Medicine Tracker And Reminder

BY

MD. Saddam Hosen
ID: 171-15-9360
Department of CSE
Daffodil International University

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Ms. Israt Jahan
Lecturer (Senior Scale)
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

JANUARY, 2025

APPROVAL

This Project titled "**Medicine Tracker and Reminder**", submitted by MD. Saddam Hosen, ID: 171-15-9360 to the Department of Computer Science and Engineering, Daffodil International University, has been acknowledged as palatable for the halfway satisfaction of the prerequisites for the level of B.Sc. in Computer Science and Engineering and approved with regards to its style and substance. The presentation has been held on 12th January,2025.

BOARD OF EXAMINERS

Dr. Md. Taimur Ahad — **Chairman**
Associate Professor & Associate Head
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Mushfiqur Rahman — **Internal Examiner**
Assistant Professor
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Mr. Rahmatul Kabir Rasel Sarker — **Internal Examiner**
Lecturer
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Sadat Hasan — **External Examiner**
Data Scientist (Senior Principal Officer)
Department of Risk Management Division
BRAC Bank

DECLARATION

I hereby declare that, this project has been done by us under the supervision of **Ms. Israt Jahan, Lecturer (Senior Scale), Department of CSE** Daffodil International University. I likewise proclaim that neither this undertaking nor any aspect of this project has been submitted somewhere else for grant of any degree or confirmation.

Supervised by:

15.01.25

Ms. Israt Jahan
Lecturer (Senior Scale)
Department of CSE
Daffodil International University

Submitted by:

Saddam
15.01.25

MD. Saddam Hosen
ID: 171-15-9360
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project successfully.

I really grateful and wish our profound our indebtedness to **Ms. Israt Jahan, Lecturer (Senior Scale), Department of CSE** Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of Android app development to carry out this project. His unending tolerance, insightful direction, ceaseless support, steady and enthusiastic management, helpful analysis, important counsel, perusing numerous second-rate drafts and remedying them at all stage have made it conceivable to finish this project.

I would like to express my heartiest gratitude to Ms. Israt Jahan, and Dr. Sheak Rashed Haider Noori Head**,** Department of CSE, for his kind help to finish my project and also to other faculty member and the staff of CSE department of Daffodil International University.

I might want to thank our whole course mate in Daffodil International University, who participated in this talk about while finishing the course work.

At last, I should recognize with due regard the steady help and patients of my parents.

ABSTRACT

The Medicine Tracker and Reminder with Emergency SOS app is a comprehensive healthcare solution designed to help individuals manage their medication schedules and health emergencies efficiently. This Android-based app provides a seamless user experience with features like Sign Up and Login, a personalized User Profile, and a Splash Screen for an intuitive start. Users can receive timely Medicine Reminder Alarms and track their Total Medicine Inventory with ease. The app allows users to log Medicine History, monitor the Last Taken Time with dates, and receive Low Medicine Stock Notifications to ensure they never run out. Additionally, it features Speech-to-Text Input for quick medicine addition and smart suggestions for medicine names when searching.

To address critical situations, the app includes an Emergency SOS feature, enabling users to add trusted contacts directly from their phone and notify them instantly during emergencies. Java serves as the core backend programming language, while Firebase Database ensures secure storage and real-time management of user data. The Alarm Manager handles precise scheduling of reminders and notifications, ensuring medicines are taken on time. This app is an ideal tool for individuals, families, and caregivers to efficiently manage medications, track health histories, and ensure timely responses in emergency situations, offering a reliable and user-friendly healthcare management solution.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

Introduction

In today's fast-paced world, managing medication schedules and ensuring timely intake can be a significant challenge, particularly for individuals with chronic illnesses, the elderly, or those juggling multiple prescriptions. Missing doses or running out of medicine can lead to severe health complications. The Medicine Tracker and Reminder with Emergency SOS app is designed to address these challenges by providing an intuitive and reliable solution to streamline medicine management and enhance user safety during emergencies.

This Android-based application combines essential healthcare features with modern technology to offer a seamless experience. Users can sign up, log in, and create a personalized user profile to manage their medication routine efficiently. The app provides timely reminders using the Alarm Manager, ensuring medicines are taken as prescribed. It includes smart features such as speech-to-text input for easy medicine addition, suggested medicine names when searching, and automatic tracking of the last taken time and total medicine count. To ensure continuity, the app notifies users when medicines are running low, helping them restock in time. An integral feature of the app is the Emergency SOS, which allows users to quickly notify trusted contacts, automatically fetched from their phone contacts, during critical situations. This ensures timely assistance when it is most needed. With Firebase Database as the backend, the app securely stores user data, medicine logs, and emergency contacts while ensuring real-time synchronization across devices.

The Medicine Tracker and Reminder with Emergency SOS app is a comprehensive tool tailored to meet the needs of individuals and caregivers, combining reliability, accessibility, and ease of use to improve medication adherence and support health management.

1.1 Motivation

Motivation Taking medications regularly and on time is critical for managing chronic conditions and preventing disease progression. Unfortunately, it can be challenging for individuals to remember to take their medication at the right time, especially if they have multiple medications to take or complex dosing schedules. A medicine tracker and reminder can help alleviate these issues and ensure that individuals receive the full benefits of their medication.

1.2 Objectives

The primary objective of the **Medicine Tracker and Reminder with Emergency SOS** app is to provide a reliable and user-friendly solution for managing medication schedules, tracking medicine usage, and ensuring emergency support through advanced features. The app aims to:

a. **Simplify User Management:** Enable users to securely Sign Up and Login with personalized **User Profiles** for efficient medicine tracking.
b. **Enhance Medication Adherence:** Provide timely Medicine Reminder Alarms to ensure users take medicines on schedule and log the Last Taken Time with Date for better tracking.
c. **Monitor Medicine Usage:** Track the Number of Total Medicines and maintain a detailed Medicine List and History to help users manage their prescriptions.
d. **Prevent Medicine Shortages:** Notify users when medicines are Running Out, ensuring they refill in time to avoid treatment interruptions.
e. **Improve Accessibility:** Enable Speech-to-Text for easy medicine input and **Smart Suggestions** for medicine names when searching, enhancing efficiency for all users.
f. **Provide Emergency Support:** Integrate an Emergency SOS feature to quickly notify trusted contacts, automatically fetched from the phone contact list, during critical medical situations.
g. **Ensure Backend Reliability:** Utilize Java, Firebase Database, and Alarm Manager to ensure smooth performance, secure data storage, and accurate alarm scheduling.

1.3 Expected Outcome

The Medicine Tracker and Reminder with Emergency SOS app is expected to deliver the following outcomes:

a) **Improved Medication Adherence:** Users will receive timely medicine reminders, ensuring they take their medications on schedule, reducing missed doses and improving overall health outcomes.

b) **Effective Medicine Management:** Users can track the total number of medicines, monitor the last taken time with date, and maintain a medicine history, leading to better organization and awareness of their medication routine.

c) **Enhanced User Convenience:** Features like speech-to-text input for adding medicines and smart suggestions for medicine names during searches will simplify the user experience, making the app accessible for all age groups.

d) **Prevention of Medicine Shortages:** The app will notify users when medicines are **running out**, ensuring timely refills and uninterrupted medication adherence.

e) **Emergency Preparedness:** The Emergency SOS feature will provide quick access to trusted contacts during medical emergencies, improving response time and ensuring user safety.

f) **Secure and Efficient Performance:** With a Firebase Database for real-time data storage and Alarm Manager for accurate scheduling, the app will offer reliable performance, secure data handling, and seamless user experience.

