

*A Project report on*

**ENHANCING MEDICATION ADHERENCE THROUGH  
MOBILE APPLICATION**

*Submitted in partial fulfillment of the requirements*

*for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

*in*

**COMPUTER SCIENCE & ENGINEERING**

*By*

**SWATHI REDDY M (204G1A05A9)**

**LEENA SRI A (214G5A0507)**

**SOUMYA C (204G1A05A0)**

**PARIMALA M (204G1A0570)**

Under the Guidance of

**Mr. C. Sudheer Kumar M.Tech., (Ph.D)**

Assistant Professor



**Department of Computer Science & Engineering**

**SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY  
(AUTONOMOUS)**

**Rotarypuram Village, B K Samudram Mandal, Ananthapuramu - 515701**

**2023-2024**

# SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY

(AUTONOMOUS)

(Affiliated to JNTUA, Accredited by NAAC with 'A' Grade, Approved by AICTE, New Delhi &

Accredited by NBA (EEE, ECE & CSE)

Rotarypuram Village, BK Samudram Mandal, Ananthapuramu-515701

## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



### Certificate

This is to certify that the project report entitled **ENHANCING MEDICATION ADHERENCE THROUGH MOBILE APPLICATION** is the bonafide work carried out by **Swathi Reddy M, Leena Sri A, Soumya C, Parimala M** bearing Roll Number **204G1A05A9, 214G5A0507, 204G1A05A0, 204G1A0570** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** during the academic year 2023-2024.

#### **Project Guide**

Mr. C. Sudheer Kumar M.Tech., (Ph.D)

Assistant Professor

#### **Head of the Department**

Mr. P. Veera Prakash M.Tech., (Ph.D)

Assistant Professor

Date:

Place: Rotarypuram

**External Examiner**

## DECLARATION

We, Ms. Swathi Reddy M bearing reg no: 204G1A05A9, Ms. Leena Sri A bearing reg no: 214G5A0507, Ms. Soumya C bearing reg no: 204G1A05A0, Ms. Parimala M bearing reg no: 204G1A0570, students of **SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY**, Rotarypuram, hereby declare that the dissertation entitled “**ENHANCING MEDICATION ADHERENCE THROUGH MOBILE APPLICATION**” embodies the report of our project work carried out by us during IV Year Bachelor of Technology under the guidance of **Mr. C. Sudheer Kumar**, M.Tech., (Ph.D) Assistant Professor, Department of CSE and this work has been submitted for the partial fulfillment of the requirements for the award of Bachelor of Technology degree.

The results embodied in this project report have not been submitted to any other Universities of Institute for the award of Degree.

SWATHI REDDY M

Reg no: 204G1A05A9

LEENA SRI A

Reg no: 214G5A0507

SOUMYA C

Reg no: 204G1A05A0

PARIMALA M

Reg no: 204G1A0570

## **VISION & MISSION OF THE INSTITUTION**

### **Vision:**

To become a premier Educational Institution in India offering the best teaching and learning environment for our students that will enable them to become complete individuals with professional competency, human touch, ethical values, service motto, and a strong sense of responsibility towards environment and society at large.

### **Mission:**

- Continually enhance the quality of physical infrastructure and human resources to evolve in to a center of excellence in engineering education.
- Provide comprehensive learning experiences that are conducive for the students to acquire professional competences, ethical values, life-long learning abilities and understanding of the technology, environment and society.
- Strengthen industry institute interactions to enable the students work on realistic problems and acquire the ability to face the ever changing requirements of the industry.
- Continually enhance the quality of the relationship between students and faculty which is a key to the development of an exciting and rewarding learning environment in the college.

## **VISION & MISSION OF THE DEPARTMENT OF CSE**

### **Vision:**

To evolve as a leading department by offering best comprehensive teaching and learning practices for students to be self-competent technocrats with professional ethics and social responsibilities.

### **Mission:**

**DM 1:** Continuous enhancement of the teaching-learning practices to gain profound knowledge in theoretical & practical aspects of computer science applications.

**DM 2:** Administer training on emerging technologies and motivate the students to inculcate self-learning abilities, ethical values and social consciousness to become competent professionals.

**DM 3:** Perpetual elevation of Industry-Institute interactions to facilitate the students to work on real-time problems to serve the needs of the society.

## ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we have now the opportunity to express our gratitude for all of them.

It is with immense pleasure that we would like to express our indebted gratitude to our Guide **Mr. C. Sudheer Kumar, Assistant Professor, Computer Science & Engineering**, who has guided us a lot and encouraged us in every step of the project work. We thank him for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

We express our deep-felt gratitude to **Mr. C. Lakshminatha Reddy, Assistant Professor** and **Mr. M. Narasimhulu, Assistant Professor**, Project Coordinators for their valuable guidance and unstinting encouragement enabled us to accomplish our project successfully in time.

We are very much thankful to **Mr. P. Veera Prakash, Assistant Professor & Head of the Department, Computer Science & Engineering** for his kind support and for providing necessary facilities to carry out the work.

We wish to convey our special thanks to **Dr. G. Balakrishna, Principal of Srinivasa Ramanujan Institute of Technology** for giving the required information in doing our project work. Not to forget, We thank all other faculty and non-teaching staff, and our friends who had directly or indirectly helped and supported us in completing our project in time.

We also express our sincere thanks to the Management for providing excellent facilities.

Finally, we wish to convey our gratitude to our families who fostered all the requirements and facilities that we need.

### Project Associates

204G1A05A9

214G5A0507

204G1A05A0

204G1A0570

## **ABSTRACT**

Human forgetfulness poses a significant challenge in the context of medication adherence, leading to adverse health consequences and, in some instances, premature deaths. Recognizing the importance of addressing this issue, our project involves the development of a specialized mobile application aimed at improving medication management. This innovative application is designed to streamline the medication adherence process by seamlessly integrating doctor's prescriptions with patient's daily routines.

The core objective of the proposed mobile application is to provide automatic medication reminders. Through this platform, healthcare professionals can prescribe medications directly, with the data seamlessly synchronized in real-time with the patient's interface. This dynamic approach ensures that patients receive accurate and timely updates regarding their medication schedules, promoting a proactive and informed approach to treatment. The proposed mobile application serves as a reliable tool for patients. By using technology to connect the prescription with the patient's adherence, our suggested solution aims to reduce the harmful health effects that are linked to medication non-adherence.

