and managing user feedback. Additionally, administrators play a vital role in adding and updating healthcare insurance information. The literature underscores the significance of a well-structured administrative interface in healthcare applications. This interface enables administrators to efficiently oversee healthcare institutions, gather essential feedback from both patients and medical professionals, and improve service quality. A responsive system that ensures data security is essential in fulfilling the evolving needs of healthcare administration.

2.3.2 User Functionality:

Users are at the heart of these applications, and their interaction with the platform is characterized by functionalities such as registration, login, hospital and doctor selection, appointment scheduling, tracking appointment status, and easy access to healthcare

information. Research highlights the importance of user-centered design in healthcare applications, with a focus on creating interfaces that enable seamless navigation, improving the overall healthcare experience. The inclusion of features such as appointment scheduling and status tracking contributes to increased patient engagement and empowerment, ultimately impacting healthcare outcomes.

2.3.3 Hospital Functionality:

Hospitals using the application can log in, manage their profiles by adding doctors, view appointment schedules, and access user feedback. Studies emphasize the pivotal role of digital tools in hospital administration, demonstrating their potential to enhance operational efficiency. The ability to manage doctors and appointments electronically optimizes resource allocation and organization within healthcare institutions. Monitoring and addressing user feedback provides valuable insights that hospitals can use to enhance the quality of their services, ultimately benefiting patient care.

2.3.4 Doctor Functionality:

Doctors using the application have the ability to log in, access appointment schedules, accept or reject appointment requests, and update their availability status. Effective doctor-patient communication is vital, and the literature highlights the value of healthcare applications in enabling doctors to manage their appointments electronically. This improves scheduling efficiency and patient satisfaction. The capability for doctors to update their availability status ensures that patients have access to real-time information regarding the availability of their healthcare providers.

CHAPTER-3

RESEARCH GAPS OF EXISTING METHODS

Research Gaps in Integrated Healthcare Application Functionalities:

Admin Functionality:

Advanced User Feedback Analysis:

Current methods may lack sophisticated techniques for sentiment analysis and natural language processing in user feedback, hindering administrators' ability to extract nuanced insights for decision-making.

Automated Healthcare Insurance Management:

The process of adding and updating healthcare insurance information may not be streamlined and automated, posing challenges to accuracy and regulatory compliance.

Security and Access Control Measures:

The evolving landscape of cybersecurity threats necessitates ongoing research to implement state-of-the-art security measures and access control mechanisms, ensuring the confidentiality of healthcare administration data.

Optimal User Training and Support:

Research is needed to assess and enhance the effectiveness of training programs and support mechanisms for administrators, ensuring optimal utilization of the system and minimizing errors.

User Functionality:

Personalized Healthcare Experience:

Research should explore methods for personalizing the user experience based on individual health profiles, preferences, and historical data to enhance patient satisfaction and operational efficiency.

engagement and adherence to healthcare plans.

Integration with Wearables for Health Monitoring:

Investigate the seamless integration of wearable devices to contribute real-time health data, fostering a more comprehensive and personalized approach to healthcare management.

Usability for Diverse User Groups:

Current applications may not adequately address the usability needs of diverse user groups, including elderly individuals, individuals with disabilities, and those with varying levels of health literacy.

Ethical Considerations in Data Ownership:

Research is essential to address ethical considerations related to user data ownership and consent, ensuring transparency and user empowerment in managing their health information.

Hospital Functionality:

Optimizing Resource Allocation Algorithms:

Current methods may lack advanced algorithms to optimize resource allocation within hospitals, considering factors such as patient load, doctor availability, and equipment utilization.

Interoperability with Hospital Information Systems:

Explore methods to enhance interoperability between the integrated health care application and existing hospital information systems to facilitate seamless data exchange and reduce duplication of efforts.

Predictive Analytics for Patient Flow:

Investigate the implementation of predictive analytics to forecast patient flow within hospitals, aiding proactive resource management and enhancing overall

operational efficiency.

Privacy-preserving Data Sharing:

Research is needed to develop techniques for privacy-preserving data sharing between hospitals within the integrated system, ensuring collaboration while maintaining patient confidentiality.

Doctor Functionality:

Seamless Telemedicine Integration:

Explore the seamless integration of telemedicine functionalities within the application to facilitate remote patient consultations, especially in situations where in-person visits are challenging.

Integration of Clinical Decision Support Systems:

Investigate the integration of clinical decision support systems to assist doctors in making evidence-based decisions, ultimately improving the quality of patient care.

Enhanced User-friendly Appointment Management:

Assess the user-friendliness of appointment management interfaces for doctors, identifying areas for improvement to streamline the scheduling process further.

Patient Communication and Education Features:

Research should focus on enhancing doctor-patient communication through the application, incorporating features for sharing educational materials, treatment plans, and progress updates.

Addressing these research gaps will contribute to the ongoing development and enhancement of integrated healthcare applications, ultimately improving their effectiveness and impact on healthcare delivery and patient outcomes.

CHAPTER-4

PROPOSED METHODOLOGY

To achieve these objectives, a mixed-method research approach will be employed, involving both quantitative and qualitative data collection and analysis. This will include:

Surveys and questionnaires to gather user feedback and assess user satisfaction with the healthcare management application.

Interviews with healthcare administrators, doctors, and hospital staff to understand the impact of the application on their daily operations.

Data analysis of application usage patterns to assess user engagement and interaction with the platform.

Literature review to provide a comprehensive understanding of the state of the art in healthcare management applications and the impact of technology in the healthcare sector.

DFD Diagram

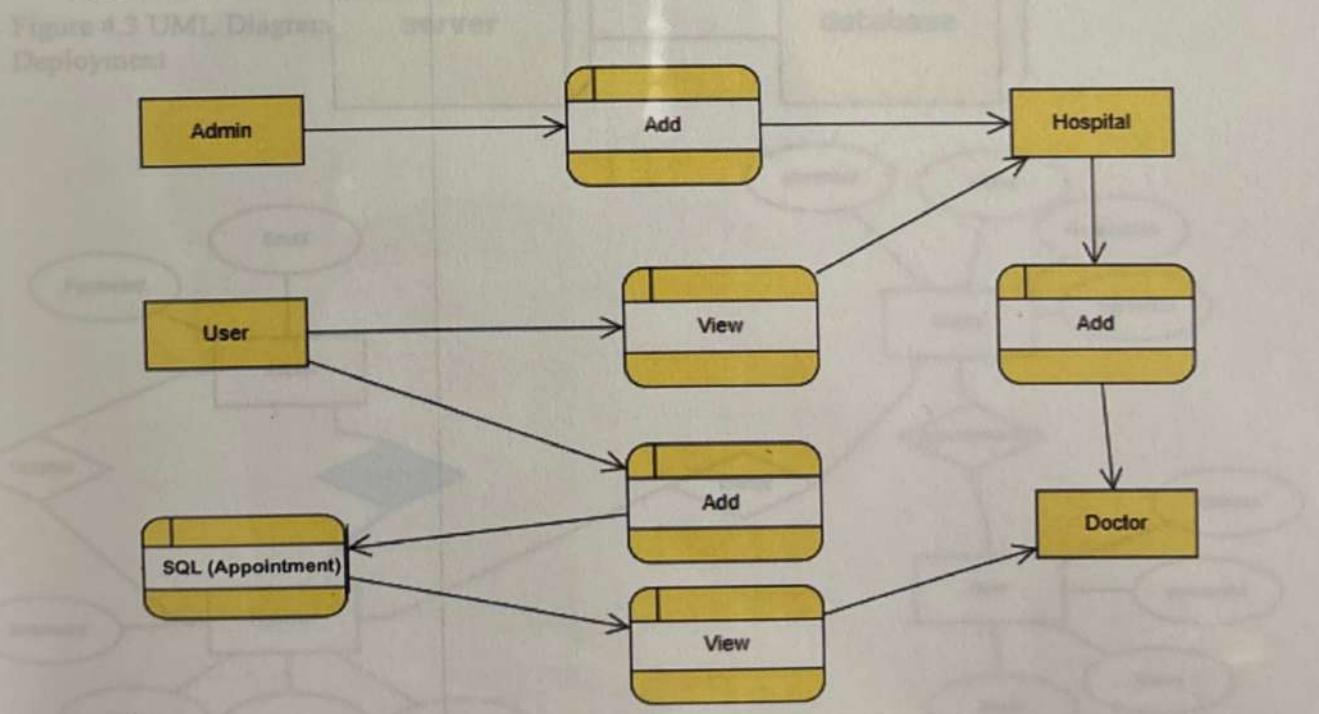


Figure 4.1 DFD Diagram

UML Diagram Component and Deployment

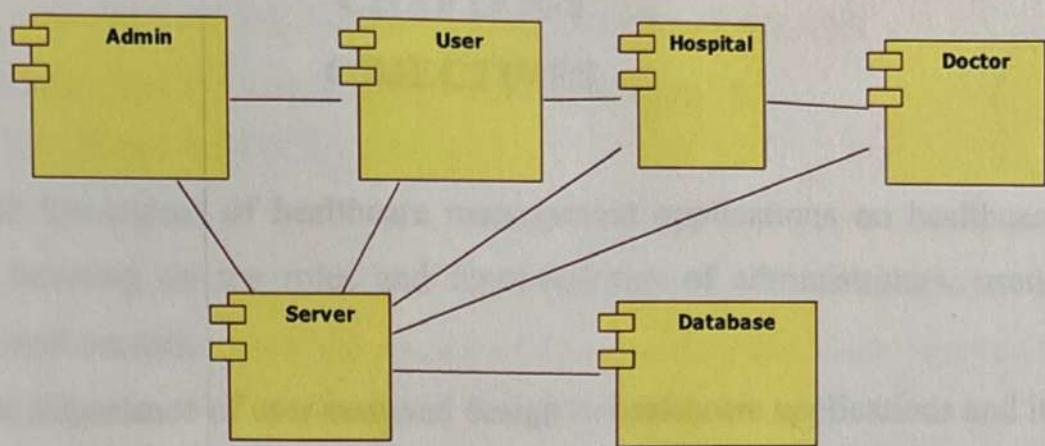


Figure 4.2 UML Diagram Component

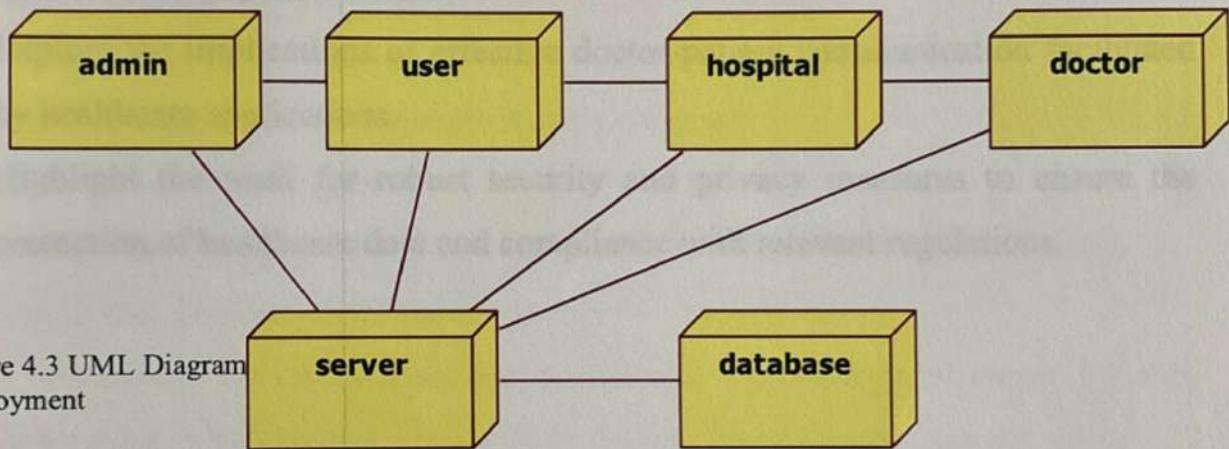


Figure 4.3 UML Diagram Deployment

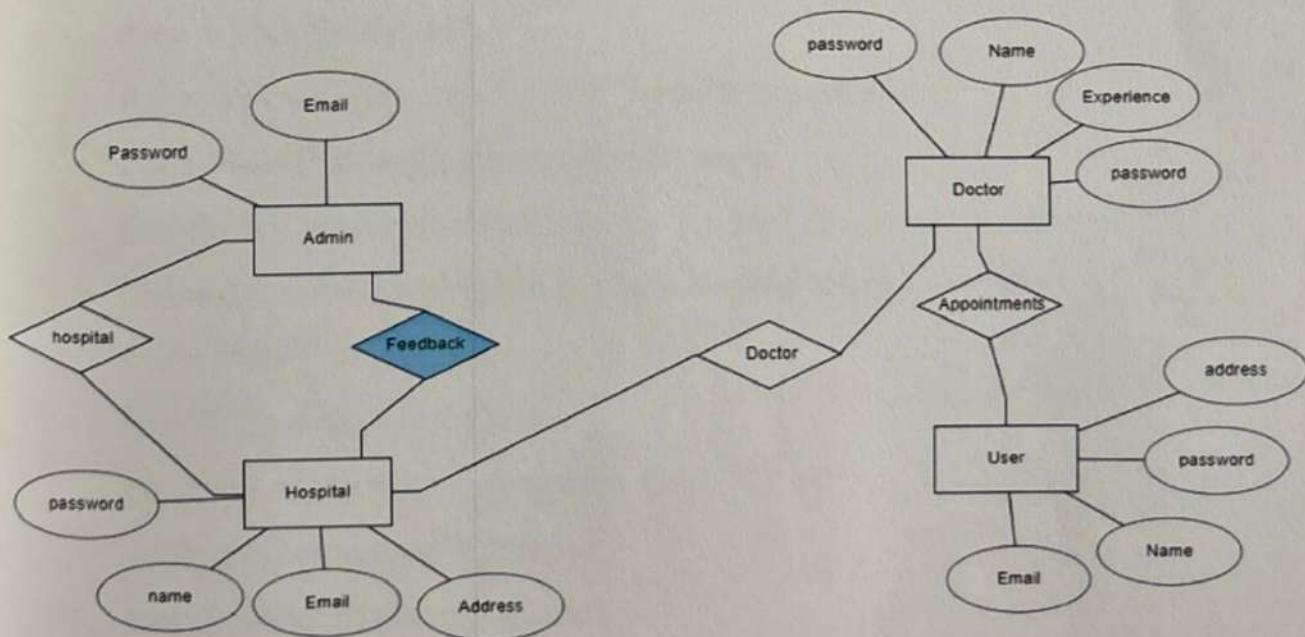


Figure 4.4 ER Diagram

CHAPTER-5

OBJECTIVES

Investigate the impact of healthcare management applications on healthcare delivery, focusing on the roles and functionalities of administrators, users, hospitals, and doctors.

Assess the importance of user-centered design in healthcare applications and its influence on patient engagement and healthcare outcomes.

Examine the contributions of digital tools in enhancing operational efficiency within healthcare institutions.

Explore the implications of effective doctor-patient communication facilitated by healthcare applications.

Highlight the need for robust security and privacy measures to ensure the protection of healthcare data and compliance with relevant regulations.

Step 1: Download Java (Optional)

If prompted, review and accept the Oracle Technology Network License Agreement. You will need to log in or sign up for an Oracle account.

Step 2: Run the Installer

After downloading, run the JDK installer executable.

Click "Next" through the installation steps.

Step 3: Set Installation Path

Choose the installation path for Java on your Windows system.

