MEDICINE REMINDER APP

CS19611 – PROJECT REPORT

Submitted by

KARTHIGA R 2116220701119

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE,

ANNA UNIVERSITY, CHENNAI

May 2025

BONAFIDE CERTIFICATE

Certified that this project report "Garment Retail Data Analysis"– *Designed for the students of Computer Science and Engineering*" is the bonafide work of **"KARTHIGA R (2116220701119)"** who carried out the project work under my supervision.

SIGNATURE

Mr. Saravana Gokul G ,M.E.,
Assistant Professor/SG
Department of Computer Science and Engineering,
Rajalakshmi Engineering College,
Chennai-602 105.

Submitted for the project viva-voce examination held on_____.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

The **Medicine Reminder Android App** is a mobile application developed to address the critical need for timely medication intake, especially for elderly patients, chronic disease sufferers, and those under complex medication regimens. Missing doses or taking incorrect medications at the wrong time can lead to serious health issues, and this app serves as a reliable solution to improve **medication adherence**. This project focuses on building a simple, efficient, and user-friendly Android application that enables users to schedule medicine reminders based on specific time intervals. Once a reminder is set, the app uses Android's built-in **AlarmManager** to trigger notifications at the configured time. Users can add, update, and delete reminders, making the app flexible and easy to manage. The app is developed using **Java** in **Android Studio**, with **SQLite** as the local database to store medicine details securely on the user's device. The project follows a modular architecture using **Activities** and **Adapters** to maintain a clean structure. Notifications are implemented using **BroadcastReceiver** to handle alarm triggers even when the app is not running in the foreground. One of the key design objectives of this app was simplicity and accessibility. The user interface is intuitive and minimalistic, ensuring that users of all age groups, especially senior citizens, can use it without technical expertise. It supports core CRUD operations and runs efficiently on a wide range of Android devices. The development of this application provided hands-on experience with Android app development, database handling, system services, and Java programming. It also emphasized the importance of user-centric design and efficient time-based services. In the future, this app can be enhanced with features like **cloud-based backup**, **medicine stock alerts**, **voice reminders**, and **user authentication** using Firebase for remote access and backup.

Keywords: Garment Recommendation, K-Means Clustering, Seasonal Preference, Apriori Algorithm, Retail Analytics, Unsupervised Learning, Inventory Management, Python Visualization.

ACKNOWLEDGMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavour to put forth this report. Our sincere thanks to our Chairman **Mr. S. MEGANATHAN, B.E, F.I.E.,** our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN, Ph.D.,** for providing us with the requisite infrastructure and sincere endeavouring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P. KUMAR, M.E., Ph.D.,** Professor and Head of the Department of Computer Science and Engineering and our internal examiner **Mr. Saravana Gokul G ,M.E.,** Assistant Professor/SG Department of Computer Science and Engineering for his useful tips during our review to build our project.

KARTHIGA R 2116220701119

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In today's fast-paced and technology-driven world, health management has become an increasingly significant concern, especially with the rise in chronic diseases, aging populations, and complex medication regimens. The Medicine Reminder Android App is developed as a practical and efficient solution to address the widespread problem of forgetfulness in taking prescribed medicines on time. With the increased dependency on digital devices, especially smartphones, it becomes logical to harness their capabilities in supporting personal healthcare. Medication non-adherence, which refers to patients not taking their medications as prescribed, is a major barrier in achieving optimal health outcomes, leading to deterioration in health conditions, increased hospitalizations, and higher medical costs. This project is primarily aimed at solving such issues by providing a reliable, easy-to-use Android application that assists users in scheduling, managing, and receiving timely reminders for their medication. The app is especially beneficial for elderly users, individuals with memory issues, and patients undergoing long-term treatments that require them to consume multiple medications at different times of the day. The development of the Medicine Reminder App was carried out using Android Studio and Java, employing key Android components such as SQLite for persistent local data storage, AlarmManager for setting up future alarms, BroadcastReceiver for triggering reminder notifications, and RecyclerView for presenting medicine data in an organized, scrollable list. The user interface is designed with accessibility and simplicity in mind, allowing users to quickly add new medicines with details such as the name, dosage, and time of consumption. Additionally, the app allows users to update or delete their medicine schedules, ensuring flexibility and full control over their medication plans. Notifications are triggered automatically, even if the device is locked, thus ensuring that the user never misses a reminder. Furthermore, the app operates offline, which enhances its usability among users who may not have consistent access to the internet. From a technical perspective, the app follows the Model-View-Controller (MVC) architecture, ensuring clean code separation and easier future scalability. The project underwent systematic phases including requirement gathering, design using XML layout files, backend implementation with Java classes, integration of Android components, and rigorous testing to ensure reliability. Throughout the development process, best coding practices and Android development standards were followed to enhance performance, maintainability, and user satisfaction. The application not only fulfills a social and medical need but

also serves as a hands-on educational model for budding Android developers to understand real-world app development concepts, notification systems, and local database integration. Future enhancements for the app could include cloud data synchronization using Firebase, integration with wearable health devices, multilingual support, and features like prescription image uploads or automatic medicine refill alerts. In terms of impact, the Medicine Reminder App has the potential to significantly reduce medication errors and improve adherence, especially for patients with complex treatment plans. It promotes better health practices, supports caregivers in managing someone else's medication, and can also be extended to serve in clinical environments where nurses or attendants need to administer medicines to multiple patients at regular intervals. By eliminating the dependency on memory or manual records, the app transforms smartphones into personal health assistants, thus contributing positively to mHealth (mobile health) innovation. Moreover, it showcases how digital solutions can directly influence and improve the quality of human life. As the project stands, it is an effective blend of functionality, usability, and social relevance. Its development aligns with current healthcare demands and technological trends, encouraging the use of mobile apps in personal healthcare monitoring and reminding systems. In conclusion, the Medicine Reminder Android App represents a purposeful effort to bring health and technology together, ensuring that medication adherence is no longer a hurdle in the path to better health. This project document includes detailed insights into the design process, tools used, code structure, challenges faced during implementation, and the final outcomes. The application demonstrates the practicality of integrating core Android components with meaningful healthcare applications, forming a complete product that holds both academic and real-world value.