**Keywords:** Mobile Application, Medication Adherence, Reminder System, Automatic alarm.

# CONTENTS

	<b>Page No.</b>
<b>List of Figures</b>	ix
<b>List of Abbreviations</b>	x
<b>List of Tables</b>	xi
<b>Chapter 1      Introduction</b>	1
1.1    Problem Definition	1
1.2    Objective of the Project	1
<b>Chapter 2      Literature Survey</b>	2-3
<b>Chapter 3      Planning</b>	4-9
3.1    Existing System	4
3.2    Disadvantages of Existing System	4
3.3    Proposed System	4-5
3.4    H/W & S/W Requirements	
3.4.1    Hardware Requirements	5
3.4.2    Software Requirement Specification	5-6
3.4.3    Software Requirements	6
3.5    Functional Requirements	6-7
3.6    Non - Functional Requirements	7
3.7    Modules	7-8
3.8    Work Schedule	8-9
<b>Chapter 4      Design</b>	10-26
4.1    Android OS & Architecture	10
4.1.1    A Brief History of Android	10-11
4.1.2    The Android Platform	11-13
4.1.3    Android Runtime Architecture	13
4.2    Flutter	14
4.3    Dart	14
4.4    Firebase	14
4.5    Download and Install Visual Studio Code	15-16
4.6    Exploring Visual Studio Code	16-17

	4.7 Flutter Installation	17-19
	4.8 Adding Firebase to Flutter	19-21
	4.9 ER/UML Diagrams	21
	4.9.1 Data Flow Diagram	21-22
	4.9.2 Use Case Diagram	22-23
	4.10 Software Model Selection	23-25
	4.11 Architecture	25-26
<b>Chapter 5</b>	<b>Implementation</b>	27-30
	5.1 main.dart	27-28
	5.2 pubspec.yaml	28-29
	5.3 Firebase cloud Firestore	30
<b>Chapter 6</b>	<b>Output Screens</b>	31-34
	6.1 Home screen	31
	6.2 Patient Details form	31
	6.3 Patient ID Generation	32
	6.4 Doctor Login page	32
	6.5 Prescribe Medicines	33
	6.6 Patient Login	33
	6.7 Patient Home	34
<b>Chapter 7</b>	<b>Testing</b>	35-38
	7.1 Unit Testing	35
	7.2 Integration Testing	35-36
	7.3 Functional Testing	36-37
	7.4 Validation Testing	37
	7.5 Acceptance Testing	38
	7.6 Test Results	38
	<b>Conclusion and Future Work</b>	39
	<b>References</b>	40
	<b>Publication Paper</b>	
	<b>Participation Certificates</b>	



## LIST OF FIGURES

<b>Fig. No.</b>	<b>Figure Name</b>	<b>Page No.</b>
4.1	Android Software layers	12
4.2	Android Runtime Architecture	13
4.3	Flutter and Firebase	14
4.4	Visual Studio Code	17
4.5	Report of flutter doctor	18
4.6	Data flow diagram	22
4.7	Use case diagram	23
4.8	Agile process model – Scrum framework	25
4.9	Application Architecture	26
5.1	main.dart	28
5.2	pubspec.yaml	29
5.3	Cloud Firestore	30
6.1	Home screen	31
6.2	Patient details form	31
6.3	Patient ID generation	32
6.4	Doctor Login page	32
6.5	Prescribe medicines	33
6.6	Patient Login	33
6.7	Patient Home screen	34

## **LIST OF ABBREVIATIONS**

VS Code	Visual Studio Code
UI	User Interface
UML	Unified Model Language
SRS	Software Requirement Specifications
DFD	Data Flow Diagram
SDK	Software Development Kit
IDE	Integrated Development Environment
YAML	Yet Another Markup Language
ID	Identification
OS	Operating System

## **LIST OF TABLES**

<b>Table No.</b>	<b>Table Name</b>	<b>Page No.</b>
3.1	Work schedule	8
3.2	Time division for Implementation	9

# **CHAPTER 1**

## **INTRODUCTION**

Today's Human life is full of responsibilities and stress which affects our health and is prone to diseases of different types. Hence, it is our responsibility to make ourselves stay fit and lead a healthy life. It is important to address these health issues proactively, supporting well-being and fitness. To meet this need, developing a mobile application that serves as a vital tool in healthcare management is important.

This innovative application empowers doctors to prescribe medications seamlessly, incorporating a feature for setting automatic reminders directly synchronized with the patient's interfaces. Moreover, recognizing the importance of familial support in healthcare, the platform allows patients to include family members. This inclusion facilitates family members in reminding patients to adhere to their medications. With a primary focus on individuals prone to forgetting medication schedules, the proposed mobile application helps to enhance medication adherence in the middle of our fast-paced lives.

### **1.1 Problem Definition**

Despite the availability of numerous reminder applications, patients occasionally encounter confusion regarding the administration of medication and certain instances of incorrectly set reminders may lead to negative health outcomes and, in extreme cases, premature deaths. To tackle this issue head-on, there is a compelling need to develop a specialized mobile application designed exclusively for managing medications. This application should not only serve as a reminder tool but also offer a seamless platform for doctors to prescribe medications directly through the app. The medication adherence can be improved which helps people lead a happy and healthy life.

### **1.2 Objective of the Project**

The main objective is to help patients take their medications on time with the help of medicine reminders and also provide an interface where doctors can directly prescribe the medications which automatically gets updated in the patient Mobile Application.

## **CHAPTER 2**

### **LITERATURE SURVEY**

Ms. S. A. Patil [1] focuses to develop Android application for patients to remind medicines and health care. Patient must fill out a form first with their name, mobile number, email address, gender, and password. Patient can log in using provided email address or mobile number in an application. However, the patient must be aware that the proper mobile number, email address, and password must be entered. It also describes about the various advantages and disadvantages of the applications.

In this application there are two important modules. First patient will get registration form. In that form patient need to enter name, mobile number, email id, gender and password. Patient need to understand that correct mobile number, email id and alphanumeric password should enter. After filling this information patient need to click on register button. In this module, there is calendar to select date and there is time to set time and there is textbox to add medicines. After set alarm module patient will get notification by email/mobile notification.

Samir V. Zanjali [2] Home based health care arrangement include communications, imaging, sensing and human computer interaction technologies. The monitoring system uses the self-powering wireless environment with the help of renewable energy. The data from transmitter will be sending to cloud for centralized monitoring takes place, the expert in remote place can view all patient data and in case of emergency can take appropriate action. Buying all the equipment increases our budget. To address these challenges, we are proposing a cost effective mobile application which efficiently delivers pill reminders and increase patient medication adherence.