Click "Next" to proceed.

Step 4: Complete Installation

Once the installation is complete, click "Close."

Setting Environment Variables

Step 1: Open System Properties

Right-click on "My Computer" and select "Properties."

Step 2: Accept License Agreement

CHAPTER-6

SYSTEM DESIGN & IMPLEMENTATION

6.1 SOFTWARE INSTALLATION FOR ANDROID PROJECT:

Software Installation of JDK kit

The Java Development Kit (JDK) is a fundamental tool that allows coding and execution of Java programs. It's noteworthy that you can have multiple JDK versions on a single PC, but it is recommended to install the latest version for optimal performance.

Here is a step-by-step guide on how to install JDK 8 on a Windows system:

JDK Installation Steps:

Step 1: Download JDK 8

Visit the Oracle website and click on "JDK Download for Java."

Step 2: Accept License Agreement and Download

Accept the License Agreement.

Choose the appropriate JDK version (32-bit or 64-bit) and click Download.

Step 3: Oracle Account (if required)

If prompted, review and accept the Oracle Technology Network License Agreement. You may need to log in or sign up for an Oracle account.

Step 4: Run the Installer

After downloading, run the JDK installer executable.

Click "Next" through the installation steps.

Step 5: Set Installation Path

Choose the installation path for Java on your Windows system.

Click "Next" to proceed.

Step 6: Complete Installation

Once the installation is complete, click "Close."

Setting Environment Variables:

Step 1: Open System Properties

Right-click on "My Computer" and select "Properties."

Step 2: Access Advanced System Settings

Click on "Advanced system settings."

Step 3: Navigate to Environment Variables

In the System Properties window, click on "Environment Variables."

Step 4: Set PATH Variable

Click "New" in the User variables section.

Set Variable Name as "PATH."

Copy the path of the "bin" folder in the JDK installation directory.

Paste the path in Variable Value and click "OK."

Step 5: Set CLASSPATH Variable

Follow a similar process to set the CLASSPATH variable if needed.

Step 6: Verify Installation

Open the command prompt and type Java commands. If Java is installed correctly, you'll see a confirmation screen.

By following these steps, you ensure that Java is not only installed on your Windows system but also properly configured through environment variables, facilitating smooth execution of Java programs.

6.1.2 ANDROID STUDIO IDE AND SDK INSTALLATION

Embarking on the Android software installation journey for your project can be both exciting and challenging. Here's a step-by-step guide to assist you, taking into account the nuances of your PC, programming knowledge, and a bit of luck.

Step 0: Pre-Installation Check List

Before installing Android SDK, ensure you have Java Development Kit (JDK) installed, preferably version 1.8 or above.

Uninstall any older versions of "Android Studio" and "Android SDK."

Recognize that installations and operations may take considerable time. Utilize this time to explore resources like the "Android Developers" website for insights.

The process involves installing two substantial packages: Android Studio (IDE) and Android SDK (Software Development Kit).

Step 1: Install "Android Studio IDE"

Verify that the JAVA_HOME environment variable is set to the JDK installation directory.

Check system requirements, ensuring your PC meets the specified RAM, disk space, and screen resolution.

Visit "Android Studio" website, download the appropriate version for your Windows system, and run the installer.

During installation, choose components (Android Studio and Android Virtual Device) and set the installation location.

Launch Android Studio and proceed.

Setting Environment Variables: PATH and CLASSPATH

Right-click on "My Computer" > Select "Properties."

Access advanced system settings.

Navigate to Environment Variables.

Set PATH variable: Add the path of the "bin" folder in the JDK installation directory.

Set CLASSPATH variable if needed.

Step 2: Installing Android SDK

Launch Android Studio, run the setup wizard, and choose "do not import previous settings."

Follow the wizard through "Welcome," "Install Type," "Select UI Theme," and "SDK Components Setup."

Note the SDK directory and let the installation proceed.

Checking SDK Packages

Use Android Studio to check installed SDK packages under "SDK Manager."

Verify the presence of necessary SDK platforms and tools.

Remember, patience is key during installations, and exploring available resources can enhance your understanding of the Android development environment.

STEPS FOR EXECUTING THE PROJECTS

Step 1:

Open Android Studio

Step 2:

Choose a virtual device or Physical device from the menu

Step 3:

Click on the project Run

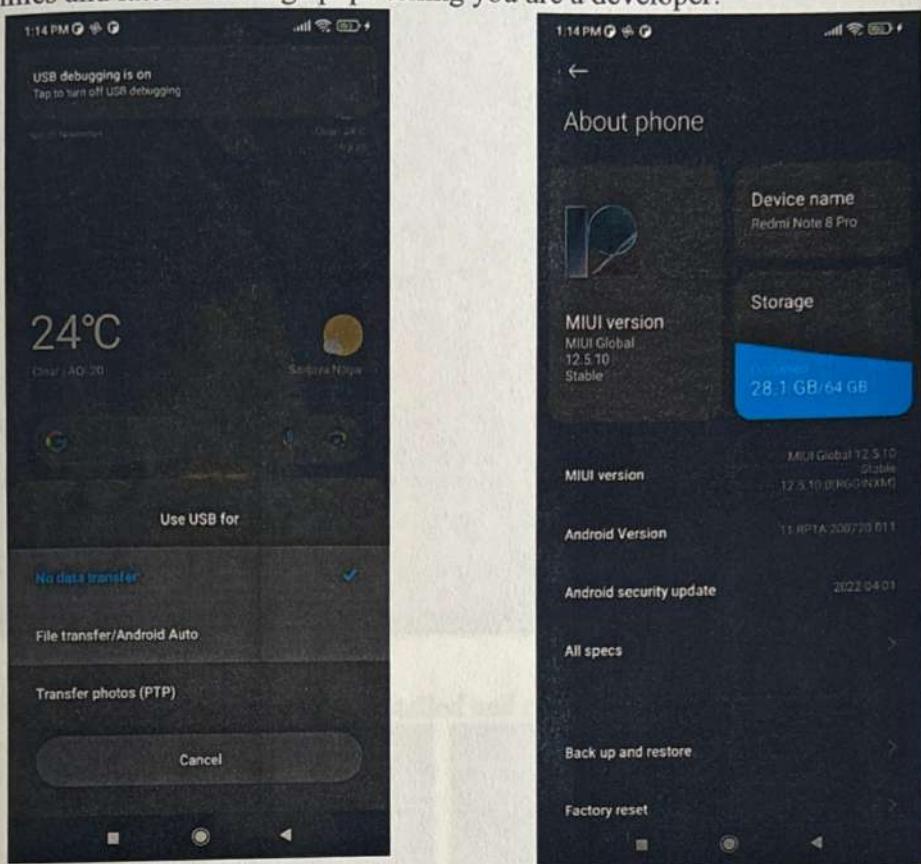
Step 4:

View the application performance on virtual or Physical device.

6.1.3 SETTING UP ANDROID DEVELOPER OPTIONS:

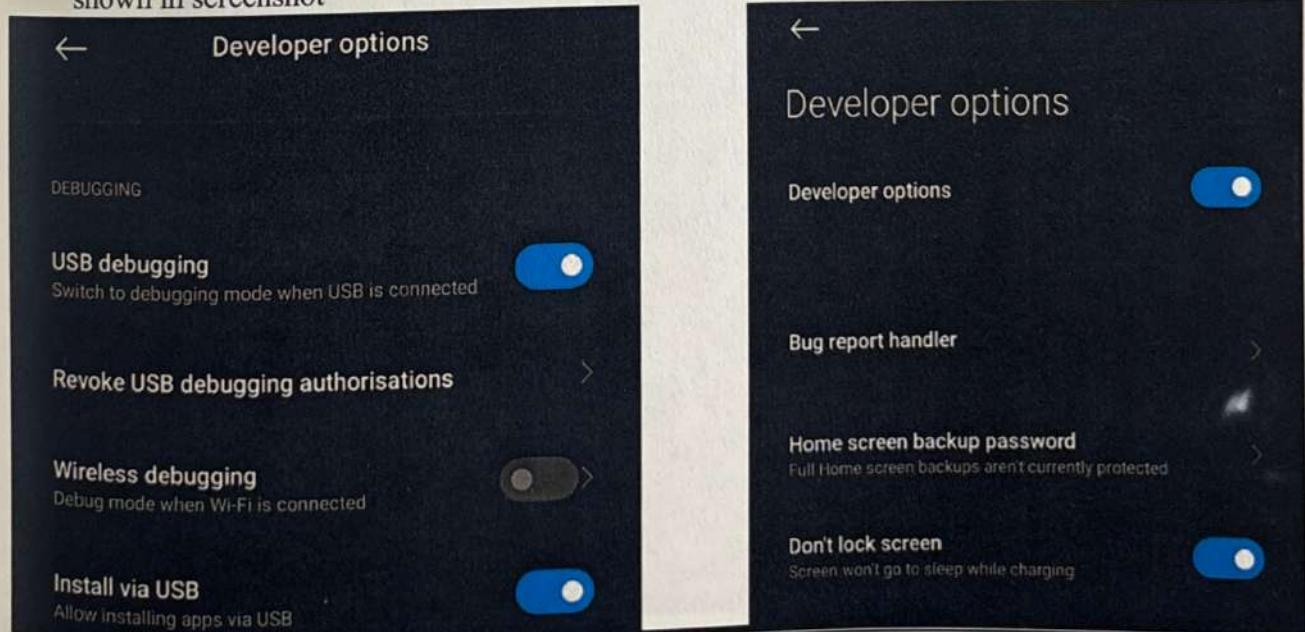
STEP 1: - Connect an Android Mobile to the Computer via USB Cable

Then switch on USB data Transfer, go to the settings choose about phone, click on the version for 5 times and later a message pops telling you are a developer!

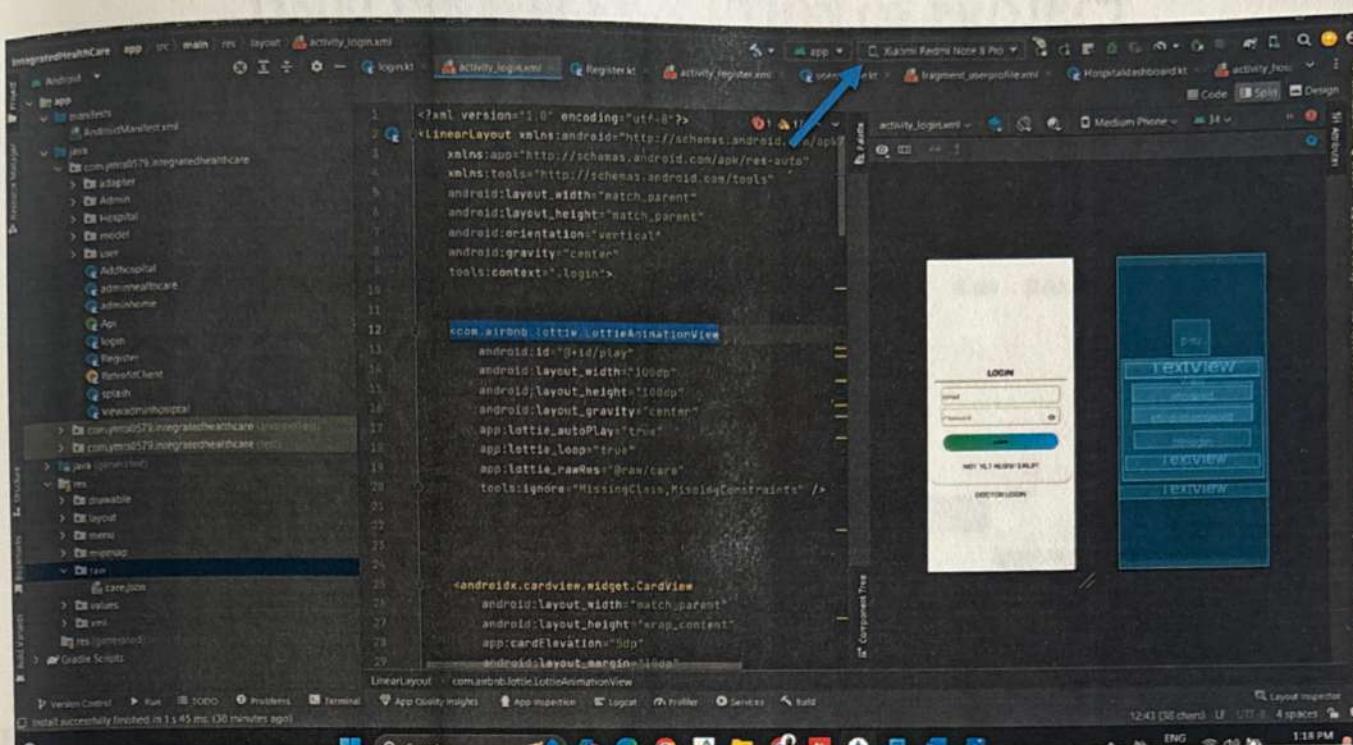


STEP 2: -

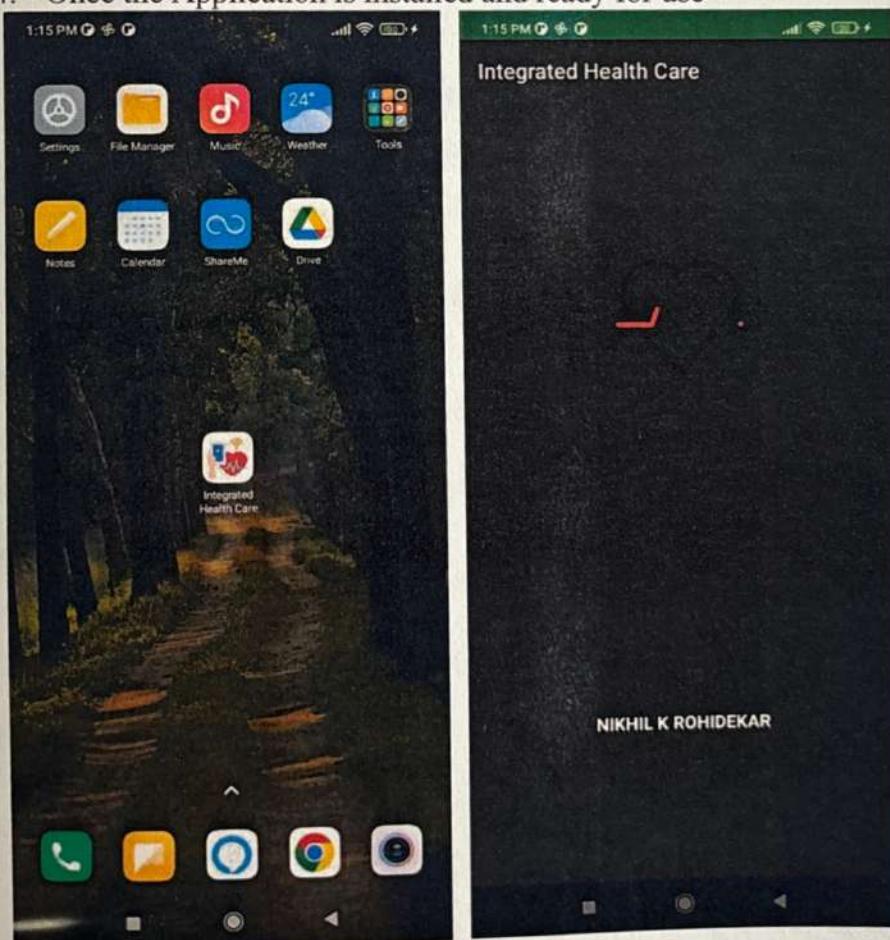
Search Developer options in in Settings and then Switch on USB debugging and scroll down and turn on Install via USB now we can see the mobile name at the top of android studio as shown in screenshot