.

1.4 Report Layout

The report is organized into seven chapters to provide a comprehensive understanding of the Medicine Tracker and Reminder. Chapter 1 introduces the project, including its motivation, objectives, expected outcomes, and report structure. Chapter 2 provides background information, related work, comparative studies, problem scope, and challenges. Chapter 3 focuses on requirement specifications, covering business process models, block diagrams, feasibility studies, use case modeling, and design requirements. Chapter 4 details the design specifications, including front-end, back-end, user experience (UX), and implementation tools such as Java, Restful APIs, and Firebase.

CHAPTER 2
BACKGROUND

2.1 Introduction

The increasing reliance on technology in healthcare has paved the way for innovative solutions that simplify medication management and enhance patient care. In developing countries like Bangladesh, where healthcare accessibility and adherence to prescribed treatments can be challenging, digital health tools are becoming essential. The Medicine Tracker and Reminder with Emergency SOS app addresses these issues by leveraging modern technologies to improve medication adherence, streamline medicine tracking, and provide emergency assistance. This chapter explores the foundational aspects of the project, including existing related work, comparative studies, and the scope of the problems faced by individuals in managing their medications. It also discusses the challenges encountered in designing and implementing such an application, particularly in the context of a resource-constrained environment. By examining these elements, this chapter provides a thorough understanding of the project's context and significance, laying the groundwork for its development.

2.2 Related Works

Medication management has been a critical focus in healthcare technology, leading to the development of various applications aimed at improving adherence and ensuring patient safety. Numerous mobile apps have been created globally to address this need, offering features such as medicine reminders, dosage tracking, and refill alerts. Apps like MediSafe and Pill Reminder have gained popularity for their ability to send timely notifications and provide detailed medication schedules. However, these applications often cater to broader audiences, lacking localization and context-specific features for regions like Bangladesh.

In the Bangladeshi context, few apps specifically address the unique challenges faced by users, such as language barriers, affordability, and limited digital literacy. While some apps attempt to localize features, they often fail to integrate critical functionalities like emergency services or seamless voice input for adding medicines. Studies in mHealth

(mobile health) also highlight the importance of combining medication management with emergency support systems, especially in low-resource settings. Previous research and apps have primarily focused on standalone functionalities, such as medicine reminders or healthcare consultations, but rarely integrate them into a comprehensive solution.

Furthermore, many existing solutions rely heavily on consistent internet connectivity, which can limit accessibility for users in rural or underserved areas. These gaps underline the need for a localized, user-friendly, and feature-rich app that not only supports medication adherence but also provides emergency assistance tailored to the Bangladeshi context.

2.3 Comparative Studies

Developing a robust Medicine Tracker and Reminder with Emergency SOS app requires an understanding of existing solutions to identify strengths, limitations, and areas for improvement. This comparative study analyzes popular medicine management apps globally, focusing on their features, usability, and relevance to the Bangladeshi context.

Apps like MediSafe, Pill Reminder, and CareClinic are widely recognized for their advanced features, such as personalized medicine schedules, refill reminders, and family sharing options. MediSafe, for example, offers integration with wearable devices and detailed analytics for user health trends, catering to tech-savvy audiences. However, these apps often lack support for low-resource settings, and features like emergency contact integration or local language support are absent, making them less accessible for non-English speaking users in developing countries.

In Bangladesh, healthcare-focused apps like Doctorola, Jeeon, and Tonic primarily emphasize telemedicine and doctor consultations. While these apps play a vital role in connecting patients with healthcare providers, they typically do not offer comprehensive medication management systems. Furthermore, these apps lack features like medicine reminders, inventory tracking, and emergency SOS services, which are crucial for individuals managing chronic illnesses or complex medication schedules.

2.4 Scope of the Problem

Medication non-adherence is a significant issue in healthcare, particularly in developing countries like Bangladesh, where limited access to medical resources, poor health literacy, and a lack of technology-driven solutions exacerbate the problem. Many individuals, especially the elderly and those with chronic illnesses, struggle to maintain consistent medication routines due to forgetfulness, unorganized schedules, or a lack of timely reminders. This can lead to serious health complications, prolonged illnesses, and increased healthcare costs. Additionally, there is often inadequate monitoring of medication usage, with no system to track the total number of medicines, the last intake time, or when a refill is required. This results in users running out of essential medicines without prior notice, disrupting treatment plans. The problem becomes even more pronounced in rural and underserved areas where access to healthcare facilities is limited, making it vital for individuals to manage their medications independently.

Emergency medical situations present another critical challenge. In Bangladesh, many individuals do not have quick access to emergency services or a reliable system to alert trusted contacts during health crises. This lack of preparedness can lead to delayed medical attention and life-threatening consequences.

Furthermore, existing medicine management apps either lack local relevance or are too complex for users with limited digital literacy. Features such as voice input for easy addition of medicines, local language support, and affordability are often missing, limiting the utility of current solutions in the Bangladeshi context. The scope of the problem highlights the urgent need for a comprehensive, user-friendly, and localized solution that addresses these challenges by improving medication adherence, providing timely reminders, tracking medicine usage, and enabling emergency support.

2.5 Challenges

The development of the Medicine Tracker and Reminder with Emergency SOS app faces several challenges, including addressing digital literacy and accessibility issues, as a significant portion of the population in Bangladesh, particularly in rural areas, has limited technological knowledge. The app needs to provide multilingual support, especially in

Bengali, and incorporate accurate speech-to-text functionality. Integrating with local emergency services poses another challenge due to the varying infrastructure across the country, particularly in rural areas with limited connectivity. Ensuring data security and privacy while complying with local laws is crucial, as the app will store sensitive health data. Additionally, maintaining affordability and scalability is challenging, as the app must be accessible to a wide range of users, including those in lower-income brackets, and function across various mobile devices with differing capabilities. Ensuring consistent user engagement and retention is also critical to the app's success, requiring an intuitive and engaging user experience. Addressing these challenges requires careful design, robust security, and continuous testing to make the app effective, accessible, and sustainable for its target audience in Bangladesh.

CHAPTER 3
REQUIRMENT SPECIFICATION

3.1.1 Business Process Model

Business process modeling is a technique used by business analysts to visually represent and analyze how work gets done within an organization. By mapping out these processes, companies can identify inefficiencies, eliminate redundancies, and streamline operations for greater efficiency and improved performance.

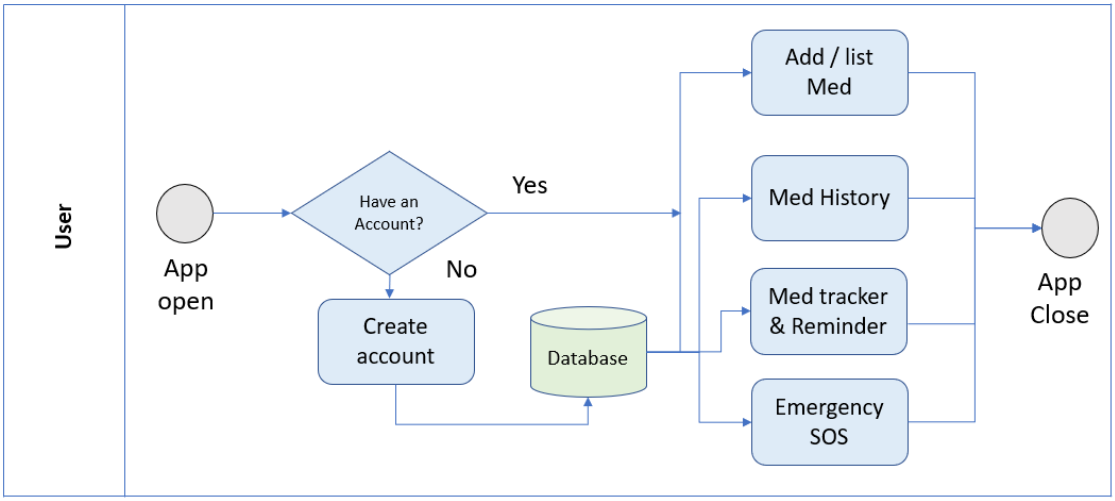

Figure 3.1: Business Process Model

The diagram illustrates the user flow for a medical app. Upon opening, the app checks for an existing account. If found, the user accesses their medical history. Otherwise, they create an account. The core functionalities include adding/listing medications, tracking intake with reminders, and an emergency SOS feature. All user data is likely stored in a database. This structure ensures a user-friendly experience for both new and existing users while providing essential features for medication management and emergency situations.