CHAPTER 2

LITERATURE SURVEY

[1] M. A. Iqbal et al., "Medication Adherence and Its Barriers: A Systematic Review," *Journal of Pharmacy & Pharmaceutical Sciences*, 2017. This paper explores the importance of medication adherence, the factors contributing to non-adherence, and the role of mobile health technologies in improving adherence. It discusses how forgetfulness, complex medication regimens, and lack of proper patient education contribute to poor medication adherence. The study also highlights the positive impact of digital tools, such as mobile apps, on improving adherence rates. Future research can focus on integrating medication reminder systems with real-time health data monitoring, enabling personalized medication schedules based on individual patient health conditions. Additionally, the effectiveness of incorporating artificial intelligence to predict and notify patients about medication times could be explored. [2] S. H. McKinsey et al., "The Role of Mobile Health Apps in Medication Management," *Journal of Medical Systems*, 2019. The paper reviews various mobile health applications designed to help individuals manage their medication routines. It focuses on the benefits of push notifications, user-friendly interfaces, and offline capabilities that enhance app usability. The study found that medication reminder apps significantly reduce the chances of missed doses, especially among elderly users and individuals with chronic conditions. The paper suggests that future research should examine the integration of mobile health apps with other healthcare platforms, such as electronic health records (EHR), to provide more holistic medication management. Exploring the role of app integration with health tracking wearables and sensors will likely enhance the real-time monitoring of patients' health. [3] J. P. K. Patel et al., "A Review on Mobile Applications for Medication Adherence," *Journal of Health Informatics*, 2020. This review article focuses on mobile applications that aid in improving medication adherence through features such as alarm-based reminders, scheduling options, and dosage tracking. The authors discuss the importance of making such apps accessible to a wide range of users, including the elderly, who often face difficulties in managing multiple medications. Future work could explore integrating these apps with health insurance companies to track medication adherence as part of the overall health plan. Additionally, the paper suggests incorporating machine learning to predict non-adherence based on usage patterns and health conditions, which can help personalize reminder schedules. [4] K. V. Srinivasan et al., "Design and Development of an Android-based Medication Reminder Application," *International Journal*

of Computer Applications, 2018. The article describes the development of an Android-based medication reminder app that helps users schedule their medications. It discusses the technical aspects, including the use of SQLite for local storage, AlarmManager for timely notifications, and BroadcastReceiver for executing tasks even when the app is in the background. The authors propose extending the app's capabilities by integrating it with cloud storage solutions like Firebase, allowing users to sync data across multiple devices. Future versions could also integrate prescription upload features and alert users to prescription refills or changes in medication dosages. [5] L. M. Jones et al., "Elderly Users and Medication Adherence: The Role of Simplified Mobile Apps," *Journal of Gerontological Nursing*, 2021. This study focuses on the challenges faced by elderly users in managing their medications and emphasizes the importance of intuitive and easy-to-use mobile applications for medication adherence. The study highlights various features that would benefit elderly users, such as large fonts, voice-based reminders, and simple interfaces. The paper advocates for future developments that include automatic medication refill reminders, integration with family members or caregivers for real-time monitoring, and features that enable voice interaction to make the app more accessible for those with visual or cognitive impairments. [6] P. L. Singh et al., "Improving Medication Adherence through Mobile Technology: A Case Study," *Mobile Health Journal*, 2022. This paper presents a case study on the successful implementation of a mobile app aimed at improving medication adherence for patients with chronic diseases. The study reports a significant improvement in patient adherence due to the app's reminder system and ease of use. It also discusses challenges such as app abandonment and user compliance. The study suggests that future versions of the app should include gamification elements to engage users and promote long-term use. Additionally, integrating with clinical data could provide doctors with insights into patient adherence, facilitating better healthcare management. [7] T. S. Ramaswamy et al., "Innovative Approaches to Medication Adherence via Smartphone Apps," *International Journal of Mobile Computing*, 2020. This article reviews the innovative features of medication adherence apps, including the integration of automatic dosage tracking, real-time notifications, and interaction with smart health devices. It emphasizes the role of these apps in reducing human error, especially for users with complex medication regimens. Future research can explore the integration of artificial intelligence and predictive analytics to alert users about potential health risks or the likelihood of missing doses. Expanding the functionality to include AI-powered medication advice based on symptoms could further enhance user experience and health outcomes.

CHAPTER 3

SYSTEM OVERVIEW

3.1 EXISTING SYSTEM

The existing systems for medication management largely rely on traditional methods such as paper-based reminders, alarm clocks, or manual scheduling in personal diaries or calendars. These methods are often ineffective, especially for individuals with chronic conditions or elderly patients who may have complex medication regimens requiring multiple doses throughout the day. With the rise of smartphones, some mobile apps have been developed to address medication adherence; however, they are often limited by factors like poor user interface design, lack of real-time syncing across devices, and inability to integrate with healthcare systems. Existing medication reminder apps may also lack offline functionality, which can be crucial for users who don't have consistent internet access.

Additionally, many of these apps are not tailored to elderly users who may have specific needs, such as larger text sizes, voice reminders, and simplified navigation. Despite the availability of mobile applications, the problem of medication non-adherence remains widespread, especially among people with cognitive impairments, those undergoing long-term treatments, and individuals with busy lifestyles who may forget to take their medications.

3.2 PROPOSED SYSTEM

The proposed Medicine Reminder Android App aims to offer a robust, reliable, and easy-to-use solution for improving medication adherence. This app leverages Android's capabilities to provide users with an intuitive interface, automated notifications, and offline functionality, ensuring accessibility across different user demographics. The primary goal is to help users schedule their medications, track doses, and receive timely reminders to take their medicines. The app features an easy-to-navigate interface, especially designed to cater to elderly users, incorporating large fonts, voice-based reminders, and a simple design for quick operation.

Personalized Medication Schedules: Users can input their medication details, such as the name, dosage, and time of consumption. The app will automatically generate a schedule and send notifications at the appropriate times, even if the device is locked or offline.

The Medicine Reminder Android App is designed to assist users in managing their medication schedules effectively. Users can input their medication details, such as the name, dosage, and time of consumption, and the app will automatically generate a personalized schedule and send timely notifications, even when the device is locked or offline. Using Android's AlarmManager and BroadcastReceiver, the app ensures reminders are triggered accurately and persist until the user confirms they have taken their medication. The app operates offline, storing all medication data locally in an SQLite database, making it particularly useful for users with inconsistent or no internet access. Future versions of the app could integrate cloud services like Firebase for data backup and seamless syncing across devices. The app's user interface is designed to be simple, accessible, and easy to navigate, with large buttons and clear reminders to cater to elderly users, individuals with memory issues, and those on long-term treatments. Medication tracking features allow users to monitor their adherence by providing a history of past doses taken and any missed reminders. To reach a broader demographic, the app will offer multilingual support. Future enhancements could include integration with wearables like smartwatches and fitness trackers to monitor health metrics such as heart rate and blood pressure, linking them to the medication regimen. Voice-activated reminders could be introduced, providing auditory notifications to improve usability for users with visual impairments or those who prefer auditory cues. The app may also allow users to upload prescriptions, automatically populating medication details and generating schedules. Finally, to ensure data security, the app will incorporate strong encryption and comply with privacy regulations such as GDPR or HIPAA to protect sensitive health data.