STEP 3: - Now we can run the Application in the android studio to get the Application installed on your mobile, the arrow mark indicates the mobile connected via USB to Computer



STEP 4: - Once the Application is installed and ready for use



CHAPTER-7

TIMELINE FOR EXECUTION OF PROJECT (GANTT CHART)

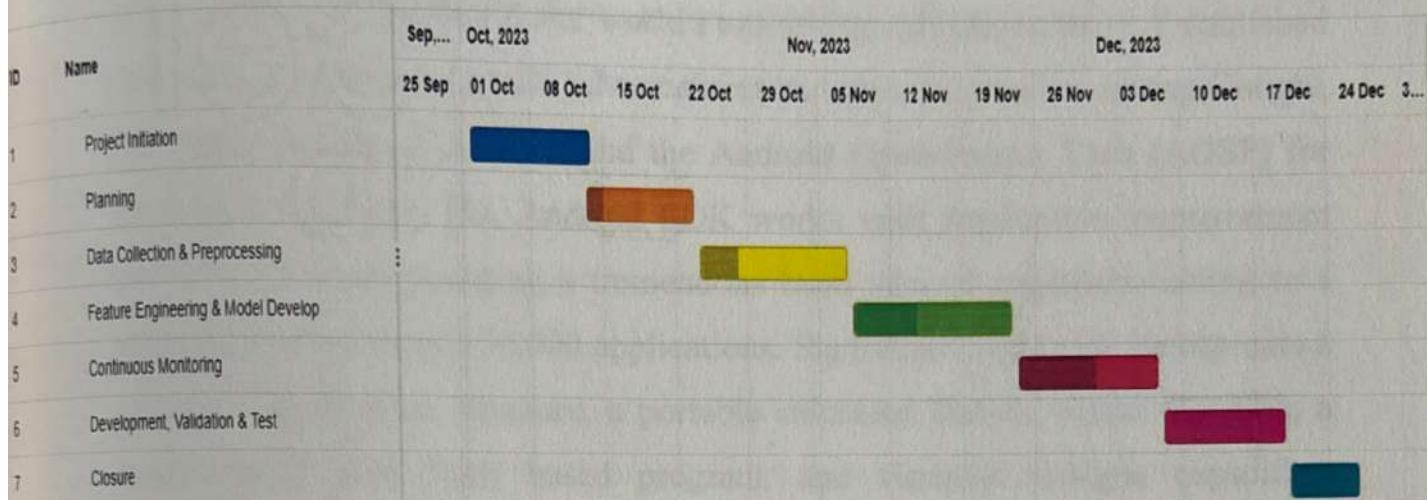


Figure 7.1 Gantt Chart

ID	Name	Start Date	End Date	Duration	Progress %
1	Project Initiation	Oct 02, 2023	Oct 13, 2023	10 days	0
2	Planning	Oct 13, 2023	Oct 23, 2023	7 days	15
3	Data Collection & Preprocessing	Oct 24, 2023	Nov 07, 2023	11 days	25
4	Feature Engineering & Model Develop	Nov 08, 2023	Nov 23, 2023	12 days	40
5	Continuous Monitoring	Nov 24, 2023	Dec 07, 2023	10 days	55
6	Development, Validation & Test	Dec 08, 2023	Dec 19, 2023	8 days	80
7	Closure	Dec 20, 2023	Dec 26, 2023	5 days	100

Table 7.1 Gantt Chart

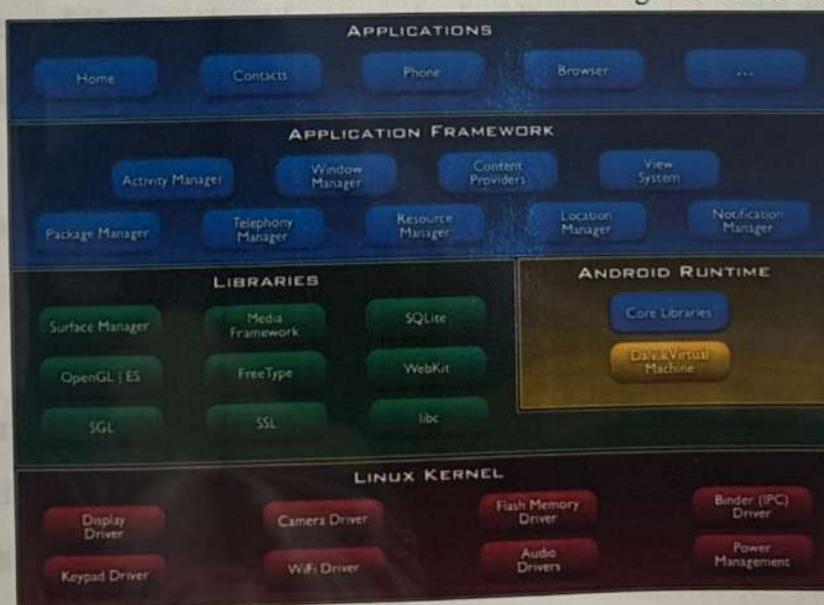
CHAPTER-8

SOFTWARE ENVIRONMENT

Android, a versatile programming stack, involves a working framework, middleware, and fundamental applications. Gained by Google in 2005 from Android Inc., it remains as the world's top-selling cell phone stage. Established in the Linux piece, Android's advancement includes cooperation among Google, the Open Handset Coalition, and the Android Open-Source Task (AOSP) for continuous upkeep. The Android SDK works with application improvement utilizing Java, encouraging a tremendous local area of engineers adding to a collection of north of 250,000 applications. Significant highlights incorporate a versatile application structure, a portable enhanced Dalvik virtual machine, a coordinated Web Pack based program, and vigorous designs capacities. Different functionalities incorporate SQLite for information capacity, media support, communication elements, and equipment subordinate credits like Bluetooth, GPS, and camera coordination. The rich improvement climate, comprehensive of troubleshooting apparatuses and a Shroud IDE module, highlights Android's noticeable quality in the versatile scene.

Android Architecture

Figure 8.1 Android architecture



Android Operation System

Android, a technological marvel underpinned by the Linux operating system and a Java programming interface, has emerged as a leading force in the mobile landscape. Crafted by the Open Handset Alliance, spearheaded by the tech titan Google, Android seamlessly integrates a suite of tools, including a compiler, debugger, and a device emulator, complemented by its Java Virtual Machine—the Dalvik Virtual Machine (DVM).

A distinctive feature of Android lies in its utilization of the Dalvik Virtual Machine, employing specialized bytecode. This unique characteristic translates to the inability to execute standard Java bytecode on Android. Enter "dx," a tool that performs the alchemical conversion of Java Class files into "dex" (Dalvik Executable) files, a process facilitated by Android's adept "aapt" (Android Asset Packaging Tool). The packaging of Android applications into the .apk (Android Package) file is orchestrated seamlessly during deployment, courtesy of Google's Android Development Tools (ADT) for Eclipse.

Android, with its prowess, extends support for both 2-D and 3-D graphics, leveraging the capabilities of OpenGL libraries. Data finds its abode in SQLite databases, adding a robust layer to Android's repertoire. A distinctive feature of the Android ecosystem is the encapsulation of each application within its process, operating under a unique user id generated dynamically during deployment. This isolation shields applications from external interference, embodying Android's commitment to security and robustness.

Important Android components

An Android application, much the same as a finely coordinated orchestra, involves an agreeable gathering of unmistakable components:

Movement: Filling in as the substance of an Android application, an Action addresses the show layer — basically the screens that clients collaborate with. Different exercises can powerfully switch during runtime.

Sees: Developing the UI of Exercises includes gadget classes that acquire from "android. view. View," with the design coordinated by "android. view. View Gatherings."

Administrations: These unrecognized yet truly great individuals perform foundation undertakings without a UI, utilizing the notice structure to caution clients unobtrusively.

Content Supplier: Offering the capacity to impart information to different applications, a Substance Supplier draws from Android's SQLite data set.

Plans: Offbeat couriers empowering usefulness demands, Expectations come in two Flavors — unequivocal (straightforwardly calling a help or action) and implied (mentioning administrations or applications enrolled for a specific goal).

Broadcast Collector: Sensitive to framework messages and certain purposes, it responds to developing framework conditions, set off by occasions an application register for.

JVM: The Loom of Interoperability

Hanging the Android environment in a shroud of interoperability, the Java Virtual Machine (JVM) arises as a crucial player:

WORA Standard: Java's "compose once, run anywhere" reasoning unfurls inside the JVM, executing bytecode across different equipment and programming stages.

JVM Execution: Past Prophet, JVMs under the "Java" brand name, sticking to Prophet's determinations, can be created by assorted elements. JVM stretches out help to dialects past Java, exemplified by Ada's similarity.

Java Runtime Climate (JRE): Packaged with standard class libraries, JRE includes APIs, working with consistent execution across different stages.

Paired Configuration: Projects bound for JVM should be gathered into a normalized double organization, frequently epitomized in .class or .container documents.

Java Application Launcher: The pervasive 'java' launcher presents a standard conductor for executing Java code.

Runtime Execution: JVM deciphers bytecode or use in the nick of time (JIT) compilers, similar to Prophet's Area of interest, for sped up execution. Quite bit early compilers offer a precompiled local code elective.

Engineering: Following a stack-based model suggestive of microcontrollers, JVM's quirky memory model incorporates Java-like classes and techniques, exemplifying a particular capacity-based design.

In the Android adventure, this gathering of parts and the JVM mix, making an orchestra of interoperability and productivity.

Download the Android SDK

If you're curious about the Android domain interestingly or looking to invigorate your tool compartment, here's a guide to set up the Android SDK, opening the doorway to inventive application improvement.

Outline of Arrangement Steps:

Set up Your Improvement PC:

Affirm that your PC lines up with the framework prerequisites, guaranteeing the fundamental Java Advancement Unit (JDK) is set up.

Introduce the SDK Starter Bundle:

Download the SDK starter bundle reasonable for your foundation. Assuming you're on Windows, influence the installer for smoothed out arrangement.

Incorporate ADT Module for Shroud:

For Obscuration aficionados, raise your improvement experience by introducing the Android Advancement Apparatuses (ADT) Module.

Extend Your SDK Weapons store:

Improve your SDK by integrating Android stages and extra parts utilizing the Android SDK and AVD Supervisor.

Dig into SDK Items (Discretionary):

Alternatively, investigate the items in the Android SDK, acquiring further experiences into the tool stash.

Jump into Establishment:

Stage 1: Setting up Your Improvement PC

Before the Android SDK shade rises, guarantee your PC is prepared:

Affirm framework prerequisites, including the JDK establishment.

In the event that Shroud is your improvement nook, confirm a viable form is cozy on your machine. "Obscure Work of art" is suggested, reachable from the Obscuration download area.

Stage 2: Introducing the SDK

To initiate your Android process:

Download the proper bundle custom fitted to your framework from the gave table.

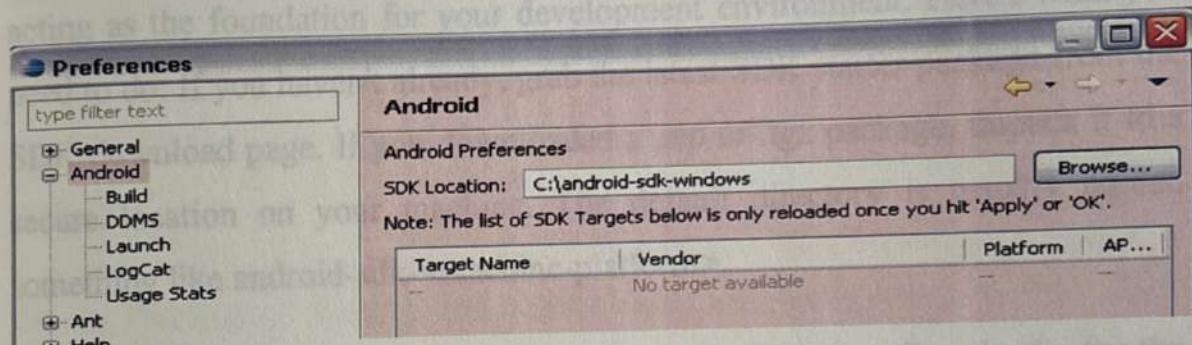
Follow the aide for Introducing the SDK, starting your vivid experience.

Design

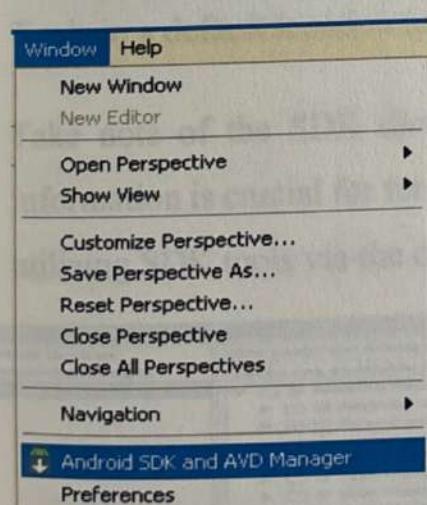
Assuming Shroud is your picked advancement safe house:

Explore to Inclinations in Obscuration by means of Windows -> Inclinations.

Investigate the Android segment, carefully setting the way for your Android SDK establishment.



Select Window -> Android SDK and AVD Manager from the menu.



Step 1: Selecting the Latest SDK Version

As you dive into the Android development universe, it's crucial to equip yourself with the latest tools. Follow these steps:

Launch the SDK Manager and select the available packages.

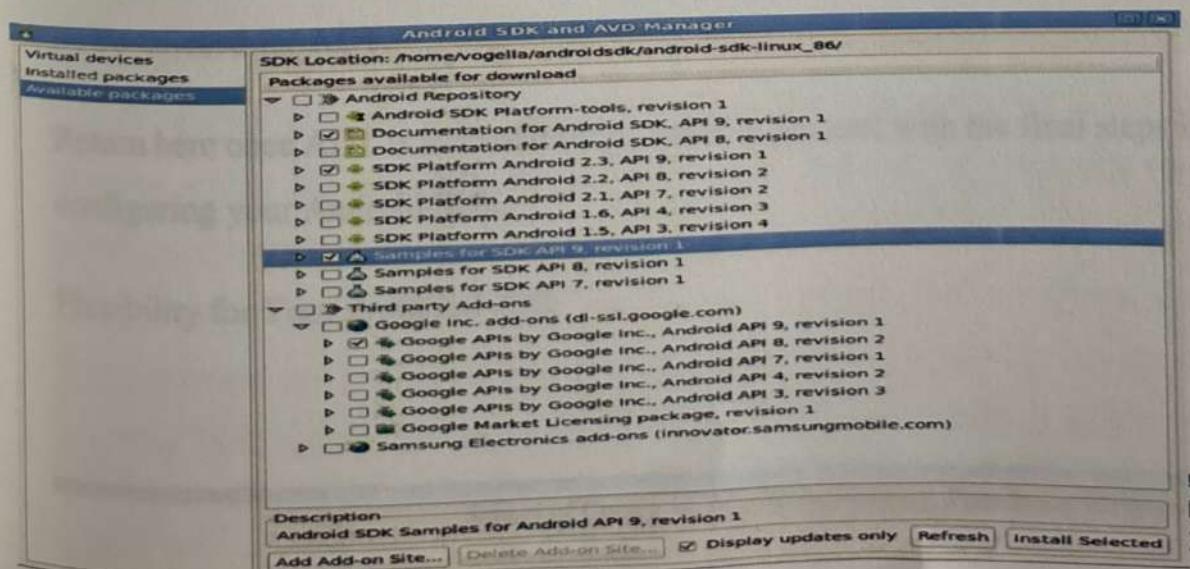
Opt for the latest version of the SDK to ensure you're harnessing the most advanced features.