KeeHyun Park [3] In this paper, Park proposed a medication reminder synchronization system that provides a patient with medications as prescribed by medical staff. In addition, medical staff can remotely send messages to the system in order to change the medication schedules or device configuration settings embedded in the medication reminder. The system supports the OMA DS protocol, which was originally proposed as a DS standard protocol that synchronizes data between mobile devices and a central server.

Lindsey Dayer [4] Medication adherence usually refers to whether patients take their drugs as prescribed and whether they continue to take a prescribed medication. Medication non-adherence is an increasing concern for clinicians, healthcare systems, and other stakeholders because to mounting evidence that it is widespread and associated with negative outcomes and greater healthcare expenditures.

This paper aims to examine the following topics: (1) various approaches to assessing adherence (2) medication non-adherence prevalence (3) non-adherence and outcomes correlation (4) non-adherence causes and (5) treatments to enhance medication adherence.

Bob Tedeschi [5] suggests that beyond the adage an apple a day, additional reminders for medication adherence are crucial for optimal health outcomes. Studies highlight the efficacy of various strategies such as smart phone applications, electronic pillboxes, text message reminders, and personalized medication schedules. These interventions not only improve adherence rates but also enhance patient engagement and satisfaction, ultimately leading to better disease management and healthcare utilization.

## CHAPTER 3

### PLANNING

#### 3.1 Existing System

Medication management remains a common challenge despite the proliferation of reminder applications. Patients often face confusion, leading to instances of incorrectly set reminders and potential lapses in medication adherence. The lack of a dedicated interface for doctors exacerbates this issue, as existing applications may not provide a platform for healthcare professionals to establish comprehensive reminders, including detailed dosage instructions.

Incorrect timing of medication intake further compounds the challenge, impacting the effectiveness of the treatment plan. Inadequate communication channels between patients and healthcare providers contribute to misunderstandings and hinder the resolution of medication-related concerns.

#### 3.2 Disadvantages of Existing System

- User Error in setting Reminders.
- May not be as discoverable as mobile apps as they are not listed in a specific database, such as the app store.
- Quality and security is not always guarantee - web apps don't need to be approved by the app store.

#### 3.3 Proposed System

To resolve the issues in the existing system our mobile application introduces a seamless integration between healthcare and technology by offering an interface that enables doctors to directly issue prescriptions linked to patient IDs. This integration ensures synchronization with the patient application, triggering automated reminders for timely medication intake.

By aligning reminders with prescribed medications, the system eliminates ambiguities, providing patients with precise dosage instructions and schedules. The application incorporates features allowing patients to mark medicines as taken, cancel doses, or provide feedback, enhancing medication tracking and adherence.

**The proposed system contains the following facilities over the existing system**

- Automated reminders based on direct prescription integration enhance adherence by reducing the chances of missed doses.
- The application establishes a direct and efficient communication channel between doctors and patients, promoting clearer instructions and feedback.
- Features like marking medicines as taken and cancelling doses provide real-time insights into patients' medication intake, aiding healthcare providers in monitoring progress.
- Automation of reminders and direct prescription issuance saves time for both healthcare providers and patients, streamlining the prescription and adherence process.

This proposed system allows not only streamlines the healthcare process but also establishes a direct and efficient communication channel between healthcare providers and patients. This approach reduces the likelihood of medication-related errors and misunderstandings, fostering a more accurate and patient-centric approach to healthcare.

By merging technology and healthcare, this application establishes a holistic solution for medication adherence, ultimately contributing to improved patient outcomes and a more efficient healthcare ecosystem.

**3.4 H/W & S/W Requirements****3.4.1 Hardware Requirements**

- RAM : 8GB and above
- Hard Disk : 1TB
- Processor : i5 and above

**3.4.2 Software Requirement Specification**

SRS is a document created by a system analyst after the requirements are collected. SRS defines how the intended software will interact with hardware, external interfaces, speed of operation, response time of system, portability of software across various platforms, maintainability, speed of recovery after crashing, Security, Quality, Limitations etc.



The requirements received from clients are written in natural language. It is the responsibility of system analysts to document the requirements in technical language so that they can be comprehended and useful by the software development team.

The SRS is a specification for a specific software product, program, or set of applications that perform particular functions in a specific environment. It serves several goals depending on who is writing it. First, the SRS could be written by the client of a system. Second, the SRS could be written by a developer of the system.

The two methods create entirely various situations and establish different purposes for the document altogether. The first case, SRS, is used to define the needs and expectations of the users. The second case, SRS, is written for various purposes and serves as a contract document between customer and developer.

### **3.4.3 Software Requirements**

- IDE : Visual Studio Code
- Backend : Firebase
- Programming Language : Dart
- SDK : Flutter
- Operating System : Windows OS

### **3.5 Functional Requirements**

- Including both doctors and patients, should have secure and reliable authentication mechanisms to access the application.
- A feature allowing admins to register doctors in the system, associating each doctors email and password.
- After registrations patients get the unique Id.
- Capability for doctors to issue digital prescriptions directly through the application, associating them with patient IDs.
- A system that automatically generates and sends reminders to patients based on the prescribed medication schedules.
- Ensuring seamless synchronization between the doctor's interface and the patient application to maintain accurate and up-to-date information.

- Inclusion of detailed dosage instructions and medication schedules associated with each prescription to eliminate ambiguities.

### **3.6 Non - Functional Requirements**

- Security
- Performance
- User - friendly
- Maintainability
- Reliability
- Fault Tolerance
- Accessibility

### **3.7 Modules**

In this project, the system after a careful analysis has been identified to be presented with the following modules.

#### **User Registration and Authentication**

- This module manages user registration for both doctors and patients.
- Patients provide their personal information including phone number, name, age, and Aadhar number during registration.
- Upon registration, patients are assigned a unique Patient ID.
- Authentication mechanisms ensure secure access to the system for registered users.

#### **Doctor Interface**

- This module provides a user interface for doctors to manage medication reminders and prescriptions.
- Doctors can log in using their credentials and access features to prescribe medications and set reminders for patients.
- It includes options to search for patients by their Patient IDs and add family members for monitoring.

#### **Medication Prescribing and Reminder Setting**

- This module enables doctors to prescribe medications and set reminder schedules.
- Doctors can link prescriptions to patient IDs, specifying medication details and dosage instructions.

- Reminder settings are directly integrated with prescription issuance, ensuring that patients receive timely notifications.