CHAPTER 4

REQUIREMENTS

4.1 Hardware Requirements

The proposed system is lightweight and optimized for offline or semi-digital retail environments. The following hardware components are recommended:

- **Processor**: A device with a minimum of 1.5 GHz Quad-Core processor or higher for smooth operation and responsiveness.

- **RAM**: At least 2 GB of RAM to support the efficient running of the app along with other background processes.

- **Storage**: A minimum of 50 MB of free storage space for the installation of the app and local database storage. Additional space may be required for storing backup data if cloud integration is implemented.

- **Display**: A screen size of at least 4.5 inches, with a resolution of 720p (HD) or higher, to ensure clear visibility of notifications and medication schedules.

- **Operating System**: Android 5.0 (Lollipop) or higher for compatibility with the app's features and functionality.

- **Battery**: A device with a minimum of 3000mAh battery capacity for uninterrupted app usage, especially for long-duration medication tracking and alarm reminders.

- **Internet Connection**: Optional for cloud synchronization features, but required for updates and backup features (Wi-Fi or mobile data).

4.2 Software Requirements

- **Development Environment**:

- **Android Studio** (latest stable version) for development, debugging, and testing of the application.

- **Programming Language**:
  - **Java** (for Android app development).

- **Database**:
  - **SQLite** for local storage and persistent data management of medication schedules and user details.

- **Android SDK**:
  - The Android Software Development Kit (SDK) is necessary for building the app and testing it on various Android devices.

- **Libraries and Tools**:
  - **AlarmManager** and **BroadcastReceiver** for scheduling and sending notifications.
  - **Google Play Services** for potential future integration features (e.g., Firebase for cloud backup and synchronization).
  - **Retrofit/Volley** (optional) for any HTTP requests, if the app is to be extended with a cloud service or external API integration.
  - **Material Design** library for UI components, ensuring a modern and responsive user interface.
  - **Google Firebase** (optional) for cloud-based data synchronization and backup (future feature).
  - **Gradle** for project automation and dependency management.

- **Version Control System**:
  - **Git** for version control, enabling collaboration and tracking changes in the project.

- **Security**:

Encryption Libraries for data security, ensuring user data privacy and regulatory compliance (such as HIPAA or GDPR).

- **Testing Tools**:

  - **JUnit** for unit testing of Java classes and logic.
  - **Espresso** for UI testing and ensuring a smooth user experience.
  - **Firebase Test Lab** (optional) for real-time device testing and debugging.

CHAPTER 5

SYSTEM DESIGN

5.1 ARCHITECTURE

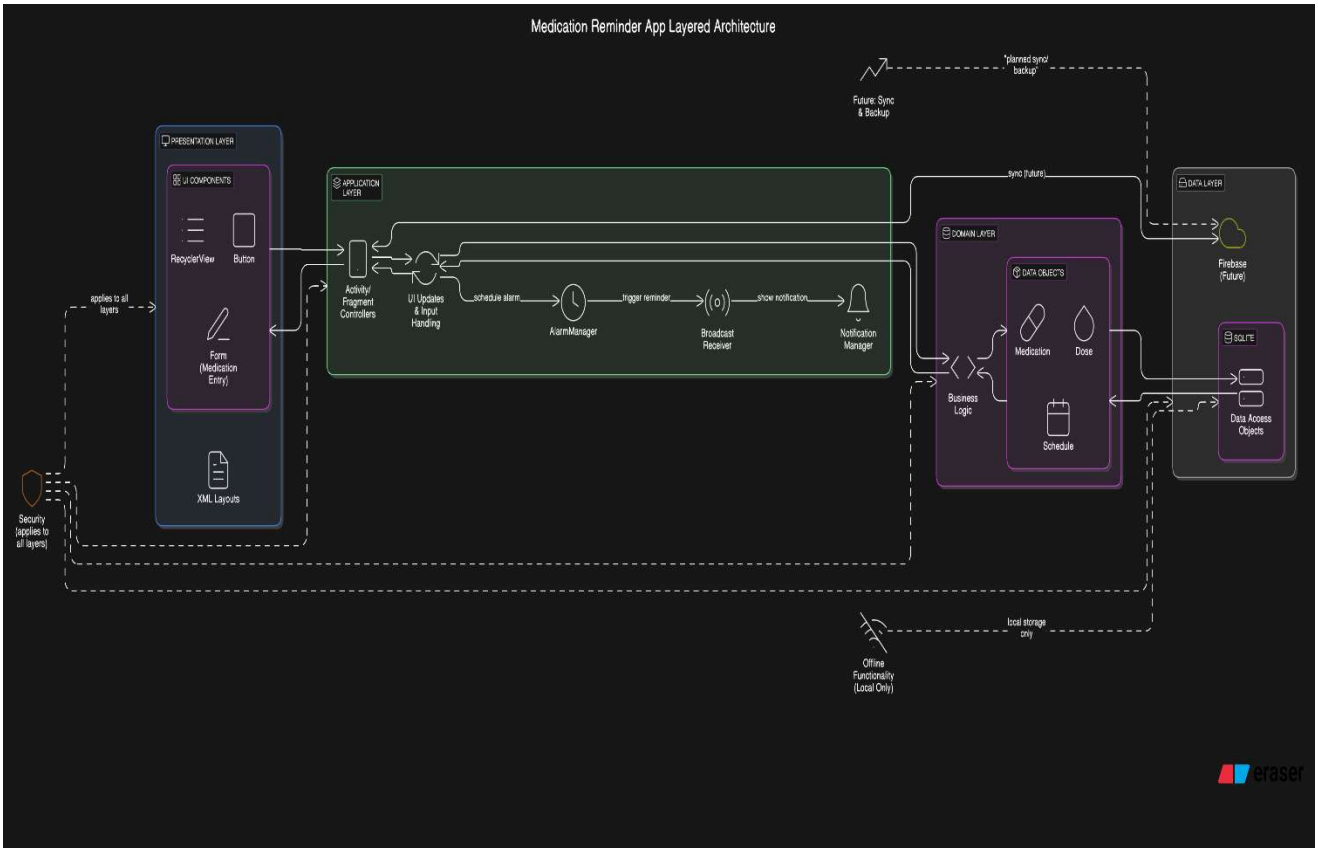

Figure 5.1: Architecture of the Medicine remainder app

The system architecture of the Medication Reminder App in Figure 5.1 is designed around a well-structured, multi-layered model that promotes modularity, offline capability, user-centered functionality, and seamless extensibility for features such as cloud synchronization and wearable device integration. The architecture is divided into five core layers: Presentation Layer, Application Layer, Domain Layer, Data Layer, and Security, each playing a distinct role in ensuring the app's performance, reliability, and maintainability.