Step 2: Downloading the SDK Starter Package

The SDK starter package is your gateway—it contains the essential SDK Tools, acting as the foundation for your development environment. Here's what you need to do: If you haven't already, grab the latest SDK starter package from the SDK download page. If you downloaded a .zip or .tgz package, unpack it to a secure location on your machine. The default directory is usually named something like android-sdk-<machine-platform>.

For Windows users with the .exe installer, run it. This installer checks for the Java SE Development Kit (JDK), installs it if needed, and then sets up the SDK Tools in a default location (modifiable by you).

Take note of the SDK directory's name and location on your system—this information is crucial for future reference when configuring the ADT plugin and utilizing SDK tools via the command line.



Step 3. Installing the ADT Plugin for Eclipse

Android beckons developers into a world of innovation, and to empower your journey, Google presents the Android Development Tools (ADT) plugin for the Eclipse Integrated Development Environment (IDE). This dynamic duo transforms Eclipse into a feature-rich hub, streamlining Android application development. Here's a glimpse into the prowess of ADT:

Why ADT?

Effortless Project Setup: Quickly initiate new Android projects with a seamless setup process.

Intuitive UI Design: Craft visually stunning application interfaces directly within Eclipse.

Robust Debugging: Identify and rectify issues efficiently using Android SDK tools.

APK Export: Easily export signed or unsigned APKs for convenient application distribution.

Installation Guide:

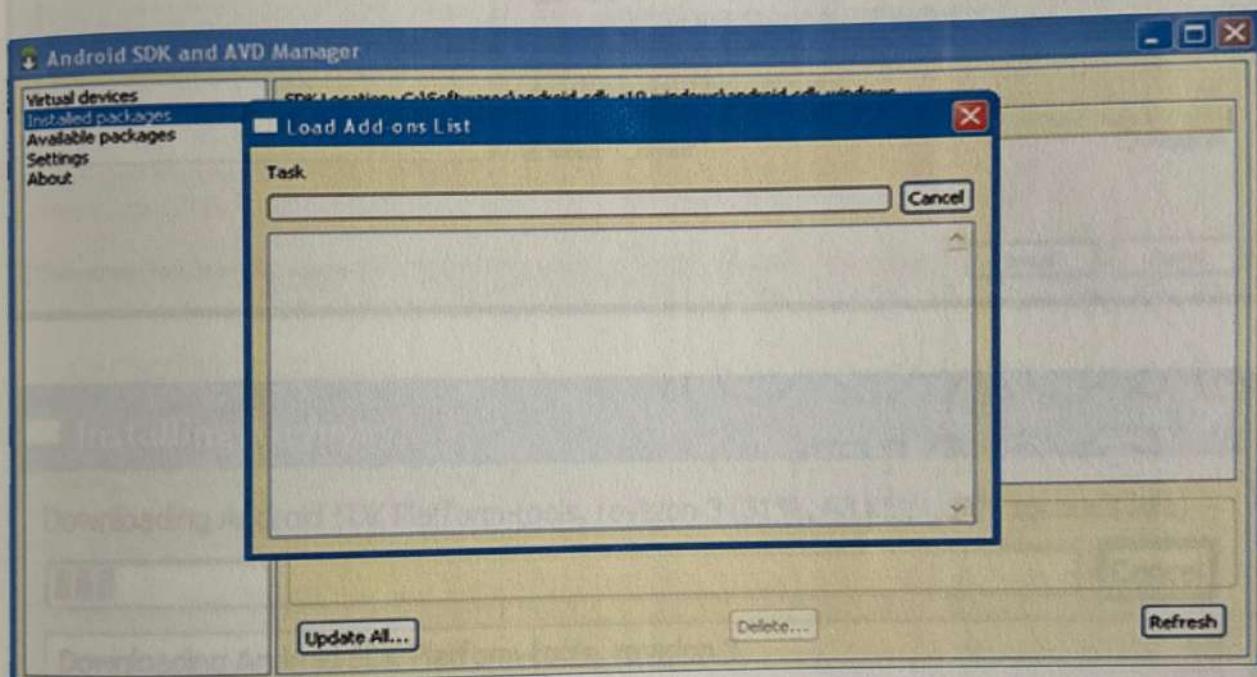
Navigate to "Installing the ADT Plugin" for a detailed, step-by-step installation guide tailored for ADT.

Return here once ADT is successfully installed to proceed with the final steps in configuring your Android SDK.

Flexibility for Your Preferred IDE:

ADT and Eclipse Harmony: If you choose ADT with Eclipse, you're opting for a swift and recommended path, accelerating your Android venture.

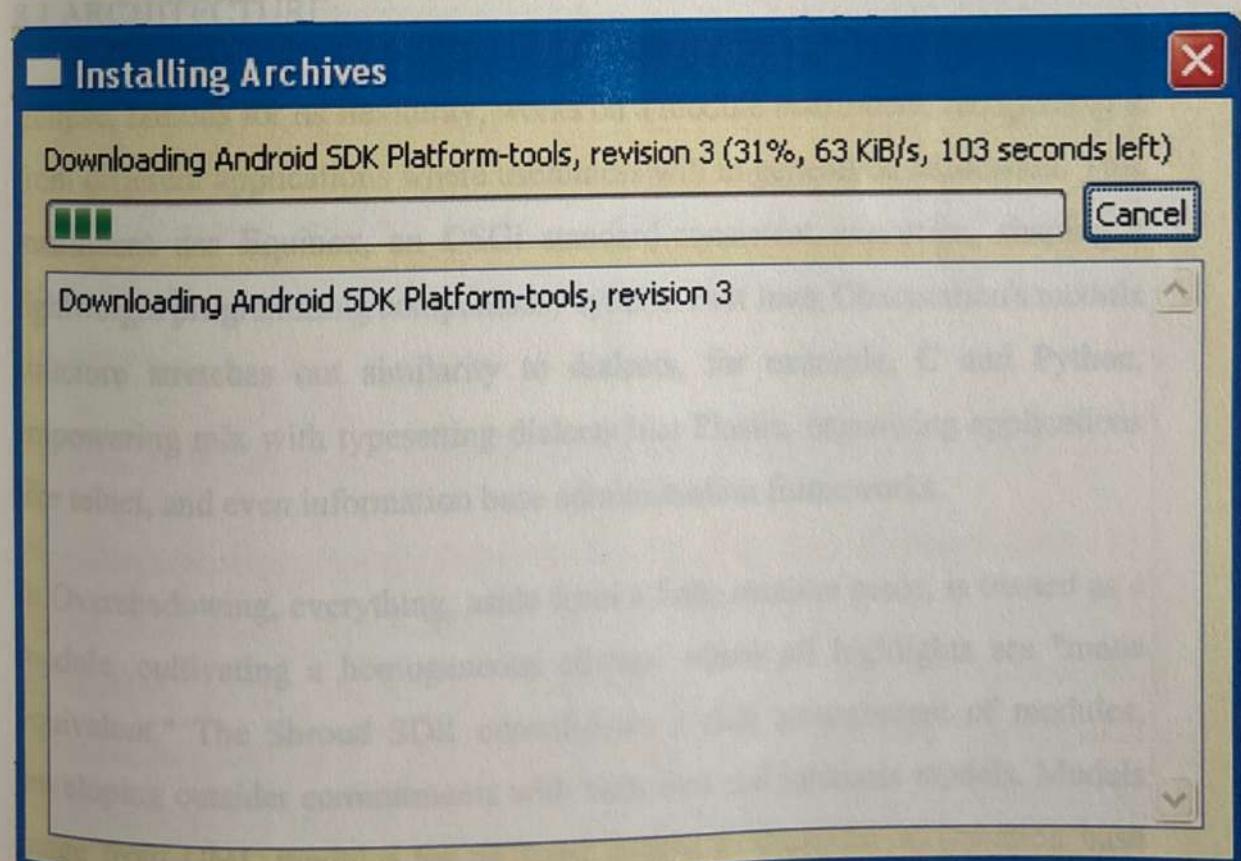
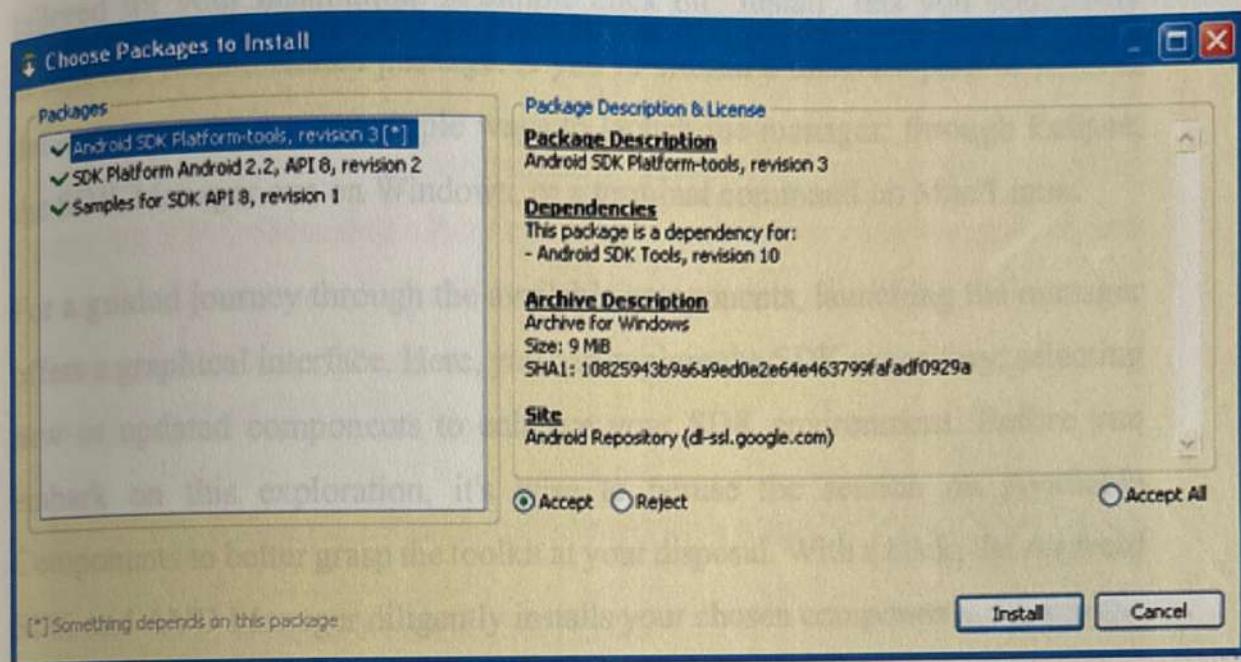
Alternative IDEs: If Eclipse doesn't align with your preferences, fear not! The Android SDK tools empower you to build and debug applications directly. The "Introduction to Android Application Development" provides insights into this alternative route. Nimbark on your Android development odyssey with the enhanced capabilities of Eclipse and ADT or explore the SDK tools directly in your preferred IDE.



Step 4. Adding Platforms and Other Components

Now that you've set up the initial groundwork for your Android SDK, let's dive into the crucial finale using the Android SDK and AVD Manager. This nifty tool, bundled within the SDK starter package, is your gateway to enriching your development environment. The SDK adopts a modular structure, neatly compartmentalizing Android platform versions, add-ons, tools, samples, and documentation into installable components. While the starter package covers the essential SDK Tools, crafting your Android masterpiece requires additional

components. Harness the power of the manager to download indispensable Android platforms and associated tools, and don't shy away from exploring other enriching components for a comprehensive development experience.



Now, let's handle the finishing touches with the Android SDK and AVD Manager. If you went the Windows route and used the installer, the wizard takes the reins, launching the manager with a default set of platforms and components tailored for your installation. A simple click on "Install" lets you seamlessly accept this recommended package. If you've chosen a different path or wish to delve deeper, there are multiple ways to launch the manager: through Eclipse, the SDK Manager.exe on Windows, or a terminal command on Mac/Linux.

For a guided journey through the available components, launching the manager offers a graphical interface. Here, you can explore the SDK repository, selecting new or updated components to enhance your SDK environment. Before you embark on this exploration, it's wise to peruse the section on Available Components to better grasp the toolkit at your disposal. With a click, the Android SDK and AVD Manager diligently installs your chosen components, paving the

8.1 ARCHITECTURE:

Eclipse, famous for its flexibility, works on a module instrument, recognizing it from different applications where usefulness will in general be hardcoded. This instrument uses Equinox, an OSGi standard-consistent execution, shaping a lightweight programming component system. Past Java, Obscuration's module structure stretches out similarity to dialects, for example, C and Python, empowering mix with typesetting dialects like Plastic, organizing applications like telnet, and even information base administration frameworks.

In Overshadowing, everything, aside from a little runtime piece, is treated as a module, cultivating a homogeneous climate where all highlights are "made equivalent." The Shroud SDK consolidates a rich arrangement of modules, enveloping outsider commitments with both free and business models. Models range from UML modules for different graphs to those for information base

investigation. The center Shroud SDK incorporates the Java Improvement Devices (JDT), highlighting a Coordinated Improvement Climate (IDE) with an inherent gradual Java compiler and an extensive Java source document model. This works with cutting edge refactoring and code examination, upgrading the advancement cycle.

Overshadowing's exceptional methodology stretches out to its gadget tool stash, known as SWT (Standard Gadget Tool compartment), digressing from the typical Java standard Unique Window Tool stash (AWT) or Swing. The transitional GUI layer, JFace, improves on application development in light of SWT, giving adaptability in UI advancement.

Obscuration's design consolidates key components inside its Rich Client Stage: Equinox OSGi: A standard packaging system.

Center stage: Answerable for booting Obscuration and running modules.

SWT: A compact gadget tool stash.

JFace: Presents watcher classes, helping with model-view-regulator programming for SWT, document cradles, message taking care of, and word processors.

Overshadow Workbench: Oversees sees, editors, points of view, and wizards.

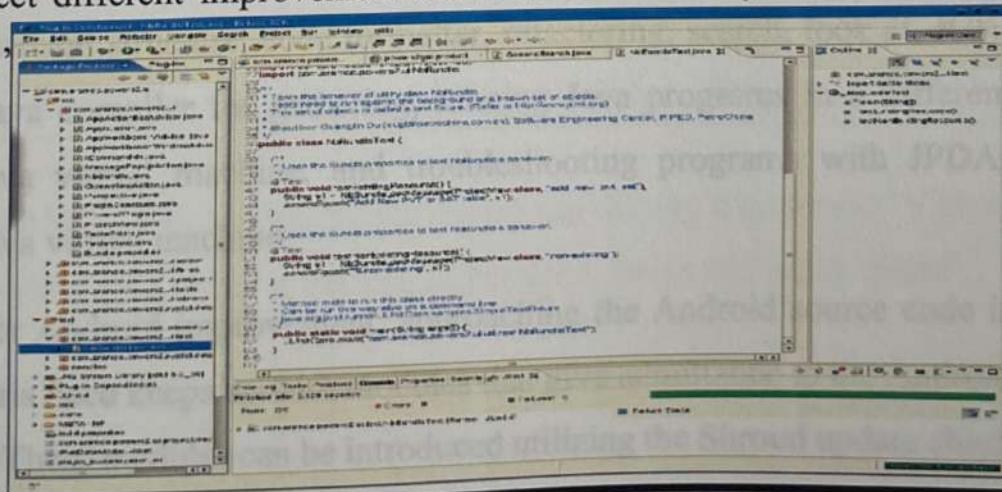
Obscuration's authentic roots follow back to an IBM Canada project started by Article Innovation Worldwide (OTI). At first planned as a Java-based replacement to the Visual Age IDE family, which was Smalltalk-based, Obscuration developed into an open-source drive. The Obscuration Establishment was laid out in 2004 to drive its turn of events. Adaptation 3.0,

delivered in 2004, took on OSGi Administration Stage determinations as the runtime engineering.