### **Patient Mobile Application**

- This module provides a mobile application interface for patients to receive medication reminders and manage prescriptions.
- Patients log in using their credentials or Patient IDs assigned during registration.
- The application syncs with the doctor's interface to display prescribed medications and reminder schedules.

### **Reminder System**

- This module generates automated medication reminders based on the prescribed schedules.
- Patients receive notifications on their mobile devices for medication intake times and dosage instructions.
- Customizable features such as snooze reminders and timestamp options enhance user experience and adherence.

By integrating these modules, the mobile application facilitates seamless communication between doctors and patients, streamlines medication management processes, and enhances medication adherence for improved health outcomes.

## **3.8 Work Schedule**

Table 3.1: Work Schedule

<b>S. No.</b>	<b>Development Stage</b>	<b>Duration</b>
1.	Domain and Title Selection	1 week
2.	Literature Survey and Problem Definition	1 week
3.	Planning	1 week
4.	Design	2 weeks

5.	Requirements Gathering	1 week
6.	Testing	2 weeks
7.	Implementation	4 weeks

Table 3.2: Time Division for Implementation

Module Name	Time estimated
1. User Registration and Authentication	1 week
2. Doctor Interface	1 week
3. Medication Prescribing	
4. Patient Mobile Application	1 week
5. Reminder System	1 week

## CHAPTER 4

### DESIGN

#### 4.1 Android OS & Architecture

Android is a popular computing platform based on the Linux operating system. The initial commercial version of Android hit the market in 2008 in the form of a mobile phone platform, back when the most popular cell phone for a business user was the BlackBerry, when the iPhone was beginning to make meaningful waves across all sectors, and when the majority of phone users were still tapping out texts from a flip phone.

In 10 year's time, Android has effectively become the world's most popular operating system by a number of measures. Despite the robust popularity of the flashy and capable Apple iPhone platform, Android shipments worldwide meaningfully outpace Apple's offerings. While Apple's devices continue to demand an ever increasing price point, Android devices scale the global marketplace. Yes, there are super pricey Android models sitting next to the latest iPhone, but there are also relatively low-cost Android phones and tablets available for sale at Walmart and on Amazon. As Android has matured, it is finding its way into a variety of devices, including televisions, projectors, automobiles, and even recreational vehicles. Want to dim the lights in your camper or activate the awning? You can use the Android-based touch screen interface to manipulate the controls. Or, use your smartphone equipped with Bluetooth to communicate with the RV's Android-based control system. There are many of these types of interfaces finding their way to the market. Some user experiences are simply fantastic (like drone controllers), and some are less than fantastic, like the controls in my uncle's RV. Introducing the Android platform and discussing how you can use it for both mobile and non-mobile applications. The ambition is to get you on a path to making awesome apps for whatever platform arena you feel called to make your contribution.

##### 4.1.1 A Brief History of Android

The Android platform was spawned from the efforts of an organization known as the Open Handset Alliance (OHA), which had at the outset the mission of collaborating to "build a better mobile phone." Today, if you visit the Open Handset

Alliance website, you might think that their efforts were in vain, as the latest news item dates to 2011. However, what was started with that group of carriers, hardware manufacturers, and software vendors has grown into the world's most popular platform. Is Google the "man behind the curtain" of the OHA? Maybe, but no matter. In the decade since its introduction, Android is offered in the market by many large players across the globe and across numerous industries. Samsung, also a member of the OHA, is the leading manufacturer of smartphone devices worldwide, thanks to Android. Although a single device (the G1 device manufactured by HTC and provisioned on the TMobile network) started it all, Android devices are now available in virtually every market on the planet — not just for mobile phones. It is beyond the scope of this project, but ask yourself if there is not a correlation between (arguably) the world's most successful Internet/search company also being the driving force behind the world's most popular mobile platform. More eyeball's view Android devices every day worldwide than any other single computing platform. If you want to write code that can run literally anywhere in the world, then you need to learn about the Android platform.

#### **4.1.2 The Android Platform**

Looking at Android's breadth of capabilities, it would be easy to confuse it with a desktop operating system. Android is a layered environment, one that is built upon a foundation of the Linux kernel and includes rich functionality. The user interface subsystem includes everything you would expect from a mature operating system environment including windows, views, and widgets for displaying common elements like edit boxes, lists, or drop-down lists. The browser is both capable for general web browsing and available for embedding directly into your own application. It is beyond the scope of this project, but ask yourself if there is not a correlation between (arguably) the world's most successful Internet/search company also being the driving force behind the world's most popular mobile platform. More eyeball's view Android devices every day worldwide than any other single computing platform.

In the past decade, the mobile web has been transformed by the adoption of smartphones across consumer and business applications, including Android. "Responsive" web technologies have made the utility of mobile devices greatly

enhanced. The user interface subsystem includes everything you would expect from a mature operating system environment including windows, views, and widgets for displaying common elements like edit boxes, lists, or drop-down lists. The browser is both capable for general web browsing and available for embedding directly into your own application. Android software contains many layers as mentioned in the below Fig. 4.1.

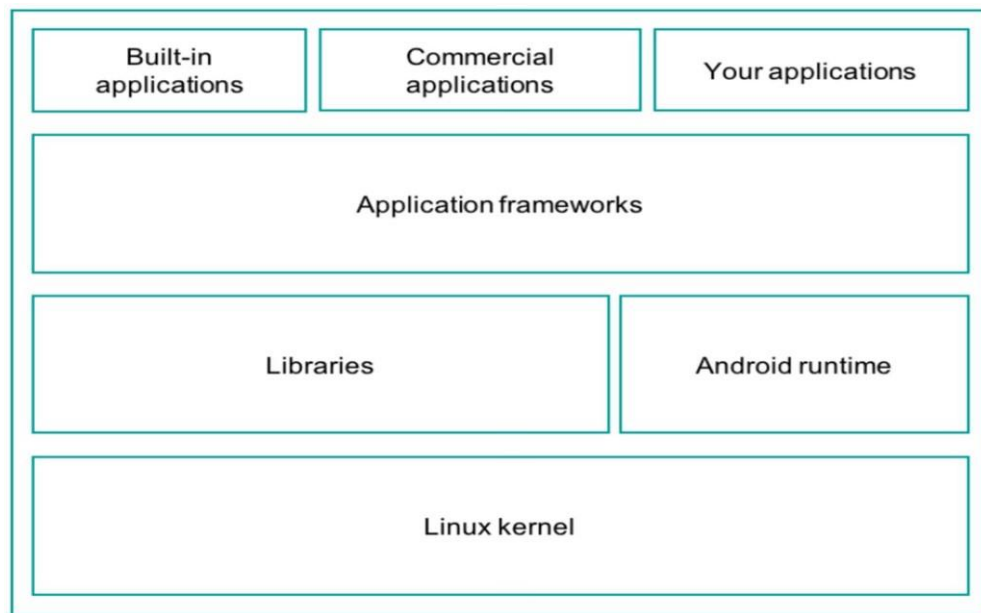


Fig. 4.1: Android Software Layers

**Applications:** Built-in applications, such as phone, contacts, browser, and more. The specific applications vary by Android version and manufacturer. Commercial applications from marketplaces, such as google play, amazon and more. Side- loaded applications, including the ones you will build. You install these via USB cable.