3.1.2 **Block Diagram**

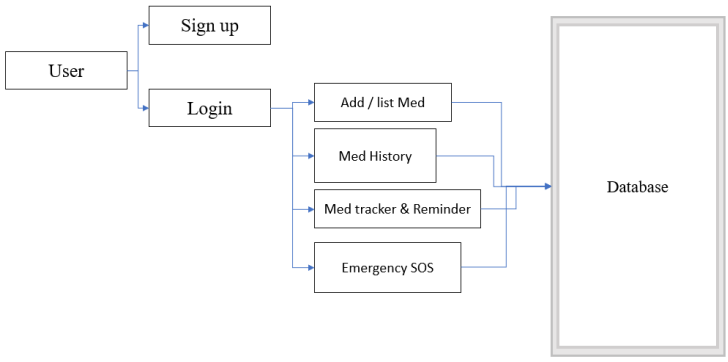

Figure 3.2: Block Diagram

This flowchart illustrates a user-centric medical management system. Users can begin by signing up and logging into the system to access its core functionalities. These include adding or listing medications, viewing medication history, tracking medications with reminders, and utilizing an Emergency SOS feature for urgent situations.

3.2 Requirement Collection and Analysis

The development of the Medicine Tracker and Reminder with Emergency SOS app faces several challenges, including addressing digital literacy and accessibility issues, as a significant portion of the population in Bangladesh, particularly in rural areas, has limited technological knowledge. The app needs to provide multilingual support, especially in Bengali, and incorporate accurate speech-to-text functionality. Integrating with local emergency services poses another challenge due to the varying infrastructure across the country, particularly in rural areas with limited connectivity. Ensuring data security and privacy while complying with local laws is crucial, as the app will store sensitive health data. Additionally, maintaining affordability and scalability is challenging, as the app must be accessible to a wide range of users, including those in lower-income brackets, and function across various mobile devices with differing capabilities. Ensuring consistent user engagement and retention is also critical to the app's success, requiring an

intuitive and engaging user experience. Addressing these challenges requires careful design, robust security, and continuous testing to make the app effective, accessible, and sustainable for its target audience in Bangladesh.

3.2.1 Feasibility Study

Conducting a feasibility study is crucial when developing an app. After selecting an idea, it's essential to evaluate whether the concept is practical and suitable. This involves assessing factors such as the uniqueness of the idea, its relevance to the target audience, the app's usability and stability, and how effectively it addresses the users' problems. For example, suppose you plan to create an app to solve a significant issue faced by Bangladeshi rickshaw pullers, investing a substantial amount of money in its development. Even if the app is well-designed and functions flawlessly, it may not succeed if your target users cannot or do not adopt it.

Therefore, before developing the app, you must study the target audience. Consider questions such as: How many rickshaw pullers own smartphones? Do they have internet access? Are they interested in using such a service? What are the conditions in their environment? Understanding these aspects ensures that your service is feasible and tailored to the needs and capabilities of your audience.

I conducted a feasibility study to evaluate my idea and its implementation. First, I searched the Google Play Store and found no app, website, or platform that offers a similar book-sharing service specifically for our university campus. This confirms that my idea is unique. Next, I reached out to students from various departments to gather their opinions. I asked if such an app would be helpful, how they would prefer to share their books conveniently, and what features they would like in the application.

Through this study, I discovered that students face several challenges. For instance, after completing a semester, many books become almost useless or irrelevant for the next semester. Additionally, students often want to sell their old books but lack a suitable platform to do so. Moreover, there are no well-stocked bookstores near the campus to address this need. Based on these insights, I believe this idea has significant potential and will likely be in high demand. In a feasibility study for an Android or web application, it

is essential to understand user activity, such as how many users will access the app or website simultaneously and the expected traffic load. This information is critical for developers to design the database, determine its storage capacity, and ensure the system can handle the maximum number of users efficiently. In this case, the app is unlikely to have a large user base initially. Therefore, I designed the app's database, storage capacity, and traffic management system to accommodate a moderate number of users effectively during its initial phase.

3.3 Use Case Modeling and Description

A use case represents how a user engages with a system to fulfill specific business goals. It highlights the functional components or products of the system. Use case diagrams typically include two main elements: actors and use cases. Actors are entities (users, systems, or external parties) that interact with the system to perform actions or achieve objectives.

The provided diagram illustrating the relationships between the actors and the system's use cases.

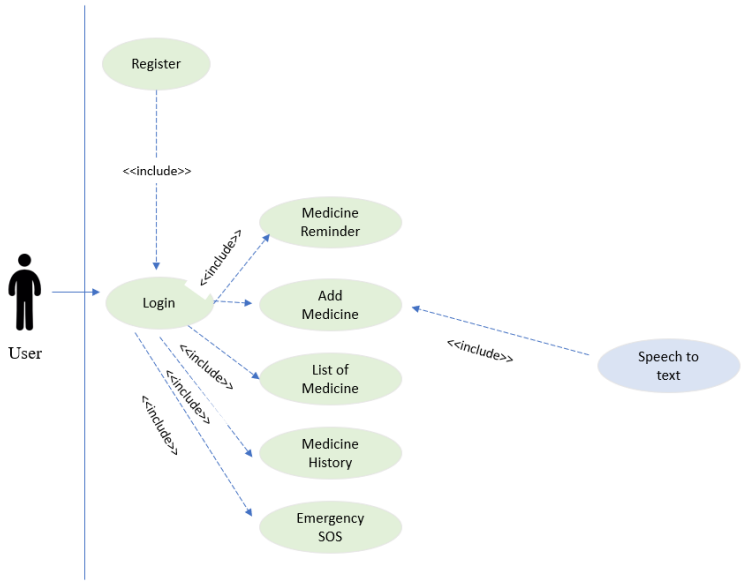

Figure 3.3: Use Case Diagram

3.4 Logical Data Model

A logical data model is a diagram that outlines how data is structured and stored in a database. It includes all the entities, their attributes, and the relationships between them. This type of diagram is commonly referred to as an Entity-Relationship (E-R) diagram.

The logical data model serves as a blueprint for designing the database and ensures that data is organized efficiently to meet the system's requirements. It defines the data elements, their properties, and the connections between different data entities, which helps in understanding how various components of the database interact.

By representing data in a logical structure, this model simplifies the process of database creation and maintenance. It provides a clear visualization for database designers and developers, making it easier to identify potential issues, optimize database performance, and ensure data integrity. Additionally, the logical data model is independent of the database management system (DBMS), allowing for flexibility in implementation across different platforms.