Overshadow, at first under the Normal Public Permit, changed to the Obscuration Public Permit, both recognized as free programming licenses. Regardless of inconsistency with the GNU Overall population Permit (GPL), Obscuration gathered inescapable reception. The name "Obscuration" decisively designated Microsoft's Visual Studio, lining up with the product's central goal to overshadow its rivals.

Obscure SDK, the Java-based open-source IDE, amalgamates different Shroud projects, including Stage, Java Advancement Apparatuses (JDT), and the Module Improvement Climate (PDE). Supporting a huge number of programming dialects, Shroud has turned into a staple in both open-source and business programming improvement. It offers highlights like Java altering with approval, steady assemblage, cross-referring to, code help, an XML Proofreader, Mylyn, and the sky is the limit from there. Delivered under the Obscuration Establishment, Overshadowing's financially well-disposed permit urges coordinated effort and commitment to the local area.

Fundamentally, Overshadowing rises above its way of life as a Java IDE, encapsulating a dynamic, module driven biological system that adjusts and grows to meet different improvement needs across a variety of dialects and applications.



Eclipse Platform

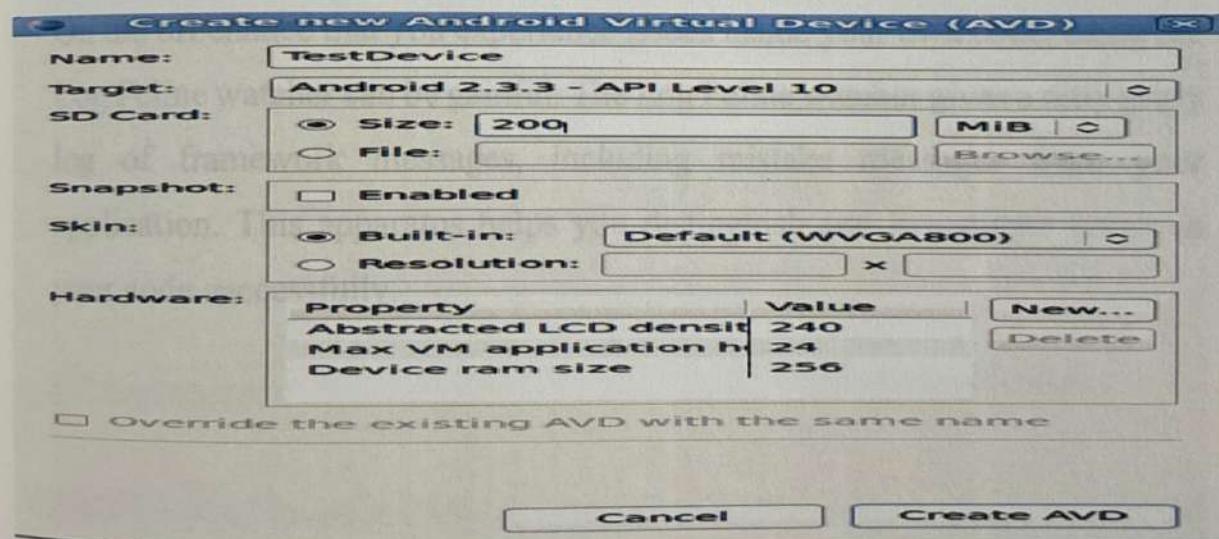
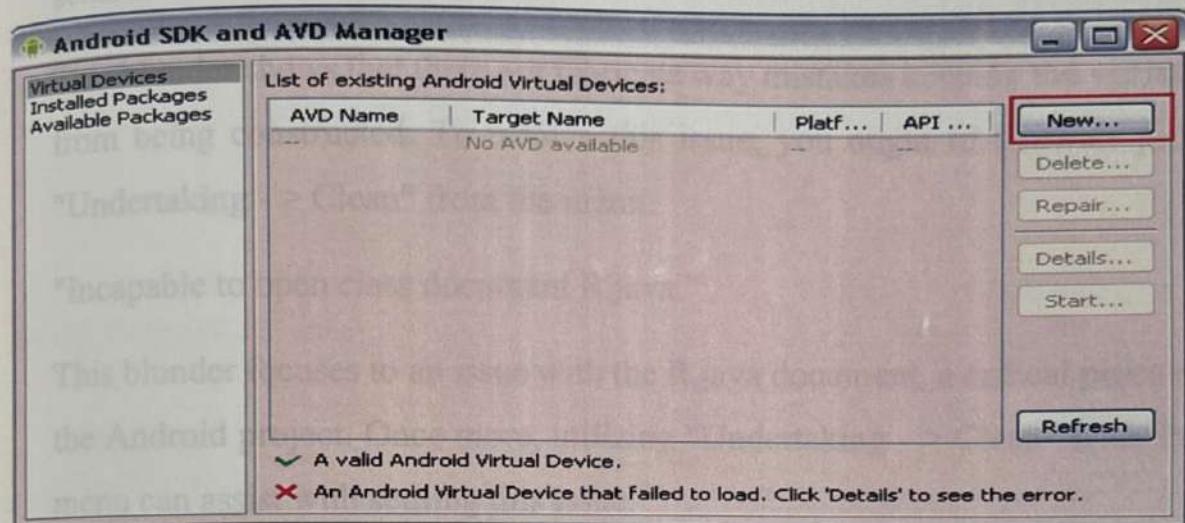


The Eclipse Platform Stage fills in as the basic structure for all module augmentations, giving the runtime climate to stacking, coordinating, and executing modules. Its essential objective is to engage instrument engineers to develop and convey coordinated devices, supporting a different scope of utilizations easily. Key elements of the Overshadowing Stage incorporate the capacity to build instruments for different application improvement errands, support for apparatus suppliers like free programming merchants (ISVs), control of inconsistent substance types (HTML, Java, C, JSP, EJB, XML, GIF), consistent joining of instruments across happy kinds and suppliers, and backing for both GUI and non-GUI-based advancement conditions.

The Java Improvement Apparatuses (JDT) project inside Shroud centre's around executing apparatus modules for Java Incorporated Advancement Climate (IDE). It upholds the improvement of any Java application, including Shroud modules. The JDT project presents a Java project nature and viewpoint to the Overshadowing Workbench, alongside different perspectives, editors, wizards, manufacturers, and code combining and refactoring instruments. Highlights of the JDT project incorporate Java projects with source records coordinated in bundle registries, altering with catchphrase and punctuation shading, frame showing statement structure, code formatter, refactoring, search, look at, JCK-consistent Java compiler for gathering, running Java programs in a different objective Java virtual machine, and troubleshooting programs with JPDA-consistent Java virtual machine.

Moreover, for Android improvement, approaching the Android source code is valuable. Haris Poco keeps up with modules that give admittance to the Android source code. These modules can be introduced utilizing the Shroud update chief.

One more fundamental stage in Android improvement is making an Android Emulator Gadget. The Android instruments incorporate an emulator that recreates a genuine Android gadget, permitting engineers to test applications without an actual gadget. Imitated gadgets are arranged utilizing "Android Virtual Gadget" (AVD), and the gadget administrator in Obscuration works with the production of AVDs with explicit setups.



Press "Create AVD". This will create the device and display it under the "Virtual devices". To test if your setup is correct, select your device and press "Start".

While working with Android improvement, experiencing blunders is a typical test. Clients frequently face issues, for example,

"Project ... is missing required source organizer: 'gen'"

This blunder recommends that the 'gen' source organizer isn't as expected arranged. To determine this, you can choose "Venture -> Clean" from the menu.

"The undertaking couldn't be worked until assemble way blunders are settled."

This blunder shows that there are fabricate way mistakes keeping the venture from being constructed. To resolve this issue, you ought to likewise pick "Undertaking -> Clean" from the menu.

"Incapable to open class document R.java."

This blunder focuses to an issue with the R.java document, a critical piece of the Android project. Once more, utilizing "Undertaking -> Clean" from the menu can assist with settling this issue.

On the off chance that you experience issues inside your own code, using the Log Feline watcher can be gainful. The Log Feline watcher gives a nitty gritty log of framework messages, including mistake messages from your application. This apparatus helps you distinguish and investigate issues in your code successfully.

CHAPTER-9

OUTCOMES

Architecture

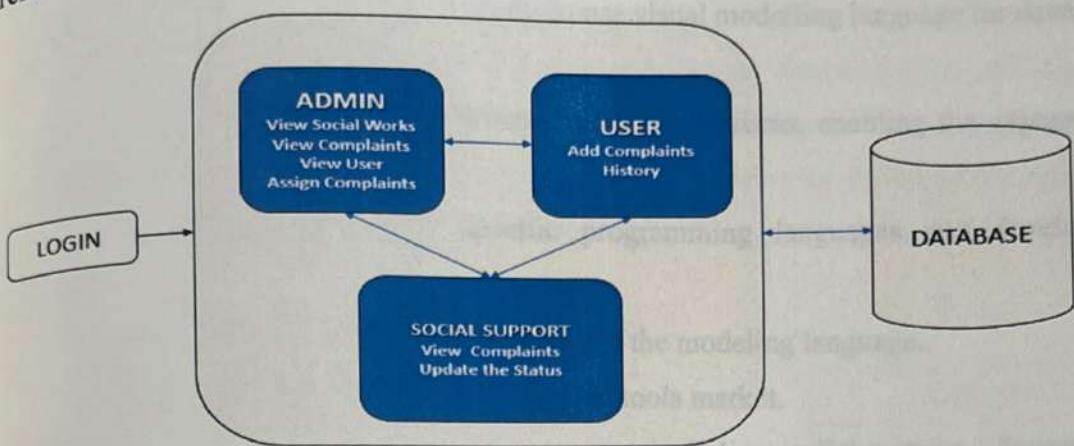


Figure 8.1 Architecture

Block Diagram:

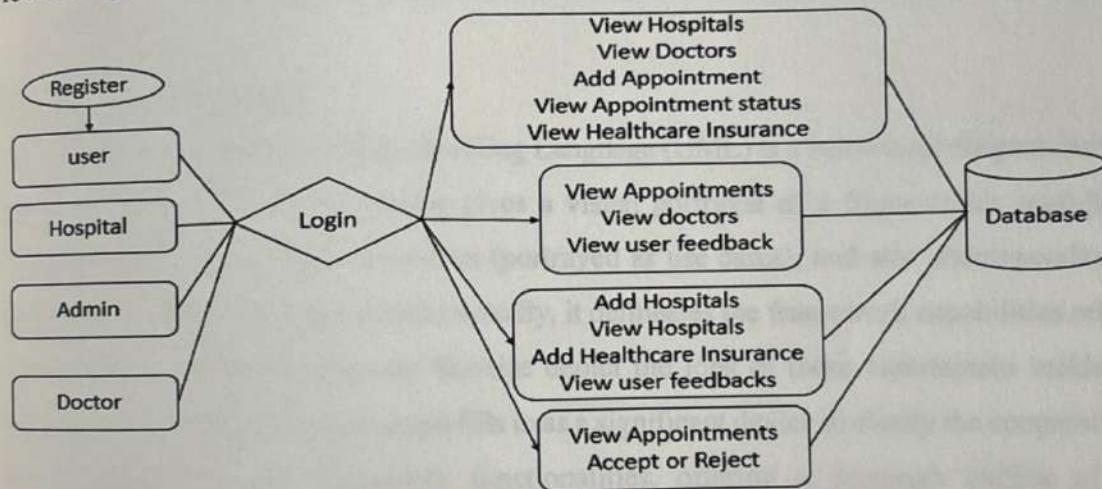


Figure 8.2 Block Diagram

UML DIAGRAMS

Unified Modelling Language (UML) is a normalized displaying language in object-situated programming, supervised by the Item The executives Gathering. With an objective to be a widespread language for making object-situated programming models, UML includes a Meta-model and documentation. It fills in as a norm for determining, picturing, building, and reporting programming framework relics, stretching out its material ness to business and non-programming frameworks. UML epitomizes effective designing practices for demonstrating huge and complicated frameworks, assuming a critical part in object-situated programming

improvement through graphical documentations that express programming project plans.

GOALS:

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

1. Furnish users with an intuitive, ready-to-use visual modelling language for developing and sharing meaningful models.
2. Incorporate extendibility and specialization mechanisms, enabling the expansion of core concepts.
3. Ensure independence from specific programming languages and development processes.
4. Establish a formal basis for comprehending the modeling language.
5. Foster the expansion of the object-oriented tools market.
6. Support advanced development concepts, including collaborations, frameworks, patterns, and components.
7. Integrate best practices to enhance the overall effectiveness of the modeling language.

USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a behavioral diagram derived from use-case analysis. This outline gives a visual portrayal of a framework's usefulness, displaying entertainers, their objectives (portrayed as use cases), and any interdependencies among these utilization cases. Fundamentally, it delineates the framework capabilities related with explicit entertainers and can likewise depict the jobs of these entertainers inside the framework. The utilization case graph fills in as a significant device to clarify the cooperation's among entertainers and framework functionalities, offering a thorough outline of the framework's way of behaving.

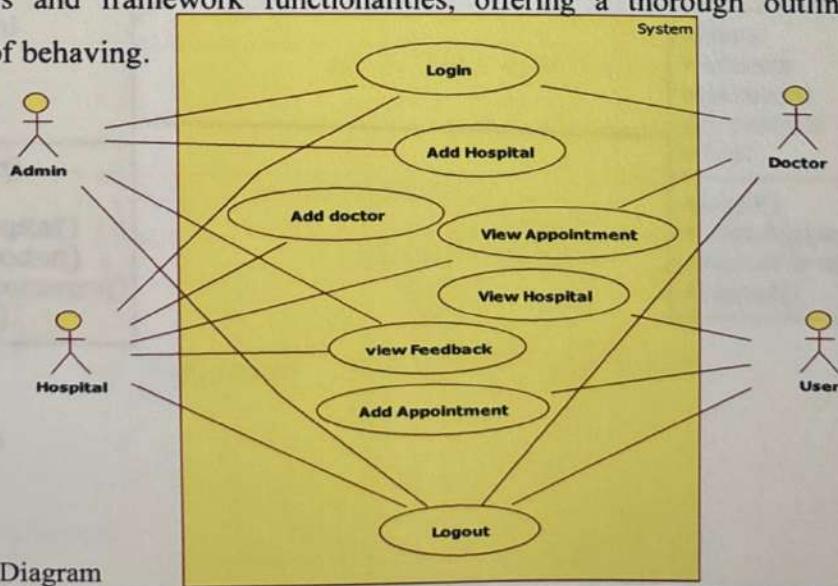


Figure 8.2 Use Case Diagram

CLASS DIAGRAM:

In programming, a class graph inside the Bound together Displaying Language (UML) fills in as a static construction outline, depicting a framework's design. This chart outwardly addresses the framework's classes, including their qualities, tasks (or strategies), and the connections existing among them. Basically, it gives bits of knowledge into how data is exemplified inside each class, offering a thorough comprehension of the framework's inward design and the collaborations between various parts. The class outline is an essential device in UML, working with a reasonable portrayal of the structure blocks and associations inside a product framework.