**Application frameworks,** such as telephony manager, location manager, notification manager, content providers, windowing, resource manager, and more. **Libraries,** such as graphics libraries, media libraries, database libraries, sensors, and so on. The **Android runtime** is responsible for executing and managing applications as they run. **Linux, Kernel,** including power, file system, drivers, process management and more.

Android boasts a healthy array of connectivity options including wifi, Bluetooth, NFC, and of course cellular connections on every network that you can imagine. Location- based services power popular mapping and navigation apps.

Digital cameras have essentially retreated up-market due to the quality of the cameras found in modern smartphones. Android supports multiple cameras with the ability to capture full motion video. One of my favourite classes of applications is machine vision, where applications use the camera as an input device to perform inspections for manufacturing. Voice-based services make the modern Android device a virtual personal assistant. If you can dream it, you can program it on the Android platform! Clearly, Android is a capable computing platform, so let's see what kinds of applications Android can run.

#### 4.1.3 Android Runtime Architecture

An important aspect of the Android application environment is that Android applications have historically been written in the Java programming language. Fig. 4.2 represents the architecture of Android runtime. However, you can also write them in a relatively new programming language from Google called Kotlin. Between the trend toward "newer is better" in programming languages (for example, Swift is overtaking Objective-C for iOS/Apple development) and the ongoing legal battle over where Java can or cannot be used, Kotlin will likely be the leading language for Android within a few years. But as of now Java is ruling the android native development with great support and more over great fanbase.

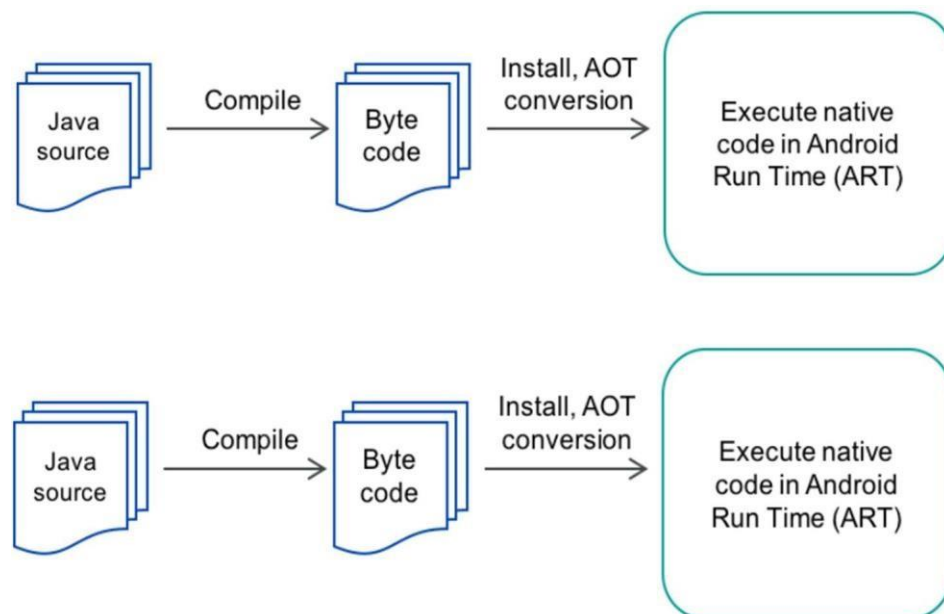


Fig. 4.2: Android Runtime Architecture



## 4.2 Flutter

Flutter is an open-source UI software development kit created by Google. It is used to develop applications for Android, iOS, Linux, Mac, Windows, and the web from a single codebase by using DART language.

### Characteristics

- Fast Development
- Expressive and Flexible UI
- Native Performance

## 4.3 Dart

Dart is a programming language developed by Google that is used to build mobile, desktop, server, and web applications. Dart is an object-oriented, class-based, garbage-collected language with C-style syntax. Dart can compile to either native code or JavaScript. It supports interfaces, abstract classes, and type inference.

## 4.4 Firebase

Firebase is a set of backend cloud computing services and application development platforms provided by Google. It hosts databases, services, authentication, and integration for a variety of applications, including Android, iOS, JavaScript, Node.js, Java, Unity, PHP, and C++.Flutter and Firebase (Fig. 4.3) work together seamlessly.



Fig. 4.3: Flutter and Firebase

## 4.5 Download and Install Visual Studio Code

### Step 1:

Open your web browser and navigate to the official Visual Studio website. You can do this by typing "Visual Studio" into the search bar and selecting the official Microsoft link.

### Step 2:

On the Visual Studio website, you'll see different editions available for download, such as Visual Studio Community, Visual Studio Professional, and Visual Studio Enterprise. Click on the edition that suits your needs. For most users, Visual Studio Community is free and suitable for individual developers and small teams.

### Step 3:

Once you've selected the edition, you'll be directed to a page where you can download the Visual Studio Installer. Look for the "Download" button and click on it to download the installer file.

### Step 4:

After the installer file has finished downloading, locate the file in your computer's Downloads folder or wherever you saved it. Double-click on the installer file to run it. You may need to grant administrative permissions to proceed with the installation.

### Step 5:

The Visual Studio Installer will launch and present you with various installation options. You'll see a list of workloads and individual components that you can choose to install. Select the workloads and components that are relevant to your development needs. You can also choose to install additional components later if needed.

### Step 6:

Once you've customized your installation options, click on the "Install" button to begin the installation process. The installer will download and install the selected components from the internet. The time required for installation may vary depending on your internet speed and the components selected.

**Step 7:**

During the installation process, you'll see a progress bar indicating the status of the installation. Once all selected components are downloaded and installed, the installation process will complete, and you'll see a confirmation message.

**Step 8:**

After the installation is complete, you can launch Visual Studio from the Start menu (on Windows) or the Applications folder (on macOS). When you run Visual Studio for the first time, you may need to sign in with your Microsoft account or create a new account if you don't have one already.