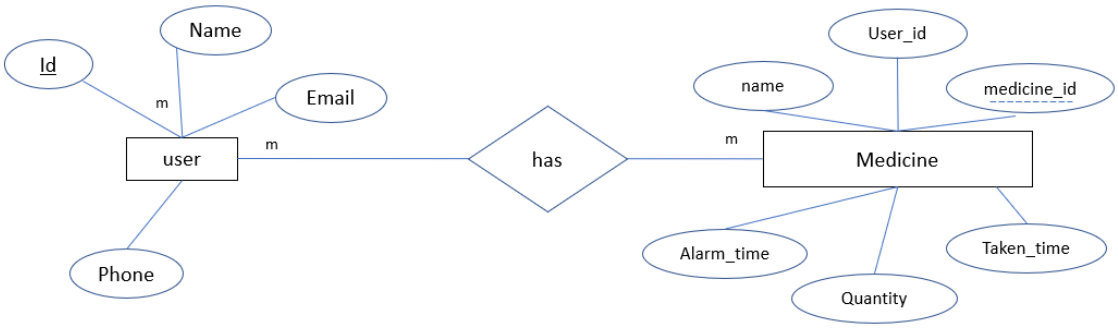

Figure 3.4: ER Diagram of the system

3.5 Design Requirements

The development of an application begins with requirement design, as it serves as the foundation for all subsequent processes. The key requirements are outlined below:

- The design of the Medicine Tracker and Reminder app must cater to the core features and functionalities outlined in the project. These requirements focus on ensuring a seamless user experience, intuitive interface, and optimized performance while delivering the app's key features. The design should balance user-friendliness, accessibility, and efficiency, addressing both functional and non-functional requirements.
- **Simplicity and Clarity:** The app should feature a clean, intuitive interface with minimal clutter, making it easy for users to navigate, especially for those with limited digital literacy.
- **Voice Input Integration:** A speech-to-text interface should be designed to allow users to easily add medicines and schedules by speaking, reducing the need for manual typing. The speech recognition system should be accurate and responsive to the Bengali language.
- **Responsive Layout:** The app's layout should automatically adjust to different screen sizes and resolutions, ensuring a consistent experience across various Android devices, from low-end smartphones to high-end models.
- **Color Scheme and Branding:** Use a soft, calming color palette (e.g., blues, greens) with high contrast for important features, such as reminders and emergency buttons, to make them stand out. The app design should reflect a health-focused, professional theme that resonates with users.

CHAPTER 4

DESIGN SPECIFICATION

4.1 Front-end Design

This is a vital element of mobile application development, emphasizing the presentation layer where users engage directly with the system. In today's fast-paced world, simplicity and efficiency are highly valued. With this in mind, I focus on designing a user-friendly interface that enables smooth interactions. To further improve the user experience, I often integrate Google Material Design principles, which offer a clean, intuitive layout for effortless navigation and interaction.

Figure 4.1: Homes Page and List of Medicine

In the figure 4.1 is home page of this app. This image shows a screenshot of a mobile app interface designed to help users track their medication intake. The app displays information about three different medications: Carvista, Cefaclav, and Napa Extend. For each medication, the following details are provided:

- **Medicine Name:** The name of the medication.
- **Prescribed Time:** The time(s) at which the medication should be taken.
- **Prescribed Medicine:** The total number of doses prescribed.
- **Total Taken:** The number of doses taken so far.
- **Remaining Medicine:** The number of doses remaining.
- **Last Taken:** The date and time the medication was last taken.

The app also includes buttons for navigating to the "Home" screen, accessing a calendar, sending a message, and viewing a profile. Additionally, there is a "+" button, likely used to add a new medication to the list.

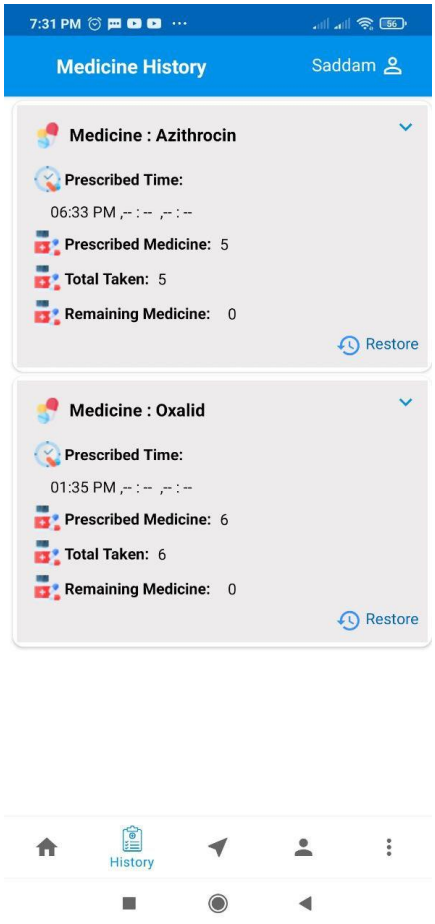

Figure 4.2: Medicine History

In figure 4.2 is a showcases a mobile app interface designed for medication history. It displays a list of two medications: Azithrocin and Oxalid. For each, it provides details like prescribed time, total prescribed doses, doses taken, and remaining doses. The app offers features like navigation, history tracking, messaging, profile access, and additional

options. A "Restore" button allows users to undo recorded doses, providing flexibility in medication tracking.

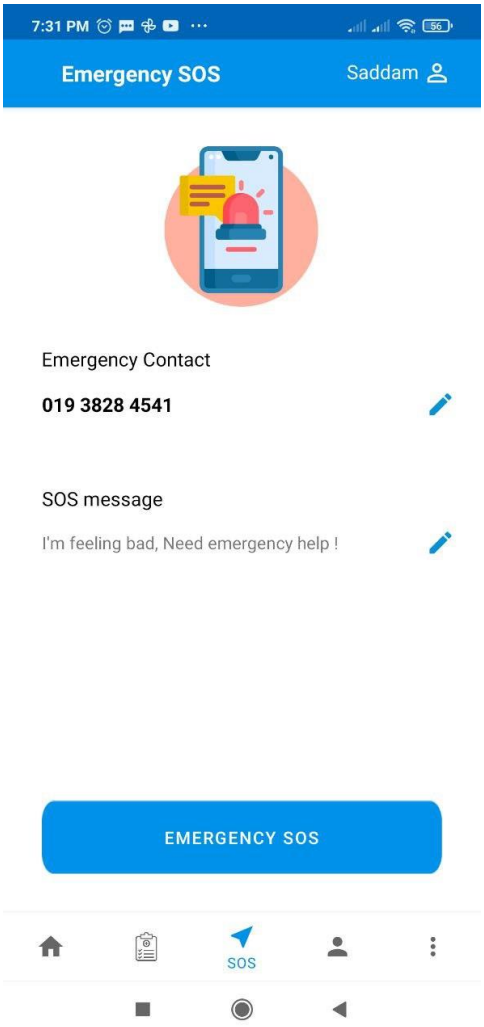

Figure 4.3: Emergency SOS

The image shows a mobile app interface designed for emergency SOS. It shows an emergency contact number, an SOS message field, and a prominent "EMERGENCY SOS" button. The app also includes a navigation bar with icons for Home, History, Messaging, Profile, and additional options. The interface is simple and clear, making it easy to use in emergency situations.