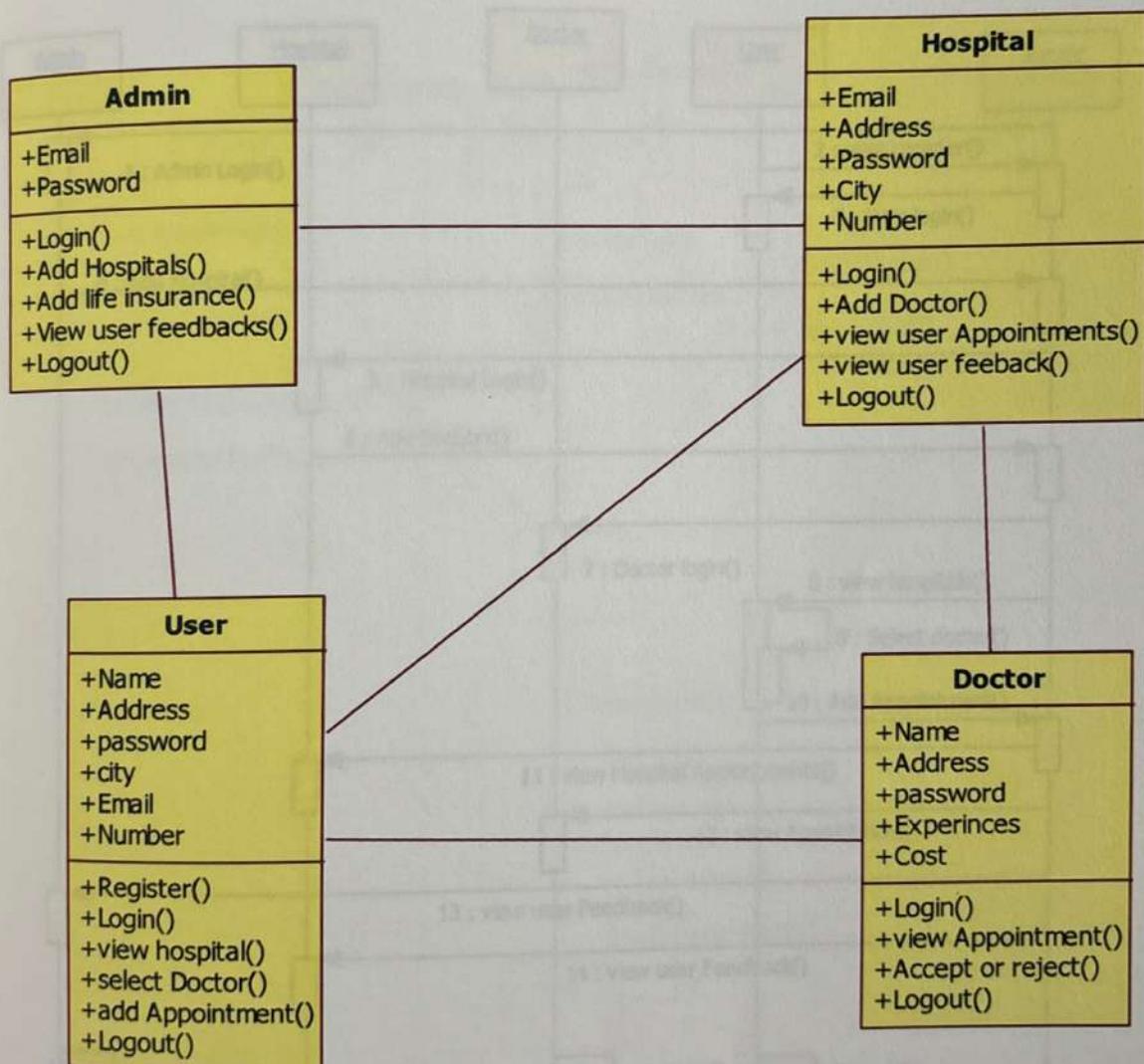


Figure 8.3 Class Diagram

SEQUENCE DIAGRAM:

In the domain of Bound together Displaying Language (UML) in programming, a succession graph stands apart as a connection outline. This chart clearly represents the stream and request of cycles as they communicate. Starting from a Message Succession Diagram, it gives a powerful portrayal of occasions and their grouping. Frequently alluded to as occasion graphs, occasion situations, or timing outlines, grouping charts assume a crucial part in depicting the sequential and intelligent parts of cycles, adding to a complete comprehension of the framework's way of behaving and correspondence stream.

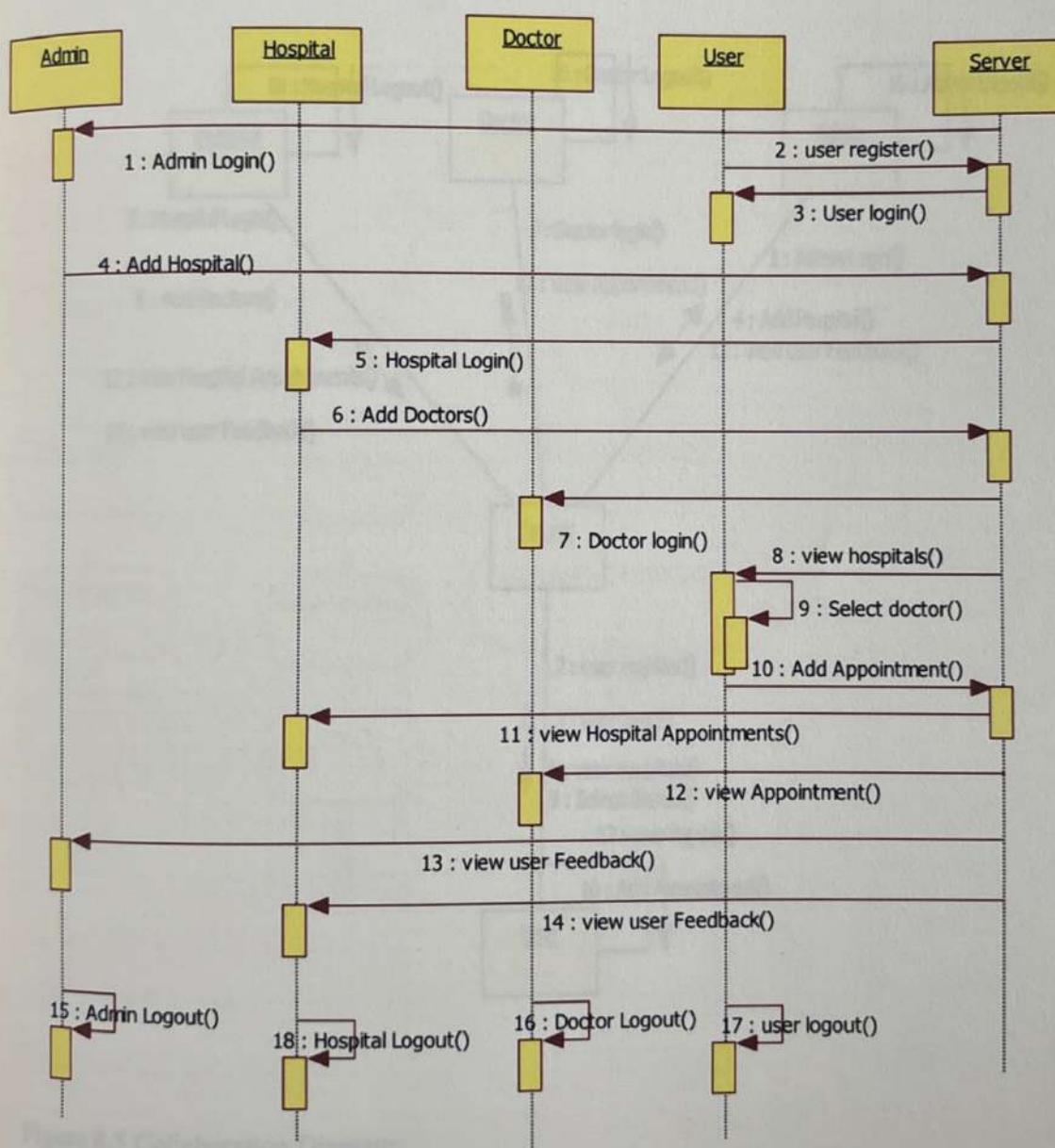


Figure 8.4 Sequence Diagram

COLLABORATION DIAGRAM:

In a collaboration diagram, the method call sequence is delineated using a numbering technique, signifying the order in which methods are invoked. For instance, in the context of an order management system, we follow a similar approach to describe the collaboration diagram. Although the method calls share similarities with a sequence diagram, the key distinction lies in their representation of object organization. Unlike sequence diagrams, collaboration diagrams explicitly depict the arrangement of objects, offering a visual insight into how different elements collaborate in a system, contributing to a more holistic understanding of the interaction dynamics.

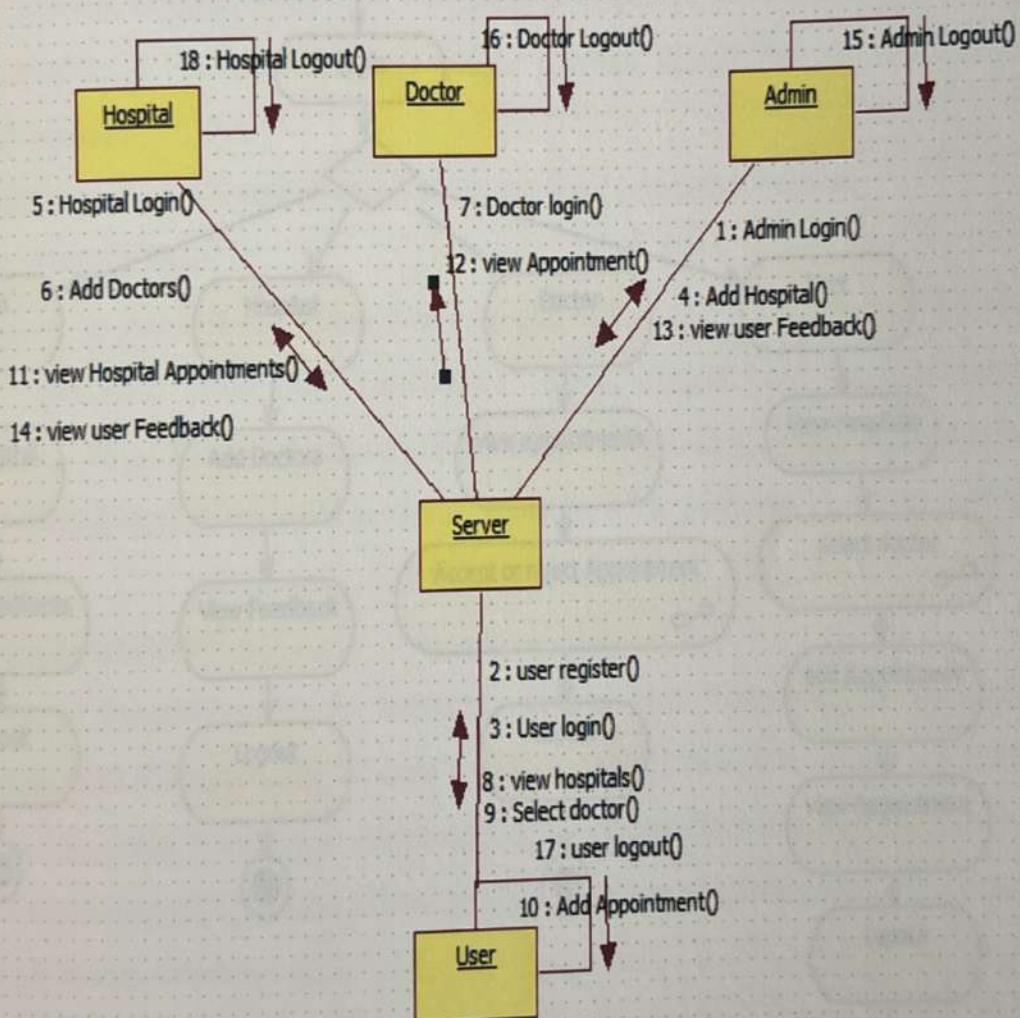


Figure 8.5 Collaboration Diagram

ACTIVITY DIAGRAM:

Action outlines act as graphical portrayals of stepwise work processes, representing exercises, activities, and supporting components like decision, emphasis, and simultaneousness. Inside the Bound together Displaying Language (UML), these graphs actually catch business and functional work processes for framework parts. They give a visual story of bit-by-bit processes, exhibiting the progression of control inside a framework. Action graphs assume a critical part in upgrading understanding by offering a reasonable and thorough portrayal of how various parts communicate and advance through different exercises inside a framework.

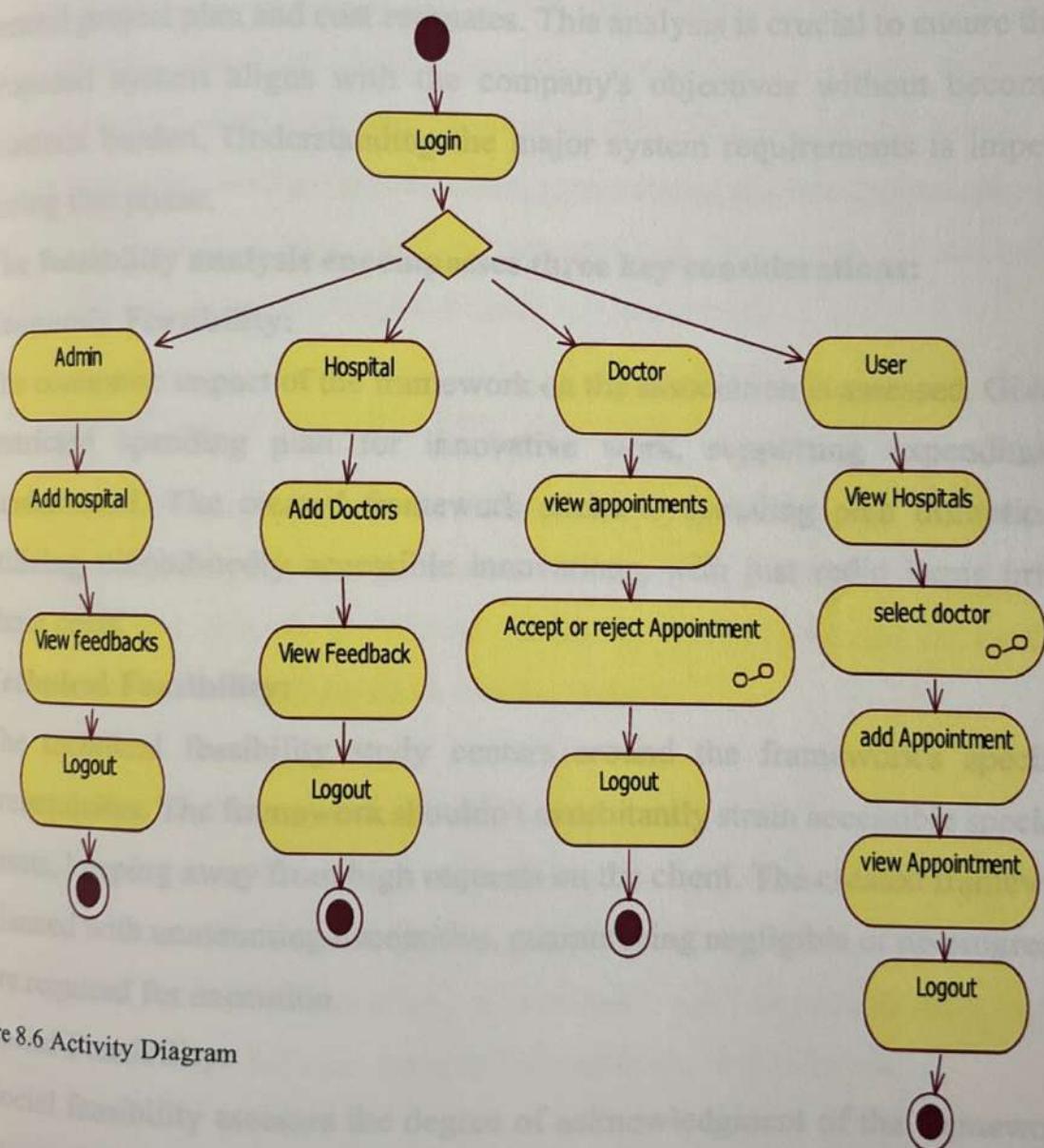


Figure 8.6 Activity Diagram

CHAPTER-10

RESULTS AND DISCUSSIONS

SYSTEM STUDY

9.1 FEASIBILITY STUDY

In the initial phase of system development, a thorough analysis is conducted to assess the project's feasibility and present a business proposal along with a general project plan and cost estimates. This analysis is crucial to ensure that the proposed system aligns with the company's objectives without becoming a financial burden. Understanding the major system requirements is imperative during this phase.