**Step 9:**

Upon launching Visual Studio, you may be prompted to customize your development environment settings, such as theme, keyboard shortcuts, and extensions. You can configure these settings according to your preferences.

**Step 10:**

Once Visual Studio is set up, you're ready to start coding! Create a new project or open an existing one to begin your development journey with Visual Studio.

## **4.6 Exploring Visual Studio Code**

Visual Studio Code is an **Integrated Development Environment (IDE)** developed by Microsoft to develop Desktop applications, GUI (Graphical User Interface), console, web applications, mobile applications, cloud, and web services, etc. With the help of this IDE, you can create managed code as well as native code. It uses the various platforms of Microsoft software development software like Windows store, Microsoft Silverlight, and Windows API, etc. It is not a language-specific IDE as you can use this to write code in C#, C++, VB(Visual Basic), Python, JavaScript, and many more languages. It provides support for 36 different programming languages. It is available for Windows as well as for macOS. VS Code contains different sections like Code editor, Solution Explorer, Output window and Properties as shown in the Fig. 4.4 below.

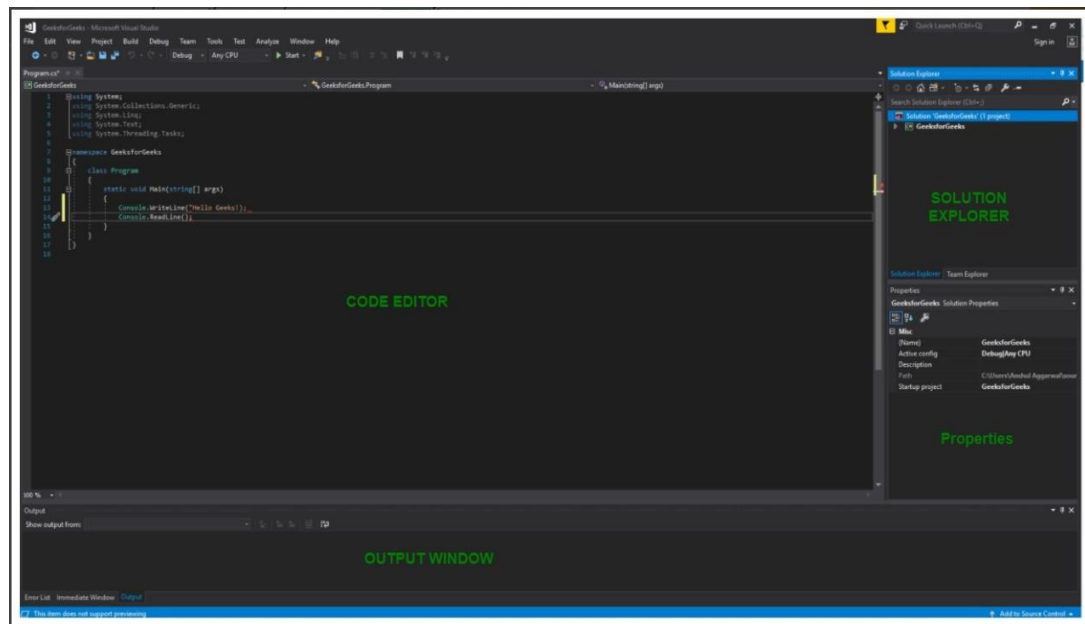


Fig. 4.4: Visual Studio Code

Coding in Flutter using Visual Studio provides a seamless development experience with powerful tools and features specifically designed for Flutter development.

## 4.7 Flutter Installation

### Get the Flutter SDK

1. Download the following installation bundle to get the latest stable release of the Flutter SDK: <https://flutter.dev/docs/get-started/install/windows>.  
For other release channels, and older builds, see the SDK releases page.
2. Extract the zip file and place the contained flutter in the desired installation location for the Flutter SDK.  
(for example, C:\Users\<your-user-name>\Documents).
3. If you do not want to install a fixed version of the installation bundle, you can skip steps 1 and 2. Instead, get the source code from the Flutter repo on GitHub, and change branches or tags as needed.

### Update your Path

If you wish to run Flutter commands in the regular Windows console, take these steps to add Flutter to the PATH environment variable

- From the Start search bar, enter 'env' and select Edit environment variables for your account.
- Under User variables check if there is an entry called Path
  - If the entry exists, append the full path to flutter\bin using ; as a separator from existing values.

- If the entry doesn't exist, create a new user variable named Path with the full path to flutter\bin as its value.

You have to close and reopen any existing console windows for these changes to take effect.

### Run flutter doctor

From a console window that has the Flutter directory in the path (see above), run the following command to see if there are any platform dependencies you need to complete the setup:

```
C:\src\flutter>flutter doctor
```

This command checks your environment and displays a report of the status of your flutter installation. Check the output carefully shown in the figure (Fig. 4.5), for other software you might need to install or further tasks to perform (shown in boldtext).

### For Example:



```
[~] Android toolchain - develop for Android devices
• Android SDK at D:\Android\sdk
X Android SDK is missing command line tools; download from https://goo.gl/XxQghQ
• Try re-installing or updating your Android SDK,
  visit https://flutter.dev/setup/#android-setup for detailed instructions.
```

Fig. 4.5: Report of flutter doctor

The following sections describe how to perform these tasks and finish the setup process. Once you have installed any missing dependencies, you can run the flutter doctor command again to verify that you've set everything up correctly.

### Emulator Setup

To set up an emulator in Visual Studio Code without Android Studio, you can use the Android Emulator extension. Here are the steps to do so:

- 1. Install Visual Studio Code:** If you haven't already, download and install Visual Studio Code from the official website.
- 2. Install Android Emulator Extension:** Open Visual Studio Code, go to the Extensions view by clicking on the square icon on the sidebar or pressing Ctrl+Shift+X, then search for "Android Emulator" and install the extension provided by Google.

**3. Install Android SDK Tools:** You'll need to have the Android SDK Tools installed on your system. You can download the SDK Tools separately from the Android Developer website.

**4. Set Android Home Environment Variable:** After installing the Android SDK Tools, set the `ANDROID_HOME` environment variable to the location where the SDK is installed. You can do this in your system's environment variables settings.

**5. Configure Emulator Settings:** Once the extension is installed, open the Command Palette (`Ctrl+Shift+P`) and search for "Android Emulator: Configure". Follow the prompts to configure the emulator settings such as the emulator image, Android API level, and hardware profile.