4.2 Back-end Design

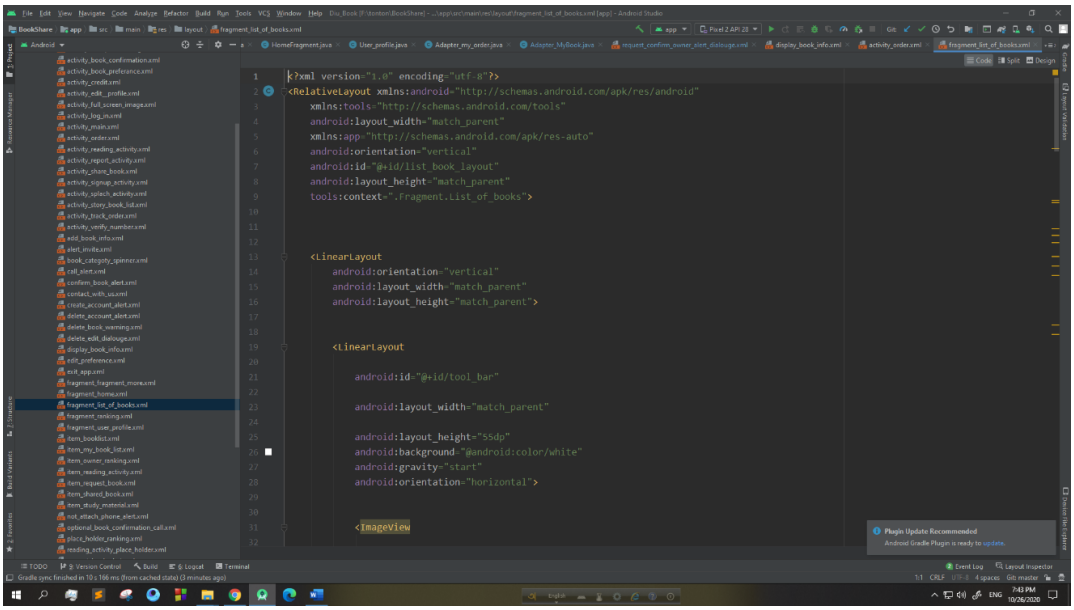

Figure 4.4: Add Medicine

The app also includes buttons for navigating to the "Home" screen, accessing a calendar, sending a message, and viewing a profile. Additionally, there is a "+" button, likely used to add a new medication to the list.

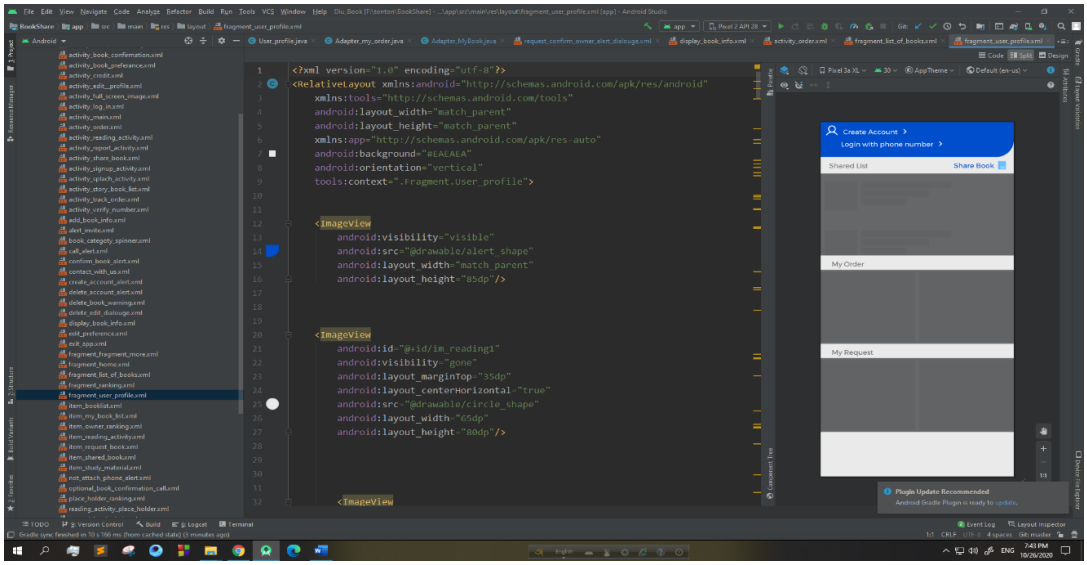

Figure 4.5: List of Medicine

This image shows a screenshot of a mobile app interface designed to help users track their medication intake. The app displays information about three different medications: Carvista, Cefaclav, and Napa Extend. For each medication, the following details are provided:

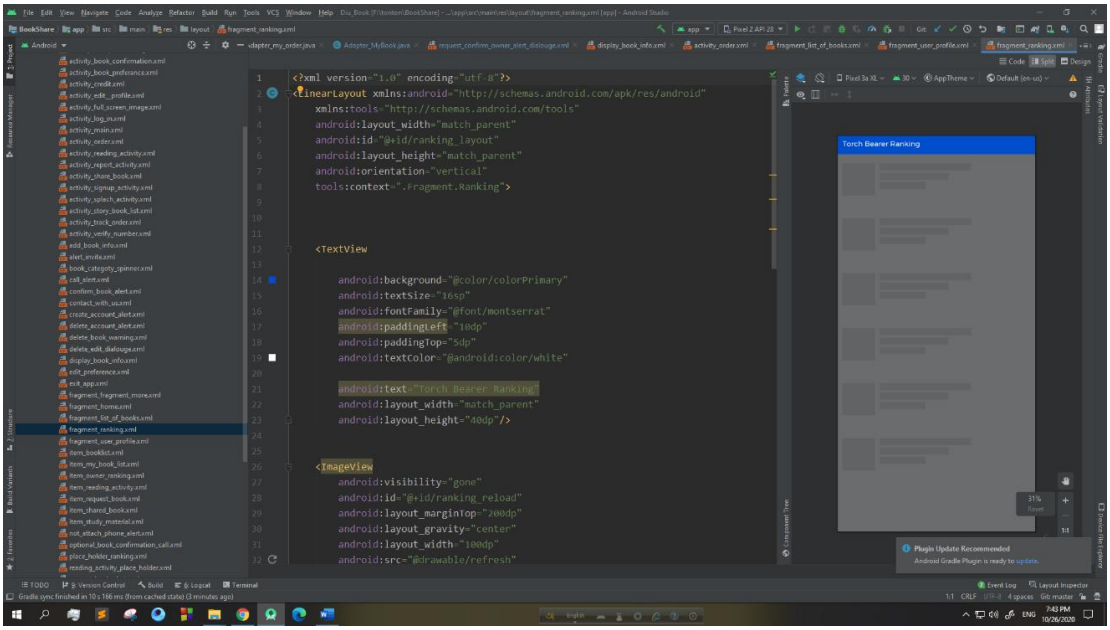

Figure 4.6: Emergency SOS

The image shows a mobile app interface designed for emergency SOS. It shows an emergency contact number, an SOS message field, and a prominent "EMERGENCY SOS" button. The app also includes a navigation bar with icons for Home, History, Messaging, Profile, and additional options. The interface is simple and clear, making it easy to use in emergency situations.