The feasibility analysis encompasses three key considerations:

Economic Feasibility:

The economic impact of the framework on the association is assessed. Given the restricted spending plan for innovative work, supporting expenditures is fundamental. The created framework sticks to spending plan limitations by utilizing uninhibitedly accessible innovations, with just redid items bringing about costs.

Technical Feasibility:

The technical feasibility study centers around the framework's specialized prerequisites. The framework shouldn't exorbitantly strain accessible specialized assets, keeping away from high requests on the client. The created framework is planned with unassuming necessities, guaranteeing negligible or no progressions are required for execution.

Social Feasibility:

Social feasibility assesses the degree of acknowledgment of the framework by clients. Client preparing assumes a critical part, guaranteeing clients feel great and see the framework as a need instead of a danger. Client acknowledgment

relies upon viable training strategies, ingraining certainty, and empowering valuable analysis from clients who are at last the framework's end-clients.

Moving to the framework testing stage, the essential objective is to find mistakes and guarantee the product meets its necessities and client assumptions. Testing includes investigating each possible shortcoming or shortcoming in the work item, going from parts to the completed item. It is the most common way of practicing the product to forestall unsatisfactory disappointments. Various kinds of tests address explicit testing necessities, guaranteeing complete coverage. The early venture stages accentuate intensive investigation, taking into account financial, specialized, and social achievability. The framework testing stage means to uncover and amend blunders, guaranteeing the last programming item lines up with prerequisites and client assumptions. This organized methodology adds to fruitful framework improvement and execution.

Healthcare Informatics fosters improved communication and collaboration.

9.2 IMPORTANCE OF HEALTHCARE INFORMATICS

Healthcare informatics plays a pivotal role in transforming the healthcare landscape, bringing about significant improvements in patient care, operational efficiency, and overall healthcare outcomes. The importance of healthcare informatics can be highlighted in various aspects:

5. Streamlined Administrative Processes

1. Efficient Data Management:

Healthcare informatics enables the efficient management of vast amounts of healthcare data. This includes patient records, treatment plans, medical histories, and administrative information. Centralized and organized data systems contribute to better decision-making by healthcare professionals.

6. Data Analysis for Public Health

2. Clinical Decision Support:

Informatics tools provide healthcare professionals with real-time access to

relevant and comprehensive patient data. This assists in clinical decision-making by offering insights into treatment options, drug interactions, and evidence-based practices. Clinical decision support systems help enhance the accuracy and effectiveness of medical decisions.

3. Improved Patient Care and Safety:

By facilitating quick access to patient information, healthcare informatics ensures that healthcare providers have a complete and up-to-date view of a patient's health status. This leads to more accurate diagnoses, timely interventions, and improved patient safety. Electronic health records (EHRs) contribute to the continuity of care and reduce the risk of medical errors.

4. Enhanced Communication and Collaboration:

Healthcare informatics fosters improved communication and collaboration among healthcare professionals, departments, and institutions. Electronic communication systems and shared databases enable seamless information exchange, ensuring that all stakeholders are well-informed and coordinated in their efforts.

5. Streamlined Administrative Processes:

Administrative tasks, such as appointment scheduling, billing, and insurance management, are streamlined through healthcare informatics. Automated processes reduce paperwork, minimize errors, and optimize resource allocation. This leads to increased operational efficiency and cost savings.

6. Data Analytics for Public Health:

Informatics tools allow for the analysis of healthcare data on a larger scale, contributing to public health initiatives. By identifying patterns and trends, healthcare informatics aids in disease surveillance, outbreak detection, and the

development of preventive strategies. This proactive approach is instrumental in managing and controlling public health challenges.

7. Research and Innovation:

Healthcare informatics supports medical research by providing a wealth of data for analysis. Researchers can use informatics tools to identify patterns, conduct clinical trials, and explore innovative approaches to healthcare. This accelerates the pace of medical discovery and the development of new treatment modalities.

8. Patient Empowerment:

Informatics empowers patients by providing them with access to their health information, appointment scheduling tools, and educational resources. This promotes patient engagement, encourages self-management of health, and fosters a collaborative relationship between patients and healthcare providers.

CHAPTER-11

SYSTEM STUDY AND TESTING

Types of test & Test Cases

Unit Testing:

Unit testing includes planning experiments to approve that the inside program rationale works accurately, guaranteeing that program inputs produce legitimate results. This testing is finished at the singular programming unit level, post the consummation of a singular unit and before joining. A primary and obtrusive testing strategy approves choice branches and inward code stream. Unit tests perform fundamental tests at the part level, zeroing in on unambiguous business cycles, applications, or framework designs. The point is to guarantee every exceptional way of a business interaction sticks to reported particulars, with clear cut inputs and anticipated results.

Integration Testing:

Integration testing assesses coordinated programming parts to decide whether they run flawlessly as a solitary program. Occasion driven testing underscores essential results of screens or fields. While unit testing guarantees individual parts capability sufficiently, incorporation testing uncovered issues emerging from their mix. It checks that the parts, however separately fruitful, work accurately and reliably together.

Functional Test:

Functional tests methodically exhibit that capabilities tried are accessible as indicated by business and specialized prerequisites, framework documentation, and client manuals. Zeroed in on legitimate and invalid data sources, distinguished capabilities, and framework yields, these tests likewise consider

communicating frameworks or methodology. The association and arrangement of practical tests focus on prerequisites, key capabilities, or unique experiments.

System Test:

System testing guarantees the whole coordinated programming framework meets prerequisites, testing a design to guarantee known and unsurprising outcomes. It includes arrangement situated framework reconciliation tests in light of cycle portrayals and streams, accentuating pre-driven process connections and coordination focuses.

White Box Testing:

White Box Testing includes testing with information on the product's internal functions, construction, and language. It targets regions inaccessible from a black box level, digging into the product's inward systems.

Black Box Testing:

Black Box Testing tests the product without information on its inward activities, design, or language. It regards the product as a "black box," zeroing in on data sources and results disregarding the interior operations. Black box tests should be composed from a conclusive source report, for example, a determination or necessities record.

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

TESTING CASES

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 Passwor d 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 and password. Click on login button 	Username, password	Username password d	When the user submits the user data, data should be authenticated successfully	As Expected,	Pass

Table 9.1 Test Cases

CHAPTER-12

CONCLUSION

In the domain of medical services, stressing care as a crucial technique holds importance from both financial and social viewpoints. Financially, focusing on preventive consideration is vital to evade hospitalization costs. Socially, it assumes a vital part in saving patient prosperity by guaranteeing progressing treatment and an excellent of care. However, the developing scene of versatile consideration administrations presents difficulties. Giving assets equivalent to those tracked down in emergency clinics, enveloping computational capacities, programming assets, and concentrated consideration, becomes principal. One outstanding element in this setting is the client's capacity to plan meetings with specialists in view of the doctor's accessibility. This engages clients to assume responsibility for their wellbeing by proactively looking for clinical consideration when required. The unique idea of portable consideration administrations, notwithstanding, requires cautious thought of asset allotment, guaranteeing that both computational and programming components line up with the principles set by customary medical care establishments. Also, keeping a customized and mindful methodology in the midst of the virtual idea of these administrations is a basic test that should be tended to. Moreover, maintaining a personalized and attentive approach amid the virtual nature of these services is a critical challenge that needs to be addressed.

CHAPTER-13

FUTURE ENHANCEMENT

The future enhancements in mobile care services hold promising possibilities for a more seamless and effective healthcare experience. One potential avenue for improvement lies in the integration of advanced artificial intelligence (AI) algorithms. Implementing AI can enhance diagnostic capabilities, streamline administrative processes, and provide personalized health recommendations based on individual patient data. Additionally, the incorporation of telemedicine technologies can further revolutionize the delivery of care. Real-time video consultations and remote monitoring tools can bridge geographical gaps, ensuring that patients, regardless of their location, have access to timely medical advice and intervention. This not only expands the reach of healthcare services but also promotes continuity of care.

REFERENCES

1. Md. Nasfikur R. Khan; A K E H Mashuk; Whomaira F Durdana; Mehdi Alam; Robin Roy, "A Customizable Android Application for Integrated Health Care", 2019
2. Christopher Kulanga; Eric Saforo; Steve Ollis; Marc Mitchell, "Integrating mHealth application", 2016
3. Mohammed Alkhawlani; Wesam Ali Husien; Saba Noori Alhamdany, "Facilitating Patient Registrations Using an Integrating Healthcare Management System", 2018
4. A. Hylton and S. S. Arayanan, "Application of Intelligent Agents in Hospital Appointment Scheduling System", 2012.
5. D. Ameta, K. Mudaliar and P. P., "Medication Reminder And Healthcare - An Android Application", 2015.
6. Y. Symeay and S. N. B. Sait, "Application of Smart Technologies for Mobile Patient Appointment System" 2013.
7. A. L., A. B., L. Marzi, F. Frosini, R. Miniati and E. Iadanza, "Careggi Smart Hospital: a mobile app for patients citizens and healthcare staff", 2014
8. S. B. Choudhari, Chaitanya Kusurkar, Rucha Sonje, Parag Mahajan and Joanna Vaz, "Android Application for Doctor's Appointment", 2014.
9. S. Gavaskar, A. Sumithra and A. Saranya, "Health Portal-An Android Smarter Healthcare Application", 2013
10. E. Fife and F. Pereira, "Digital home health and mHealth: Prospects and challenges for adoption in the U.S", 2011.
11. K.Nishanthan, & S.Mathiyvathana, & R.Priyanthi, & A.Thusara, & Silva, D.I. & Cooray, Dulanji. (2022). The Hospital Management System. International Journal of Engineering and Management Research. 12. 135-149. 10.31033/ijemr.12.5.17.
12. An Enhanced Entity Model for Converting Relational to Non-Relational Documents in Hospital Management System Based on Cloud Computing
13. IETE Technical Review

APPENDIX-A

SCREENSHOTS

Start Page



Login Page



LOGIN

Email

Password

LOGIN

NOT YET REGISTERED?

DOCTOR LOGIN



Application Icon

(Android)



Integrated Health Care Application

ADMIN PORTAL (Login email: admin@gmail.com // Password : admin)



USER FEEDBACKS

ADD HEALTH CARE INSURANCE

LOGOUT

HOME ADD HOSPITAL VIEW HOSPITALS

USER FEEDBACK

Hospital Email: sang@gmail.com
Feedback: No feedback

Appointment: MED4332
Hospital Email: sai@gmail.com
Feedback: No feedback

Appointment: MED4332
Hospital Email: sai@gmail.com
Feedback: No feedback

Appointment: MED5888
Hospital Email: sai@gmail.com
Feedback: No feedback

Appointment: MED4937
Hospital Email: hospital@gmail.com
Feedback: nice service

Appointment: MED1814
Hospital Email: nikhilhosp@gmail.com
Feedback: ok

Appointment: MED7716
Hospital Email: nikhilhosp@gmail.com
Feedback: No feedback

Appointment: MED7580
Hospital Email: apollo@gmail.com
Feedback: No feedback

Add Health care Insurance

PRESIDENCY INSURANCE

92848282838

Itgalpur Rajanakunte, Yelahanka, Bengaluru,
Ittagallpura, Karnataka 560064, India

full family insurance

ADD

Name: Insurance
Number: 1234567890
Address: Chennai

Description
health life insurance

Name: Star Insurance
Number: 8754216020
Address: C 23 double road

ADD HOSPITAL

Enter Hospital Name:
Apollo

Enter Hospital Number:
9856235854

Enter Hospital Address:
Hebbal

Enter Hospital city/(Eg Rayapuram):
Hebbal

Enter Hospital Services:
24/7 service

Enter Hospital Email:
apollo@gmail.com

Enter Hospital Password:
apollo

ADD

successful!

HOSPITAL

Number of Hospitals->3

Hospital 1

Name: hospital
Email: hospital@gmail.com
Number: 1234567890
city: Tirupati
service: all types medical needs

Hospital 2

Name: hello
Email: nikhilhosp@gmail.com
Number: 0805288555
city: Rajankunte
service: heart specialist

Hospital 3

Name: Apollo
Email: apollo@gmail.com
Number: 9856235854
city: Hebbal
service: 24/7 service

Success

LOGOUT

Do you Want to Logout?

NO YES

LOGOUT

HOSPITAL PORTAL (LOGIN WITH EMAIL AND PASSWORD)

LOGIN

Email: manipal@gmail.com
Password: manipal

DOCTORS

Add Doctor Name: Kitty
Enter Doctor Number: 9741400373
Enter Doctor Address: hebbal
Enter Doctor Specialization: mbbs
Enter Doctor Experience: 14
Enter Doctor Case: 650
Enter Doctor Password: kitty

USER APPOINTS

USER FEEDBACKS

DOCTOR LOGIN

ADD DOCTORS

NOT YET REGISTERED?