The **Presentation Layer** is responsible for rendering the user interface and managing interactions. It comprises UI components such as RecyclerViews for displaying reminder lists, buttons for various user

actions, and dedicated forms for entering medication details. These elements are defined using XML layouts that govern the visual structure and style of the interface. This layer is intentionally crafted for clarity and accessibility, with a particular focus on ease of use for elderly users or those with visual impairments. It captures user inputs effectively and displays medication schedules in a clean, intuitive format.

The **Application Layer** handles the core functionalities of the app and manages user interaction logic. It includes activity and fragment controllers that facilitate screen transitions and maintain lifecycle management. This layer is responsible for updating the user interface based on interactions, processing input data, and coordinating with system services. Key components include the AlarmManager, which schedules medication alerts based on user-defined times, and the BroadcastReceiver, which listens for these scheduled events and triggers appropriate actions. Additionally, the Notification Manager plays a vital role by displaying persistent notifications, ensuring users are reminded of their medication until they confirm intake. This reminder mechanism remains functional even when the device is offline or locked.

The **Domain Layer** serves as the backbone of the app's business logic. It acts as an intermediary between the user interface and the data components, orchestrating application flow and validating medication schedules. Central to this layer are the data objects, including Medication, Dose, and Schedule. These objects represent the primary data structures that the app manipulates and are modeled to ensure the architecture remains clean, consistent, and scalable.

The **Data Layer** governs all aspects of data storage and retrieval. The current implementation uses a local SQLite database, allowing the app to operate fully offline. This database stores information related to medications, dosage details, and scheduling. To facilitate easy and efficient data access, the architecture incorporates Data Access Objects (DAOs), which abstract SQL operations and provide a simplified interface for querying and updating records. Although local storage is the current standard, future development includes integrating Firebase, enabling cloud-based data backup and synchronization. This will allow users to seamlessly recover data or access the app across multiple devices.

The **Security** layer is a cross-cutting aspect integrated throughout the entire architecture. It ensures the privacy and integrity of sensitive medical data by employing encryption mechanisms and adhering to

standards like GDPR and HIPAA. The security model protects both local and future cloud-stored data. As the app evolves, additional measures such as user authentication and secure encrypted backups will be introduced, particularly when Firebase integration is implemented. This architecture reflects a robust and extensible design that supports a responsive and user-friendly medication reminder app. Its offline-first approach ensures reliability in various environments, while the modular design simplifies maintenance and future enhancements. The architecture not only addresses immediate functional needs but also lays the groundwork for advanced features like cross-device synchronization, wearable integration, and multilingual support, making it well-suited for a broad and diverse user base.

5.2 WORKFLOW OF THE MODEL

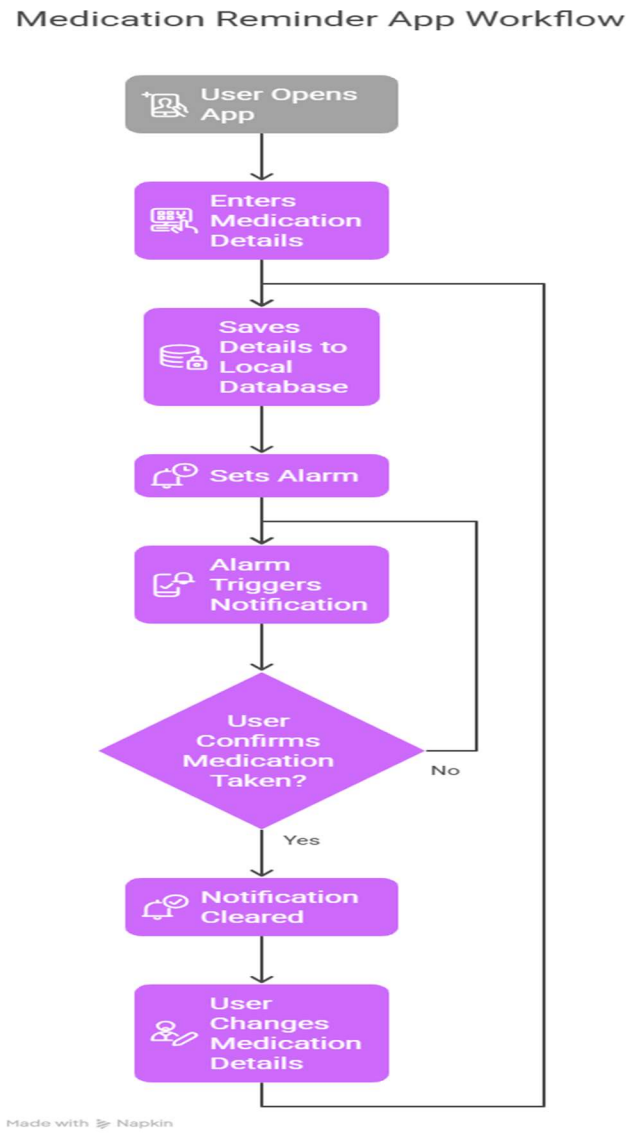

Figure 5.2: Detailed Workflow Medicine remainder app

The diagram in Figure 5.2 illustrates the workflow of the Medication Reminder App follows a logical and user-centric process that begins with user interaction and proceeds through a series of system-managed operations to deliver accurate and timely medication reminders. Upon launching the application, the user is greeted with a simple and accessible interface designed with elderly and visually impaired individuals in mind. From the home screen, users can add new medications by entering

relevant details such as medication name, dosage, frequency, and time schedules through a dedicated medication entry form.

Once the medication information is submitted, the application layer processes the input by validating the data and passing it to the domain layer, where core business logic ensures proper formatting and consistency. The validated data is then passed to the data layer, where it is stored in a local SQLite database. This local storage capability ensures that all features of the app remain fully functional even in offline environments.

After storing the medication schedule, the application registers alarms using the Android AlarmManager. These alarms are configured to trigger based on the times specified by the user. When an alarm goes off, it is intercepted by a BroadcastReceiver, which activates the Notification Manager. The app then displays a persistent notification reminding the user to take their medication. The notification remains active until the user interacts with it, confirming that the medication has been taken. This persistent reminder system plays a critical role in ensuring adherence, especially for users who may miss or ignore standard alerts.