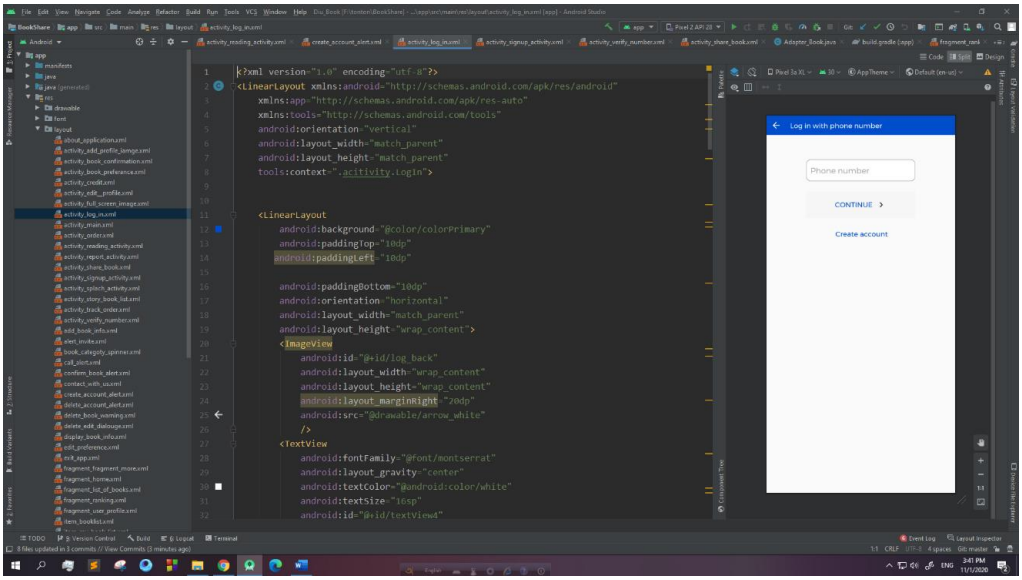

Fig 4.7 : User Login

The Account Settings section enables users to modify preferences like changing their password, managing notification settings, and configuring privacy options to control the visibility of their information. Additionally, an Activity Log provides a detailed history of the user's actions, such as login records, recent interactions, or account updates, to help monitor usage and detect unauthorized activity. Finally, the page includes a Logout Button to ensure users can securely end their session and prevent unauthorized access to their account.

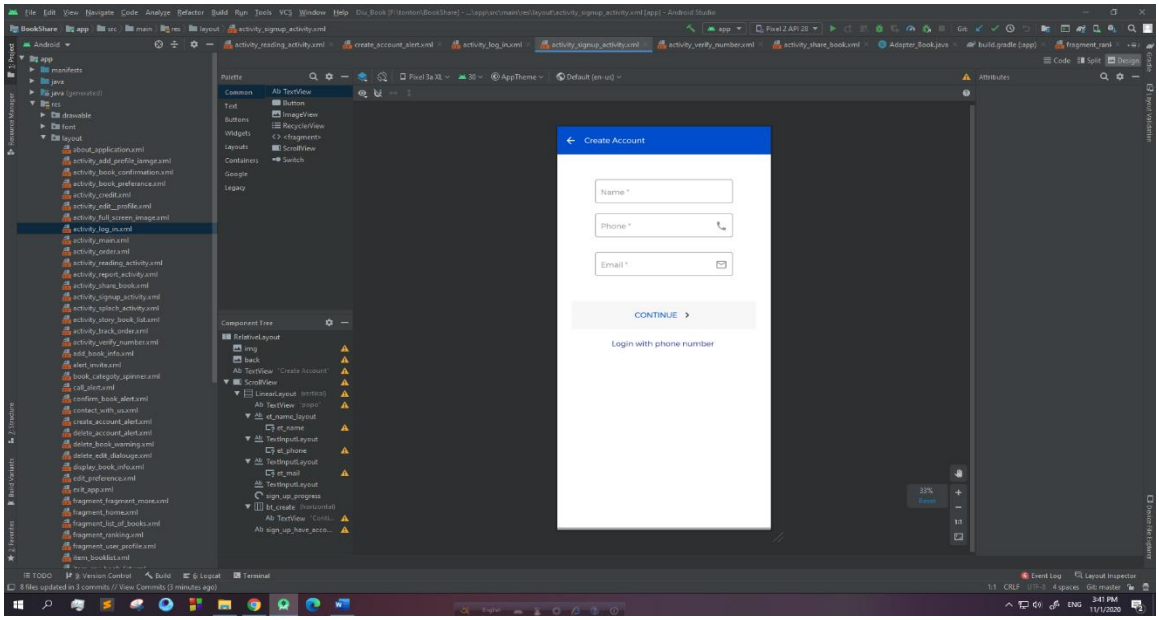

Figure 4.8: User create account

The User Account Page is designed to allow users to view and manage their personal information, account settings, and activity history. It includes a Profile Section displaying details such as the username, full name, email address, and profile picture, with options to edit these details.

4.3 Interaction Design and UX

Interaction design is a fundamental aspect of product development, focusing on planning and designing the communication and interaction between users and the system or content. While it is commonly associated with digital products such as applications or websites, its principles can apply to various other interfaces.

Interaction design involves a systematic process aimed at understanding user problems, analyzing and measuring these issues, identifying expected outcomes, and implementing solutions based on the findings. By addressing these challenges, interaction design seeks to create solutions that are both effective and user-centered.

The ultimate goal of interaction design is to develop products that enable users to achieve their objectives in the most efficient and straightforward way possible. This includes designing interfaces that are intuitive, responsive, and tailored to user needs, thus enhancing the overall user experience. Effective interaction design not only resolves user issues but also ensures that the product aligns with their expectations and fosters seamless engagement with the system.

4.4 Implementation Requirements

To successfully implement this app, several programming languages and tools are required. Key languages include **Java**, **PHP**, and **SQL**, each serving a specific purpose in the development process. Java is essential for building the core functionality of the application, especially for Android development. PHP can be used to manage server-side operations and backend logic, while SQL handles database management, ensuring efficient storage, retrieval, and manipulation of data.

Additionally, the development process relies on **Android Studio**, an open-source platform provided by Google. Android Studio offers a comprehensive environment for

developing, testing, and debugging Android applications, making it an indispensable tool for the project.

To meet all the app's requirements, integrating third-party APIs or libraries may also be necessary. These external resources can add advanced features, such as payment gateways, social media integration, or location services, without requiring extensive custom development. Leveraging these tools and technologies ensures that the application is robust, functional, and capable of delivering a seamless user experience.

.

4.4.1 Java

While languages like Kotlin, Flutter, and Dart are gaining popularity, Java remains a cornerstone for developing native Android applications. Built upon the object-oriented principles of Java, this approach leverages the Android SDK, providing a robust and widely-adopted framework. Java's portability, security, and ability to effectively utilize device-specific features make it a strong choice for creating high-quality mobile applications, particularly within the constraints of limited memory and processing power often found in mobile devices.

4.4.2 PHP

Our web application employs a common architecture where MySQL serves as the underlying database. PHP acts as the intermediary, retrieving and manipulating data from MySQL based on parameters received from our application. This allows for dynamic data interaction, enabling features like fetching specific records or executing complex queries. PHP is also instrumental in creating RESTful APIs, providing a robust interface for communication with other systems. While PHP is a widely-used choice for this purpose, Node.js offers an alternative for building efficient and scalable APIs.

4.4.3 Restful API

Retrofit is a REST client library for Android and Java, designed to make HTTP requests and handle responses from RESTful APIs. Developed by Square, Retrofit supports data formats beyond JSON, such as Simple XML and Jackson. In this context, a REST API refers to a set of functions that allow programmers to send and receive data over HTTP

protocols like GET, POST, UPDATE, and DELETE. Essentially, a RESTful API enables seamless data exchange through HTTP requests.