SHOW DOCTORS

DOCTOR

Number of Doctor->1

Name: Lakshmi
Number: 9964106139
Specialization: Md
Experience: 1

APPOINTMENT

Number of Appointments->1

Appointment: MED1506
Doctor: Lakshmi
Time: 2023-12-22,5:33
status: Completed

HOSPITAL PROFILE

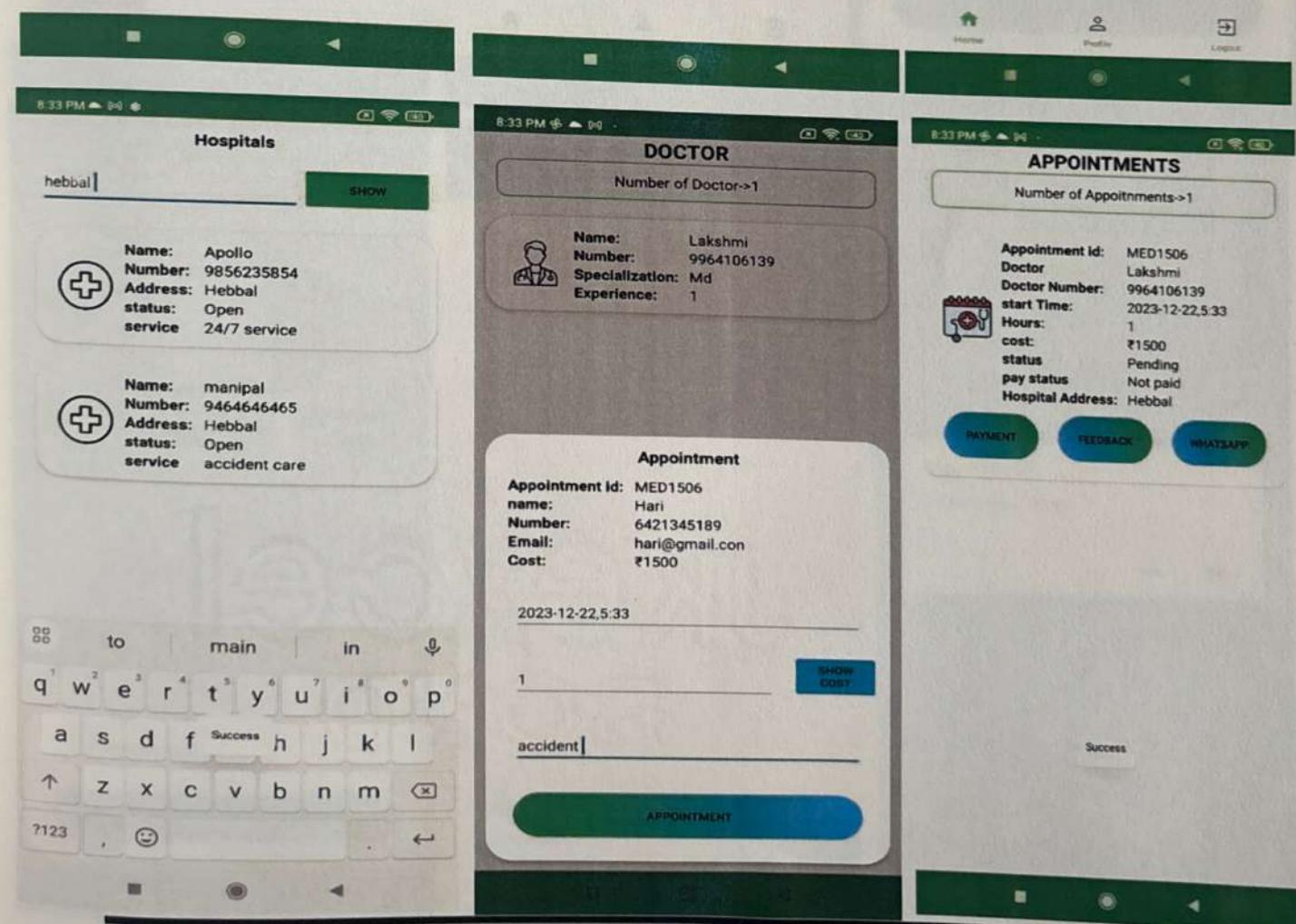
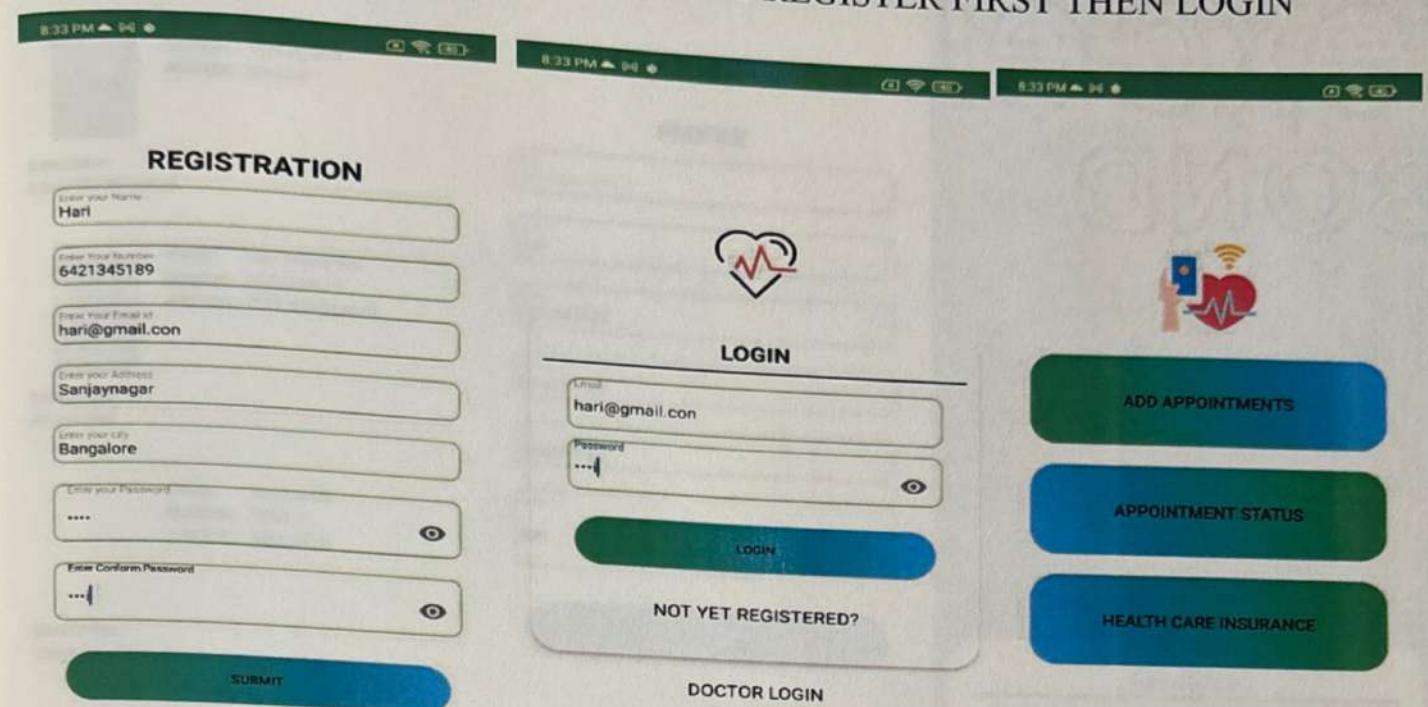
Hospital Email: manipal@gmail.com
Hospital Name: manipal
Hospital Number: 9464646465
Hospital Services: accident care
Hospital Address: Hebbal
Hospital city: hebbal
Hospital Password: manipal

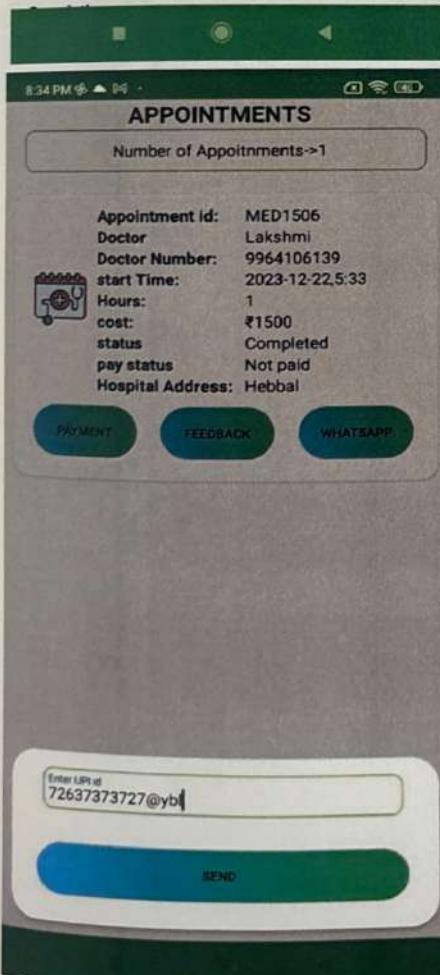
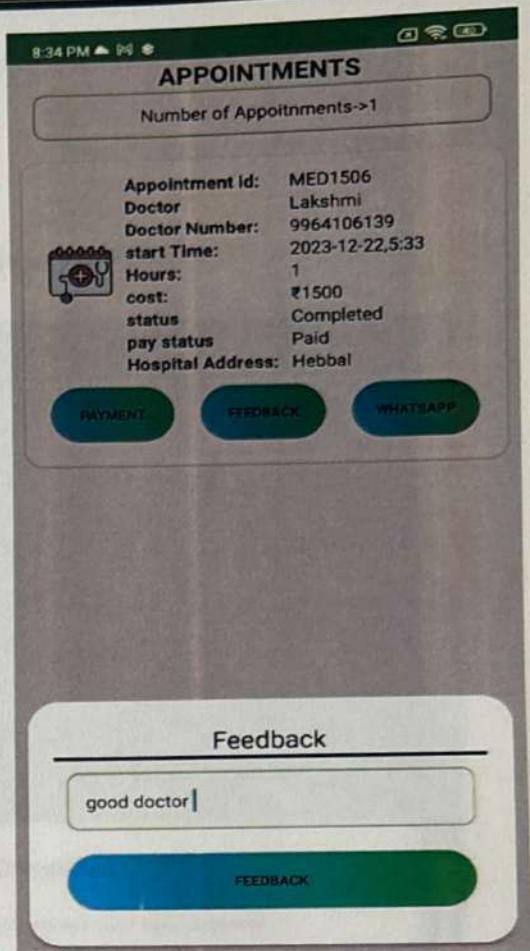
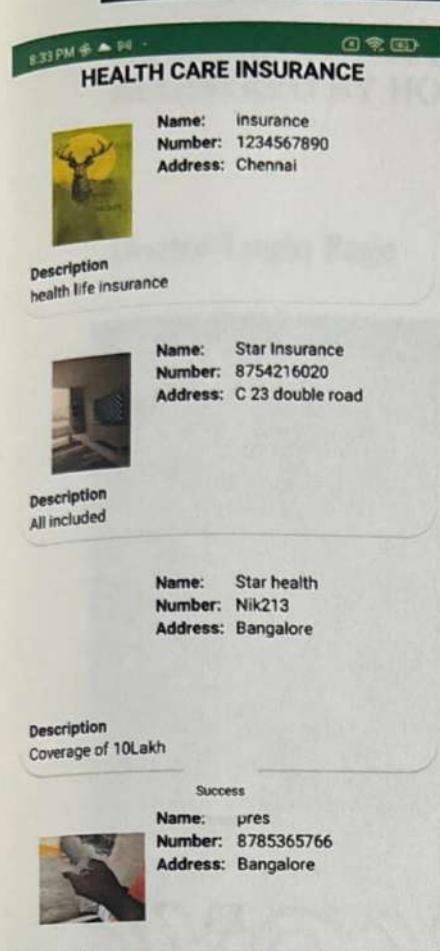
status

Open

Close

USER / PATIENT PORTAL NEEDS TO REGISTER FIRST THEN LOGIN



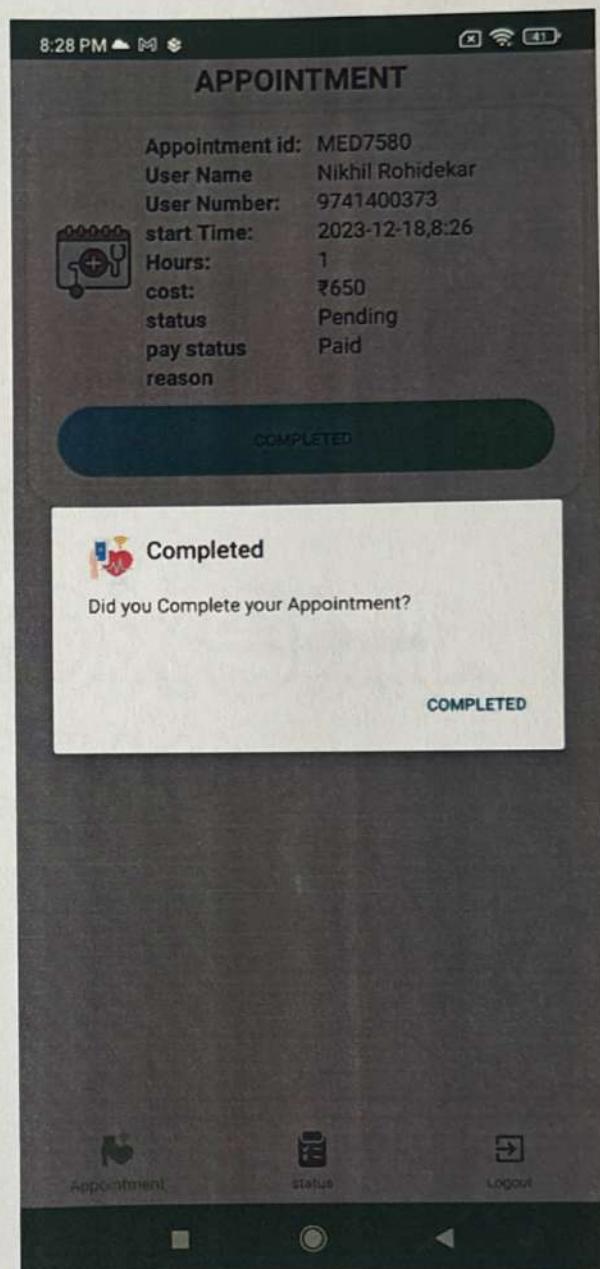


DOCTOR PORTAL (LOGIN WITH DOCTOR NAME AND PASSWORD REGISTERED BY HOSPITAL)

Doctor Login Page



Confirmation by Doctor



APPENDIX-B

ENCLOSURES

1. Conference Paper

Conference Paper Link:

https://drive.google.com/file/d/1C1c-WDyAyJfCP15n44VtJAbKJtQYc12a/view?usp=drive_link



Similarity Index / Plagiarism Check report:

Design and Implementation of an Integrated Healthcare Management System: Enhancing Administrative, User, Hospital, and Doctor Functionalities for Optimal Patient Care

ORIGINALITY REPORT

3%

SIMILARITY INDEX

2%

INTERNET SOURCES

1%

PUBLICATIONS

2%

STUDENT PAPERS

PRIMARY SOURCES

1

ssgmjournal.in

Internet Source

1 %

2

easychair.org

Internet Source

1 %

3

Submitted to Vaal University of Technology

Student Paper

1 %

4

Submitted to Australian Catholic University

Student Paper

1 %

5

Submitted to University of Bedfordshire

Student Paper

<1 %

6

Submitted to University of Illinois at Urbana-Champaign

Student Paper

<1 %

Exclude quotes Off
Exclude bibliography Off

Exclude matches Off

2. Similarity Index / Plagiarism Check Report

Integrated Health Care Application

ORIGINALITY REPORT

17%
SIMILARITY INDEX

13%
INTERNET SOURCES

5%
PUBLICATIONS

11%
STUDENT PAPERS

PRIMARY SOURCES

1	kupdf.net Internet Source	4%
2	Submitted to Sreenidhi International School Student Paper	1%
3	ijsrcseit.com Internet Source	1%
4	www.geeksforgeeks.org Internet Source	1%
5	ijsrset.com Internet Source	1%
6	ukdiss.com Internet Source	1%
7	Submitted to Manipal University Student Paper	<1%
8	repository.stcloudstate.edu Internet Source	<1%
9	Submitted to Visvesvaraya Technological University, Belagavi Student Paper	<1%