If the user modifies or deletes a medication entry, the app updates the database accordingly and resets or cancels the associated alarms to reflect the changes. The interface consistently reflects the current medication schedule, dynamically updating to display upcoming doses and any changes made by the user.

Throughout the workflow, the app enforces privacy and data integrity using built-in security mechanisms. Future workflow enhancements will include user authentication for data protection and cloud synchronization using Firebase, which will allow users to access their schedules from multiple devices and restore data seamlessly.

In conclusion, the workflow is streamlined to ensure ease of use, reliability, and timely medication reminders, forming a closed loop from user input to notification confirmation. This design empowers users to manage their health independently and confidently while setting the foundation for advanced features like wearable integration and voice-based interaction.

CHAPTER 6
CONCLUSION AND FUTURE ENHANCEMENT

The Medication Reminder App was developed with the primary objective of improving patient adherence to prescribed medication schedules by offering a simple, reliable, and user-friendly mobile solution. Its architecture is thoughtfully layered to ensure modular development, maintainability, and a clear separation of responsibilities. The app's offline-first design, persistent notification system, and intuitive interface make it especially suitable for elderly users and individuals in areas with limited or no internet connectivity. Through the integration of local storage using SQLite, the application manages medication data efficiently, enabling users to store and retrieve schedules without the need for constant internet access. Android components such as AlarmManager, BroadcastReceiver, and NotificationManager work in harmony to ensure timely and uninterrupted reminders, even when the device is idle or locked. These features collectively encourage users to take their medications consistently, promoting better health outcomes and medication compliance. Security has also been considered, with data handling mechanisms designed to preserve user privacy and prepare the app for compliance with healthcare regulations such as GDPR and HIPAA. With its current functionalities, the app stands as a robust, standalone reminder system ready for real-world use. Looking ahead, several enhancements are planned to expand the app's capabilities and reach. Cloud synchronization using Firebase will allow data to be backed up and synced across multiple devices, ensuring users can recover their data after reinstalling or switching devices. To enhance security and personalization, user authentication features such as login and registration will be introduced. The app will also extend support to wearable devices like smartwatches and fitness bands, delivering reminders in more accessible and convenient ways. To further increase its usability, the app will offer multilingual support to cater to non-English speaking users and voice interaction features to aid visually impaired individuals or those preferring hands-free operation. Analytical tools will be incorporated to generate weekly or monthly adherence reports, which can help both users and caregivers monitor medication habits. Additionally, calendar integration will allow users to view and manage medication reminders alongside their daily schedules for better planning and organization.