4.4.4 Firebase

Some common relational database management systems (RDBMS) that utilize SQL include Oracle, Sybase, Microsoft SQL Server, Access, and Ingres, among others. While most database systems rely on SQL, many also offer proprietary extensions unique to their platforms. Despite these variations, the standard SQL commands—such as **SELECT**, **INSERT**, **UPDATE**, **DELETE**, **CREATE**, and **DROP**—are sufficient to perform almost all essential database operations.

This tutorial will guide you through the basics of these commands and provide practical examples using an SQL interpreter, allowing you to apply and practice what you learn effectively.

4.4.5 Google Material Design

Material Design, developed by Google, is a design language that offers a comprehensive set of guidelines to help designers and developers create interfaces that reflect Google's design principles. It explains why Google's elements look and behave as they do. In my app, I have implemented Material Design principles, including content spacing, text size, colors, margins, grids, scaling, active buttons, and app color schemes. Overall, Material Design provides a complete framework for creating visually consistent and user-friendly apps.

4.4.6 Third Parity API Library

Third-party libraries are widely used in Android applications to streamline development and enhance functionality. However, they can introduce potential security and privacy concerns, as well as blur the distinction between application and library code. Accurate library detection becomes crucial, especially since developers may customize library code during integration, and code obfuscators may remove unused parts during the build process. In the app, I have utilized several third-party libraries, such as EventBus, Retrofit, OkHttp, Picasso or Glide, and Gson, to make development more efficient and productive.

CHAPTER 5

IMPLEMENTATION

5.1 Implementation of Database

Databases store and organize various types of data, providing a simple, efficient, and flexible platform for storage and retrieval. Data is structured into tables created by analyzing related records. Designing a database requires a clear understanding of the table structure, including its name, columns, primary keys, and foreign keys.

In my application, I used MySQL as the database management system to create and manage the database. It offers a systematic approach to retrieving, creating, updating, and deleting data efficiently.

5.1.1 Firebase Database:

Firebase, a backend-as-a-service platform provided by Google, offers a wide range of tools and services for developers to build, manage, and scale applications with ease. It is known for its real-time database, which allows data to be synchronized across all clients instantly, making it ideal for apps requiring live updates, such as messaging or collaborative platforms. Firebase provides a variety of additional services, including cloud storage, Firebase Machine Learning, cloud functions, user authentication, hosting, cloud messaging, push notifications, and test labs, making it a comprehensive solution for both front-end and back-end needs. Developers can get started quickly with a Google Gmail account, and the platform simplifies database management with its parent-child tree structure, storing data in a JSON-like format. While Firebase offers a free tier, its services are not entirely free, and users must pay if they exceed the usage limits, according to Firebase's pricing model. Its flexible pay-as-you-go structure, combined with powerful features, makes Firebase an excellent choice for developers seeking to streamline backend services while scaling applications.

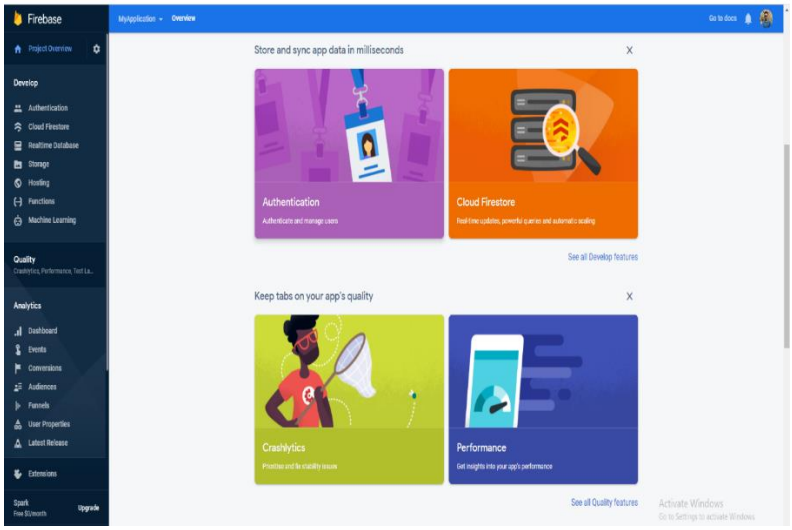

Figure 5.1: Firebase Dashboard

5.2 Implementation of Front-end Design

Front-end design is crucial for mobile applications, as it is the primary interface users interact with. Regardless of how robust the back-end is, a poorly designed front-end can deter users. A key aspect of effective design is responsiveness, ensuring the app looks consistent across various screen sizes. In my app, I primarily use ConstraintLayout, which is highly responsive and adaptable for all display sizes. Additionally, I occasionally use LinearLayout with weight sum to achieve responsive designs.

Android Layout Design: The Layout Editor simplifies creating layouts by allowing developers to drag and drop UI elements into a visual design editor instead of manually writing XML code. It previews layouts on different Android devices and screen sizes, ensuring compatibility.

ConstraintLayout Design: As the default layout in Android Studio, ConstraintLayout offers flexibility by allowing objects to be positioned relative to containers, other objects, or specific guidelines. It supports complex, dynamic, and responsive designs in a flat hierarchy, even allowing for animations.

5.3 Testing Implementation

Based on USER Activity

Test Case	Test input	Expected output	Actual Output	Result	Tested On
1. Database Connection	Tested via API	Connection established successfully.	Connection successful.	Passed	28-10-2024
2. Create account	Tested by filling the required fields and using the Insert API.	Account created successfully.	Account creation successful.	passed	29-10-2024
3. login	Tested by filling the required fields and using the login API.	Login working	Login Authentication working	passed	27-10-2024
4. Add Med	Name and others info	Displayed invalid information warning.	Warning message displayed	passed	27-10-2024
5. List of Med	Empty filled	Displayed invalid information warning.	Displayed invalid information warning.	passed	27-10-2024
6. Show calculation	Quantity	Will show actual data	Working	passed	27-10-2024
7. Delete 8. Account	Delete API	Successfully deleted	Working	Passed	27-10-2024

Table 5.1: Table of App Unit Test Case

This table outlines the results of various test cases conducted on a system. Each test case includes details about the functionality being tested, the input provided during testing, the expected outcome, the actual output observed, and the overall result. The tests cover several core functionalities such as database connection, account creation, login, adding medications, listing medications, showing calculations, and deleting accounts.

- **Database Connection**: The test was conducted by testing the API connection, and the expected output was a successful connection. The actual output matched the expected result, and the test passed successfully on 27-10-2024.
- **Create Account**: The test involved editing the test fields and inserting data through the API. The expected result was a successfully created account, which was confirmed in the actual output. The test passed on 27-10-2024.
- **Login**: During this test, login credentials were entered, and the login API was tested. The expected result was a successful login, which was achieved. The test passed on 27-10-2024.
- **Add Medication**: The test involved entering the medication name and other relevant information. The expected outcome was an error message for invalid information, and the system displayed the warning message as expected. The test passed on 27-10-2024.
- **List of Medications**: This test involved leaving the fields empty and expecting the system to show an error. The system displayed the warning message as expected, and the test passed on 27-10-2024.
- **Show Calculation**: The test focused on entering a quantity value and checking if the system displayed the correct data. The actual output matched the expected result, and the test passed successfully on 27-10-2024.
- **Delete Account**: The test was done through the delete API, and the expected result was the successful deletion of the account. The actual output confirmed the account was deleted, and the test passed on 27-10-2024.