**6. Start Emulator:** After configuring the emulator, open the Command Palette again and search for "Android Emulator: Start". Select the emulator configuration you created and wait for the emulator to start.

**7. Run and Debug Your App:** With the emulator running, you can now run and debug your Android app directly from Visual Studio Code. Make sure your project is set up correctly with the necessary build configurations.

Following these steps, you can set up and use an Android emulator in Visual Studio Code without relying on Android Studio.

## 4.8 Adding Firebase to Flutter

### Step 1: Register your App with Firebase

1. In the center of the Firebase console's project overview page, click the iOS icon (plat\_ios) to launch the setup workflow.

If you've already added an app to your Firebase project, click Add app to display the platform options.

2. Enter your app's bundle ID in the iOS bundle ID field.

Find this bundle ID from your open project in XCode. Select the top-level app in the project navigator, then access the General tab. The Bundle Identifier value is the iOS bundle ID (for example, `com.yourcompany.ios-app-name`).

3. (Optional) Enter other app information as prompted by the setup workflow.
  - App nickname: An internal, convenience identifier that is only visible to you in the Firebase console.

- App Store ID: Used by Firebase Dynamic Links to redirect users to your App Store page and by Google Analytics to import conversion events into Google Ads. If your app doesn't yet have an App Store ID, you can add the ID later in your Project Settings.

4. Click Register app.

### **Step 2: Add a Firebase Configuration File**

1. Click Download GoogleService-Info.plist to obtain your Firebase iOS config file (GoogleService-Info.plist).
2. Using Xcode, move the file into the Runner/Runner directory of your Flutter app.
3. Back in the Firebase console setup workflow, click Next to skip the remaining steps.
4. Continue to Add FlutterFire plugins.

### **Step 3: Add Flutter Fire Plugins**

Flutter uses plugins to provide access to a wide range of platform-specific services, such as Firebase APIs. Plugins include platform-specific code to access services and APIs on each platform.

Firebase is accessed through a number of different libraries, one for each Firebase product (for example: Realtime Database, Authentication, Analytics, or Cloud Storage). Flutter provides a set of Firebase plugins, which are collectively called FlutterFire.

Since Flutter is a multi-platform SDK, each FlutterFire plugin is applicable for both iOS and Android. So, if you add any FlutterFire plugin to your Flutter app, it will be used by both the iOS and Android versions of your Firebase app.

1. Ensure that your app is not currently running in your emulator or on your device.
2. From the root directory of your Flutter app, open your pubspec.yaml file.
3. Add the FlutterFire plugin for the Firebase Core Flutter SDK.

dependencies:

flutter:

sdk: flutter

**Add the dependency for the Firebase Core Flutter SDK****firebase\_core: ^0.4.0+9**

Add the FlutterFire plugins for the Firebase products that you want to use

dependencies:

flutter:

sdk: flutter

Check that you have this dependency (added in the previous step).

firebase\_core: ^0.4.0+9

**4.9 ER/UML Diagrams**

As the strategic value of software increases for many companies, the industry looks for techniques to automate the production of software and to improve quality and reduce cost and time to the market. These techniques include component technology, visual programming, patterns and frameworks. Additionally, the development for the World Wide Web, while making some things simpler, has exacerbated these architectural problems. The UML was designed to respond to these needs. Simply, systems design refers to the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements which can be done easily through UML diagrams. Fig. 4.6 depicts the data flow diagram of the application.

**4.9.1 Data Flow Diagram**

A data flow diagram (DFD) is a graphical representation of the flow of data within a system. It's a visual tool used to depict the processes involved, the data exchanged, and the interactions between different components or entities within a system. Data flow diagrams are powerful tools for visualizing the flow of data within a system. They provide a clear and structured representation of processes, data flows, data stores, and external entities, making it easier to understand and communicate the workings of a system.



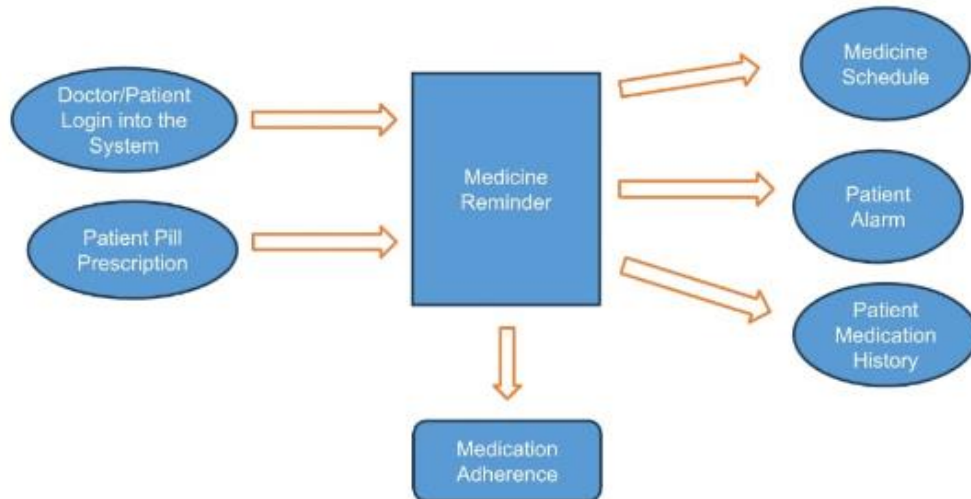


Fig. 4.6: Data flow diagram

#### 4.9.2 Use Case Diagram

A use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behaviour (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation (i.e., use case diagram). A key concept of use case modelling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behaviour in the user's terms by specifying all externally visible system behaviour. Use case diagram for enhancing medication adherence application is shown in Fig. 4.7 below.

##### Purpose of Use case diagram

Use case diagrams are typically developed in the early stage of development and people often apply use case modelling for the following purposes:

- Specify the context of a system.
- Capture the requirements of a system.
- Validate a system's architecture.
- Drive implementation and generate test cases.