CHAPTER 7

IMPLEMENTATION AND RESULTS

7.1 OUTPUT SCREENSHOTS

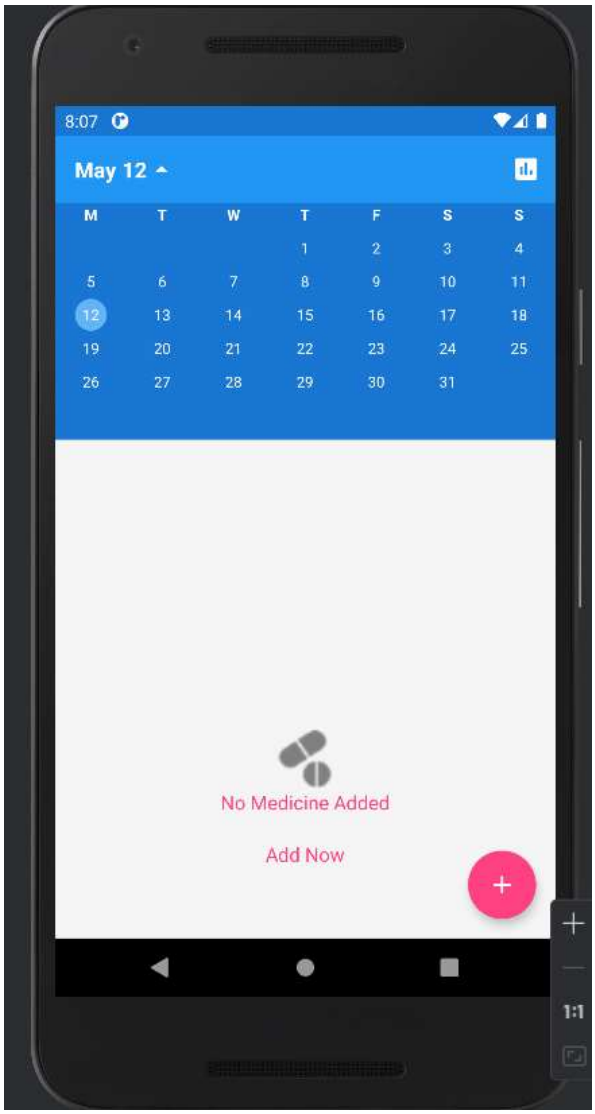

Figure 7.3: HomePage

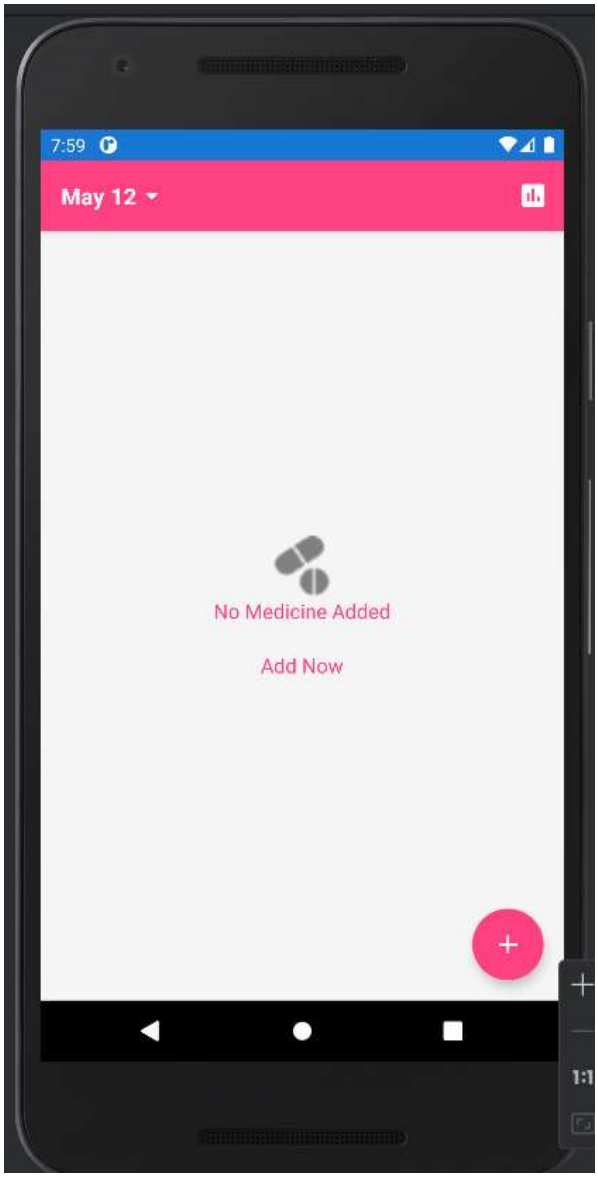

Figure 7.4: Click add button

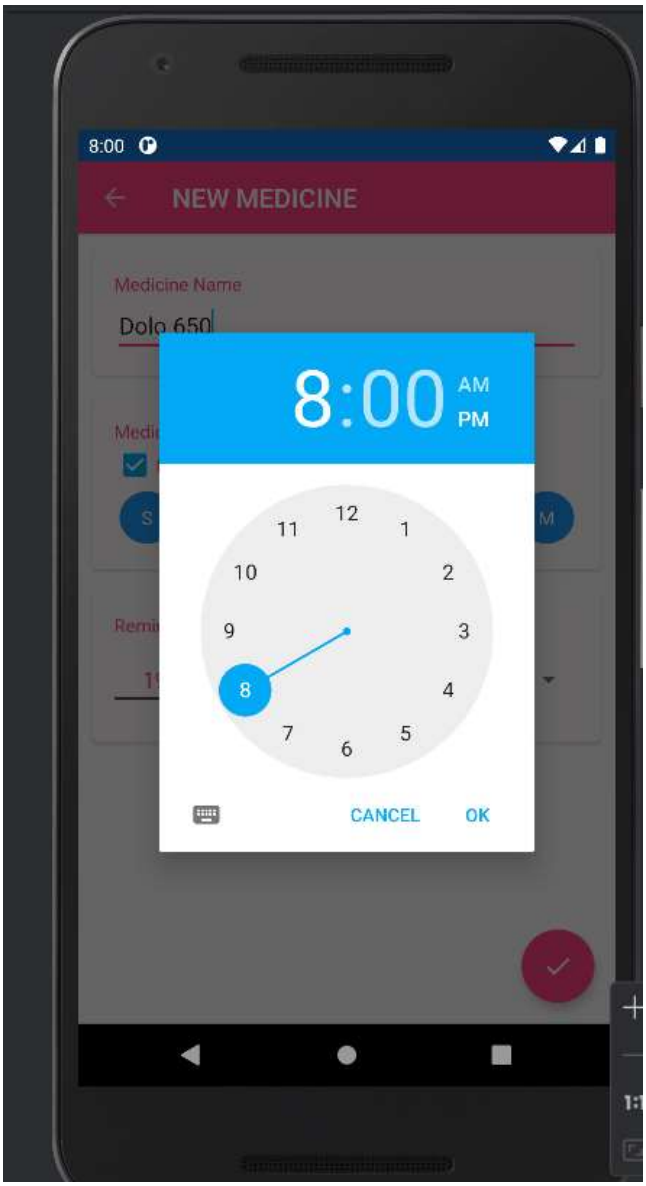

Figure 7.5: Set time to take medicine

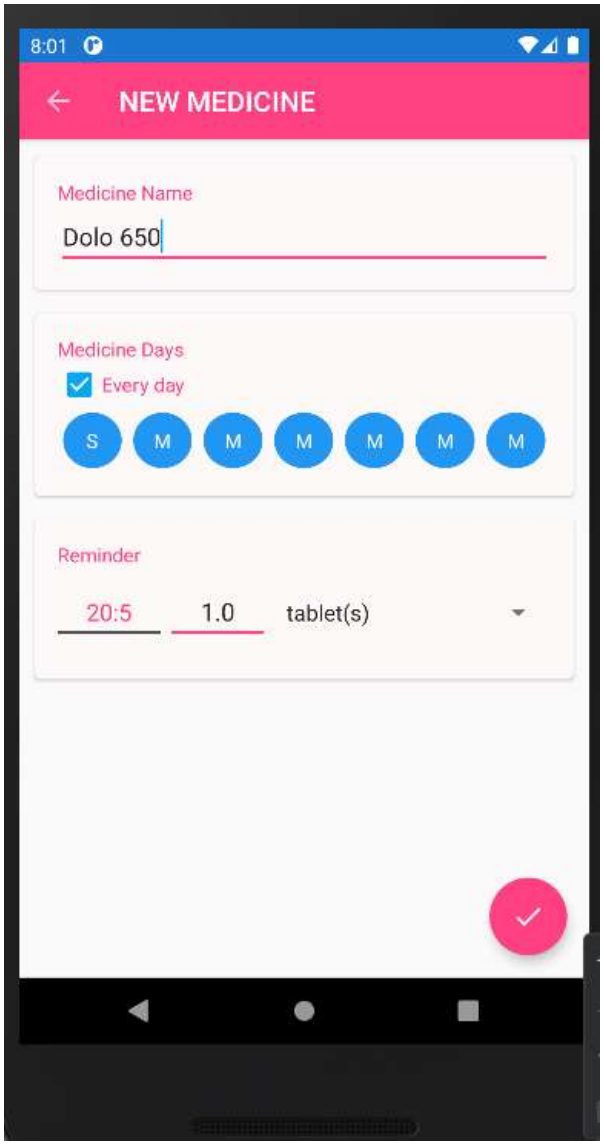

Figure 7.6: Choose medicine days

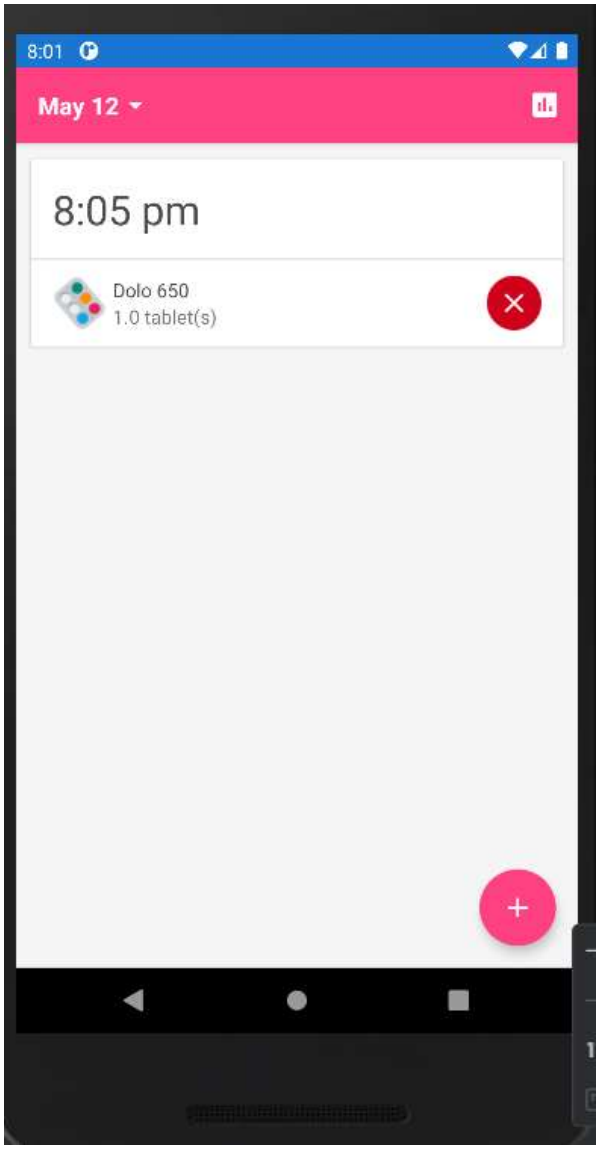

Figure 7.7: Set for alert

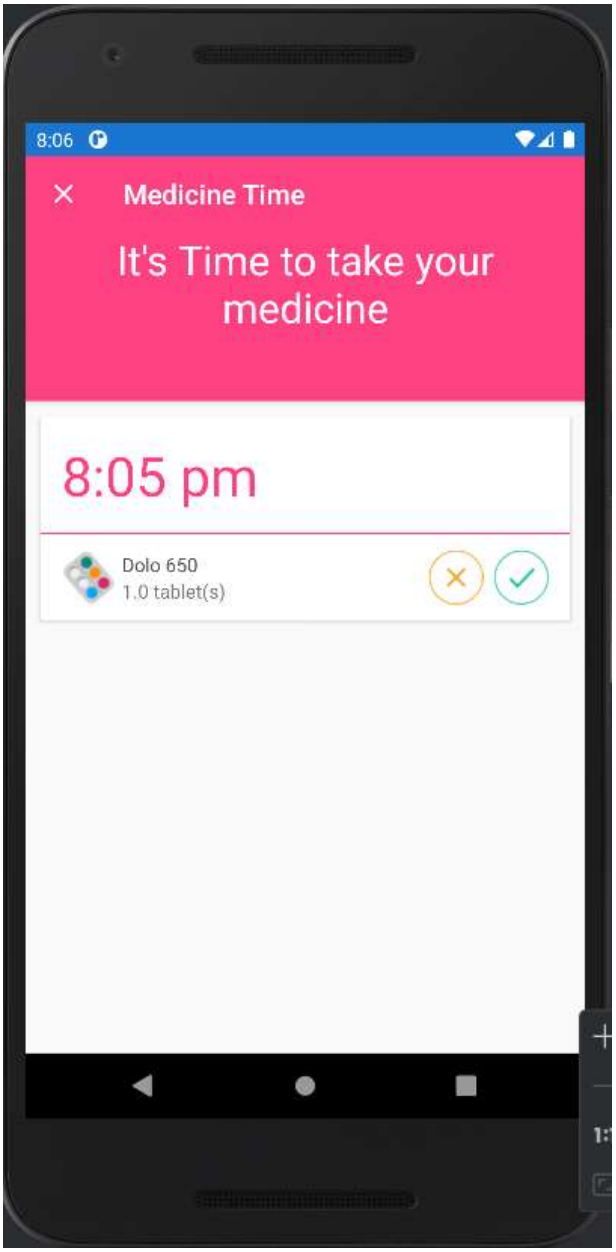

Figure7.8: Alarm ringing

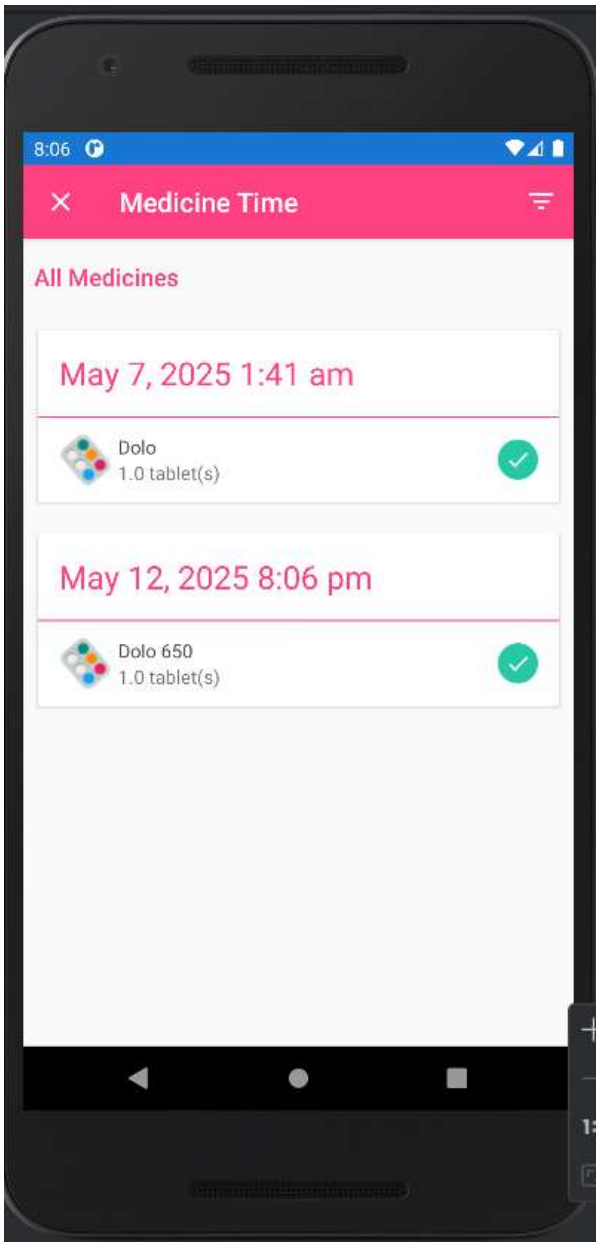

Figure7.9: Showing all medicine

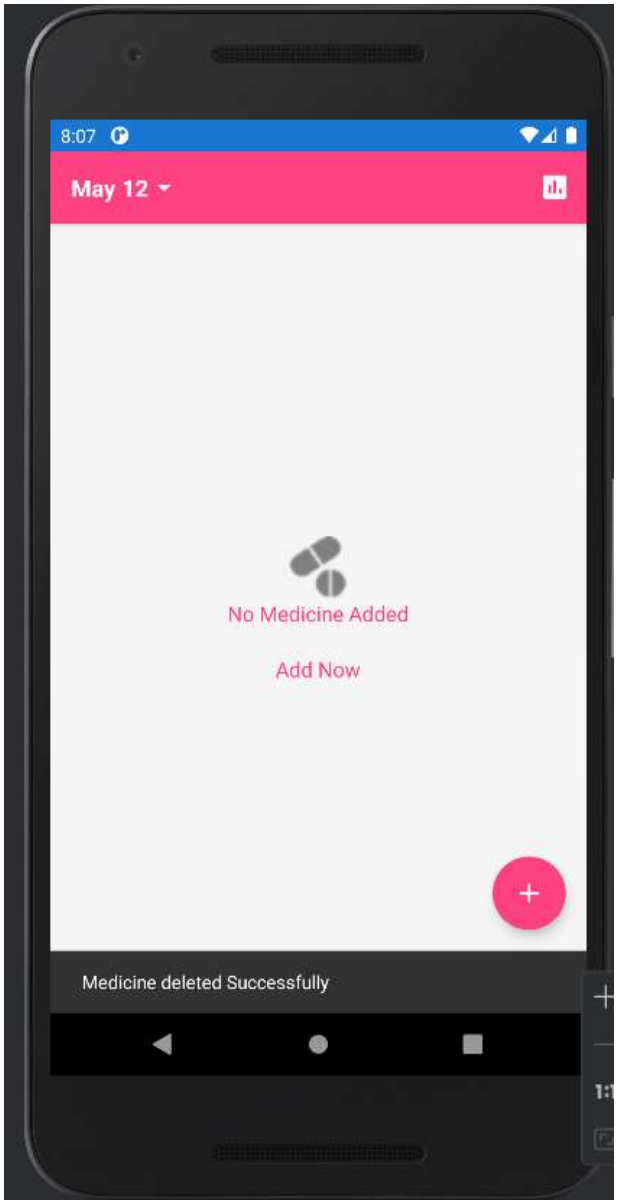

Figure7.10:Medicine deleted

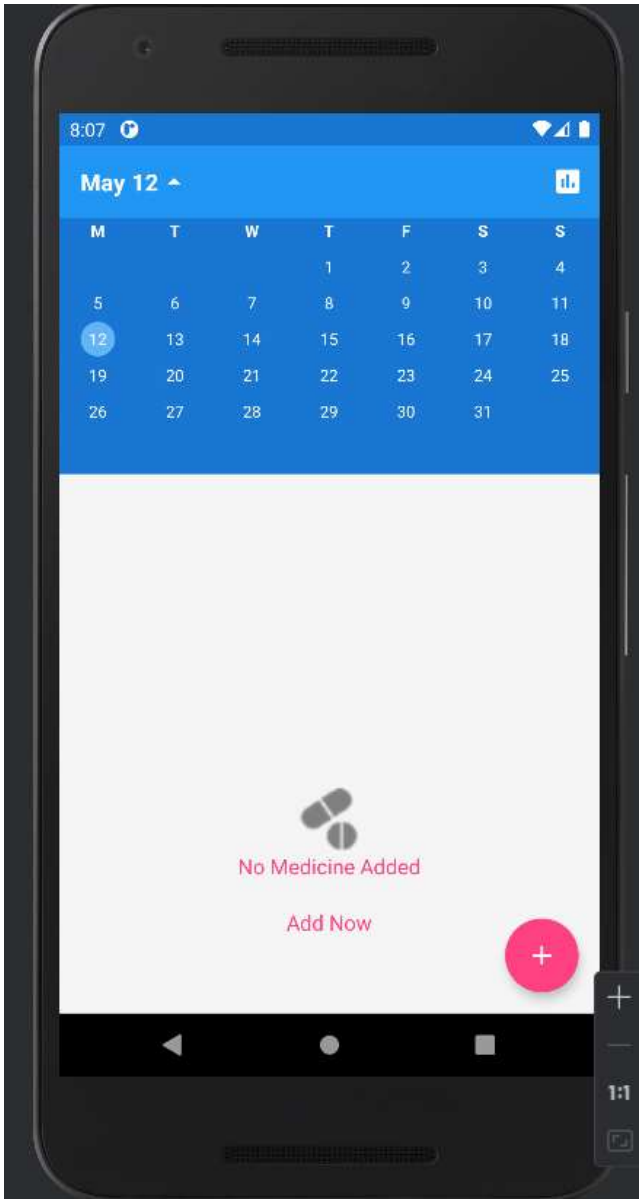

Figure7.11: Add medicine again