CHAPTER 6
ENGINEERING STANDARDS AND DESIGN CHALLENGES

6.1 Software Standards

Programming Languages:

- Frontend: JavaScript, HTML5, CSS3 (for web applications); Kotlin or Java (for Android apps)
- Backend: Java, PHP, Python, or Node.js (depending on the server-side framework)
- Database: SQL (MySQL, PostgreSQL) or NoSQL (Firebase, MongoDB)

Frameworks:

- Frontend: React, Angular, or Vue (for web-based applications); Flutter or React Native for cross-platform mobile apps.
- Backend: Spring Boot (Java), Laravel (PHP), Express (Node.js)

Libraries and Tools:

- Use of Google Firebase for real-time database, authentication, and cloud services.
- Google Material Design for UI/UX design patterns.

Database Standards:

- SQL: Use of normalized schema for relational databases.
- NoSQL: Use of a flexible document-based structure for Firebase or MongoDB.

Security:

- Encryption: Use of SSL/TLS for secure data transmission.
- Authentication: OAuth 2.0, Firebase Authentication for user login and management.
- Data Privacy: Compliance with GDPR, HIPAA, or other relevant standards based on project scope.

6.2 Hardware Standards

 Server Specifications:

- Cloud-Based Hosting: Use of cloud services like Google Cloud Platform database management, and serverless computing.
- Server Instance: Virtual machines or containers (Docker) with sufficient processing power (CPU, RAM) based on application load.
- Storage: SSD-based storage for high performance and quick data access.

Mobile Device Compatibility (for mobile apps):

- Android: Devices with Android versions 7.0 (Nougat) or higher.
- iOS: For cross-platform apps, devices supporting iOS 12 or later.

Peripheral Devices: If needed, support for hardware like barcode scanners, printers, or IoT devices (depending on the project).

Network Infrastructure:

- Reliable internet connectivity (minimum 4G for mobile apps, or Wi-Fi for web apps).
- Support for both IPv4 and IPv6.

Backup Systems: Regular backups of databases and application data, ideally in multiple geographic locations for redundancy.

6.3 Communication Standards

Network Protocols:

- HTTP/HTTPS: For secure communication between the client and server.
- RESTful APIs: For communication between front-end and back-end systems.
- WebSockets: For real-time communication, especially for live updates in applications.

Messaging Protocols:

- MQTT or HTTP-based push notifications: For sending messages to users in real time.
- APNs (Apple Push Notification Service) for iOS and FCM (Firebase Cloud Messaging) for Android to enable push notifications.

Data Formats:

- JSON: For structured data exchange between the server and client.
- XML: If required for legacy systems or specific data interchange needs.

6.4 Impact on Society

The Medicine Tracker and Reminder with Emergency SOS app can have a transformative impact on society, particularly in Bangladesh. By improving medication adherence, it ensures timely treatment for individuals managing chronic conditions, thereby enhancing overall health outcomes. The app empowers vulnerable groups, such as the elderly and those with limited literacy, by offering a user-friendly interface and voice-input features. It also promotes emergency preparedness through its SOS functionality, enabling quick responses in critical situations. Additionally, the app encourages healthcare awareness by helping users track medication history and inventory, fostering a proactive approach to health management. Ultimately, the app contributes to reducing healthcare costs by preventing complications and hospitalizations, while promoting healthier communities through widespread adoption, offering significant benefits to both individuals and society as a whole.

6.2 Impact on Environment

The **Medicine Tracker and Reminder with Emergency SOS** app has a minimal environmental impact, but it contributes positively in several indirect ways. By promoting better healthcare management through digital solutions, the app reduces the need for physical paper prescriptions, medical records, and reminder systems, thereby decreasing paper waste and supporting environmental sustainability. Additionally, it encourages users to adopt digital healthcare solutions, reducing the carbon footprint associated with in-person doctor visits and paper-based prescriptions. The app’s lightweight and energy-efficient design ensures low resource consumption, minimizing its impact on device battery life and reducing overall electronic waste. Furthermore, by improving health outcomes and preventing unnecessary hospital visits, the app helps lower the demand on healthcare infrastructure and associated energy consumption. In essence, while the app's direct environmental impact may be modest, it plays a role in promoting sustainability by reducing resource consumption and encouraging eco-friendly digital healthcare practices.

6.3 Ethical Aspects

The **Medicine Tracker and Reminder with Emergency SOS** app must prioritize user privacy and data security, ensuring compliance with privacy laws and encrypting sensitive health data. It should be inclusive, accessible to users with varying literacy and technical skills, and transparent in how it collects and uses data. The app must avoid bias, ensuring equal benefits for all users, regardless of background.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

The Medicine Tracker and Reminder with Emergency SOS app presents a promising solution to address key healthcare challenges in Bangladesh, particularly regarding medication adherence and emergency preparedness. By leveraging technology to remind users about their medications, track their health history, and provide an SOS feature for emergencies, the app aims to improve individual health outcomes and enhance the overall healthcare system. With its user-friendly design, accessibility features, and integration with trusted contacts, the app is poised to empower vulnerable groups and promote better health management. Furthermore, its potential to reduce healthcare costs and environmental impact underscores its long-term sustainability. Ultimately, this app not only addresses immediate health needs but also contributes to creating a more inclusive, efficient, and sustainable healthcare ecosystem in Bangladesh.

7.2 Future Scope

The **Medicine Tracker and Reminder with Emergency SOS** app offers significant potential for future enhancements and expansions. Firstly, integrating AI-driven features, such as personalized medication recommendations based on user health data, could improve medication adherence and treatment outcomes. Additionally, expanding the app’s functionality to include integration with wearable health devices (e.g., smartwatches and fitness trackers) could provide real-time health monitoring and more accurate medication tracking. Another avenue for development is expanding the app's reach by including multilingual support to cater to diverse populations in Bangladesh and other regions. Furthermore, incorporating telemedicine capabilities would allow users to consult healthcare professionals directly through the app, bridging the gap between remote areas and medical services. Lastly, expanding the emergency SOS feature to include GPS-based location tracking for better response times from local authorities could enhance the app’s effectiveness in life-threatening situations. These developments would not only improve user experience but also broaden the app’s impact on public health.

REFERENCE

[1] Firebase Console for Realtime database, available at << https://firebase.google.com/ >>, last accessed on 14-01-2025

[2] Android Studio IDE, available at << https://developer.android.com/studio>>, last accessed on 14-01-2025

[3] MySQL database, available at << https://www.mysql.com/>>, last accessed on 14-01-2025

[4] JavaScript Web API , available at << https://www.w3schools.com/js/js_validation_api.asp>>, last accessed on 14-01-2025

[5] XML Tutorial, available at << https://www.w3schools.com/xml/>>, last accessed on 14-01-2025

[6] DocTime Telemedicine System, available at << https://doctime.com.bd/>>, last accessed on 14-01-2025

[7] Arogga Telemedicine Android App, available at << https://www.arogga.com/ >>, last accessed on 14-01-2025

Report Checking v1

ORIGINALITY REPORT

14%	12%	1%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	7%
2	Submitted to BRAC University Student Paper	2%
3	Submitted to Daffodil International University Student Paper	2%
4	Yaser Al-Worafi. "Patient Safety in Developing Countries - Education, Research, Case Studies", CRC Press, 2023 Publication	<1%
5	patents.google.com Internet Source	<1%
6	Submitted to Teaching and Learning with Technology Student Paper	<1%
7	Submitted to Southern Nazarene University Student Paper	<1%
8	Submitted to University of Greenwich Student Paper	<1%