CHAPTER 8

REFERENCES

[1] Android Developers, *"Layouts in Android,"* Android Developers, 2025. [Online]. Available: https://developer.android.com/guide/topics/ui/declaring-layout. [Accessed: 12-May-2025].

[2] R. Meier, *Professional Android*, 4th ed., Indianapolis, IN, USA: Wrox, 2018.

[3] Android Developers, *"Material Components for Android,"* Android Developers, 2025. [Online]. Available: https://developer.android.com/jetpack/androidx/releases/compose-material. [Accessed: 12-May-2025].

[4] M. Gargenta and B. Nakamura, *Learning Android*, 3rd ed., O'Reilly Media, 2014.

[5] M. Herath, *Android UI Design Patterns*, Packt Publishing, 2013.

[6] G. Blake, *Mastering Android Layouts: Best Practices for Modern UIs*, Tech Press, 2021.

[7] Google Codelabs, *"Build a Responsive UI with ConstraintLayout,"* Google, 2025. [Online]. Available: https://developer.android.com/codelabs/constraint-layout. [Accessed: 12-May-2025].

[8] Stack Overflow Community, *"Best Practices for Android Toolbar and AppBar,"* Stack Overflow, 2024. [Online]. Available: https://stackoverflow.com/questions/25416948. [Accessed: 12-May-2025].

[9] A. Kothari, *Android Development for Beginners: A Practical Approach*, 2nd ed., Delhi: TechKnowledge Press, 2020.

[10] Google Material Design, *"Typography,"* Material.io, 2025. [Online]. Available: https://m3.material.io/styles/typography. [Accessed: 12-May-2025]