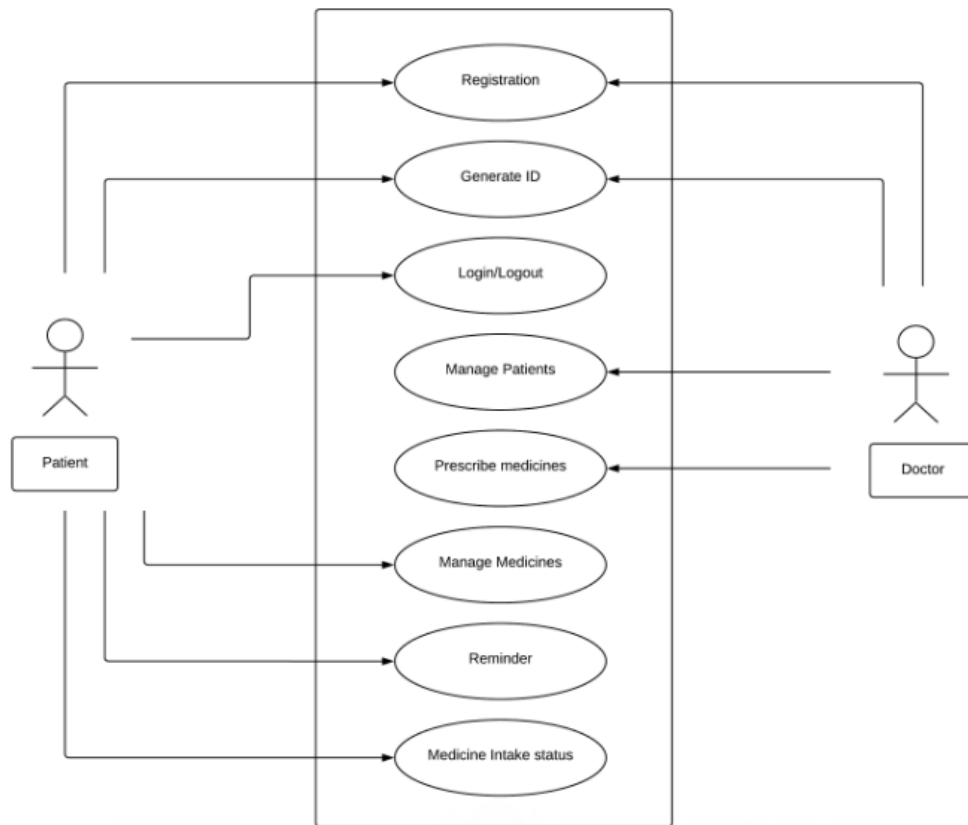


Fig. 4.7: Use case diagram

## 4.10 Software model selection

### Agile Process Model - Scrum Framework

The meaning of agile is swift or versatile. "Agile process model" refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance. Each iteration is considered as a short time "frame" in the agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements.

Following are the phases in the agile model are as follows:

1. Requirements gathering
2. Design the requirements
3. Construction/ iteration
4. Testing/ Quality assurance

5. Deployment
6. Feedback

## Scrum

Scrum is an agile development process focused primarily on ways to manage tasks in team-based development conditions. The Agile process model – scrum framework is shown in the Fig. 4.8. There are three roles in it, and their responsibilities are:

- **Scrum Master:** The scrum can set up the master team, arrange the meeting and remove obstacles for the process.
- **Product owner:** The product owner makes the product backlog, prioritizes the delay and is responsible for the distribution of functionality on each repetition.
- **Scrum Team:** The team manages its work and organizes the work to complete the sprint or cycle. A team of 5 to 9 members work on the tasks.

With scrum:

- Teams can deliver deliverables efficiently.
- It enables to use their time and money efficiently.
- Teams have greater visibility.
- Gain feedback from clients and customers.
- Individual effort of team members is given focus.

### 1. Product Backlog

A list of tasks that need to be completed to achieve the stakeholder's goals is set up.

### 2. Sprint Planning

The team determines the tasks from the product catalogue they aim to work on during the sprint.

### 3. Sprint Backlog

The tasks discussed during the sprint planning process are added to the sprint backlog.

### 4. Daily Scrum

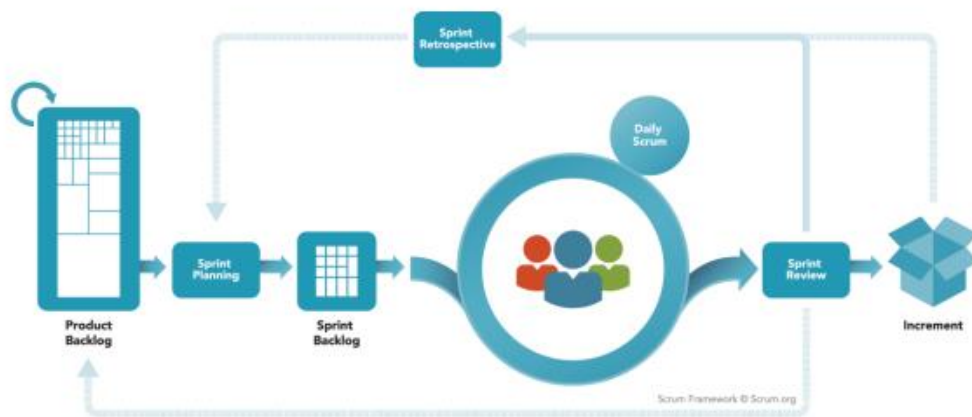
The team has daily scrum meetings of 15 minutes, where the team members synchronize their activities and plan what they plan to do for the next 24 hours.

## 5. Sprint Review

Once the sprint is complete, a sprint review takes place. This review involves the scrum master, product owner, and stakeholders. During this meeting, the team shows off what they accomplished during the sprint. During this time, questions are asked, observations are made, feedback and suggestions are also given.

## 6. Increment

A workable output is given to the stakeholders.



Fig, 4.8: Agile process model – Scrum framework

## 4.11 Architecture

The proposed mobile application for medication management requires a robust architecture to ensure effective functionality and seamless user experience. At its core, the application comprises several key components that work together to address the challenges of medication adherence. The application's architecture as illustrated in the Fig. 4.9 includes a client-server model, where the client side consists of the mobile application interface accessible to both doctors and patients. This interface enables doctors to prescribe medications directly within the application, leveraging a secure and synchronized data exchange mechanism.

On the server side, a centralized database stores medication prescriptions, patient profiles, and scheduling information. This database facilitates real-time synchronization between doctors and patients, ensuring accurate updates and notifications regarding medication schedules.

Additionally, the architecture incorporates a notification system responsible for delivering automatic medicine reminders to patients at scheduled times. This system leverages push notifications to alert patients about upcoming doses, helping to mitigate forgetfulness and promote adherence to prescribed schedules. Furthermore, the application architecture prioritizes security and privacy measures to safeguard sensitive medical information.

Overall, the architecture of the medication management application is designed to streamline the process of medication adherence by providing automated reminders, facilitating doctor-patient communication, and ensuring data accuracy and security.

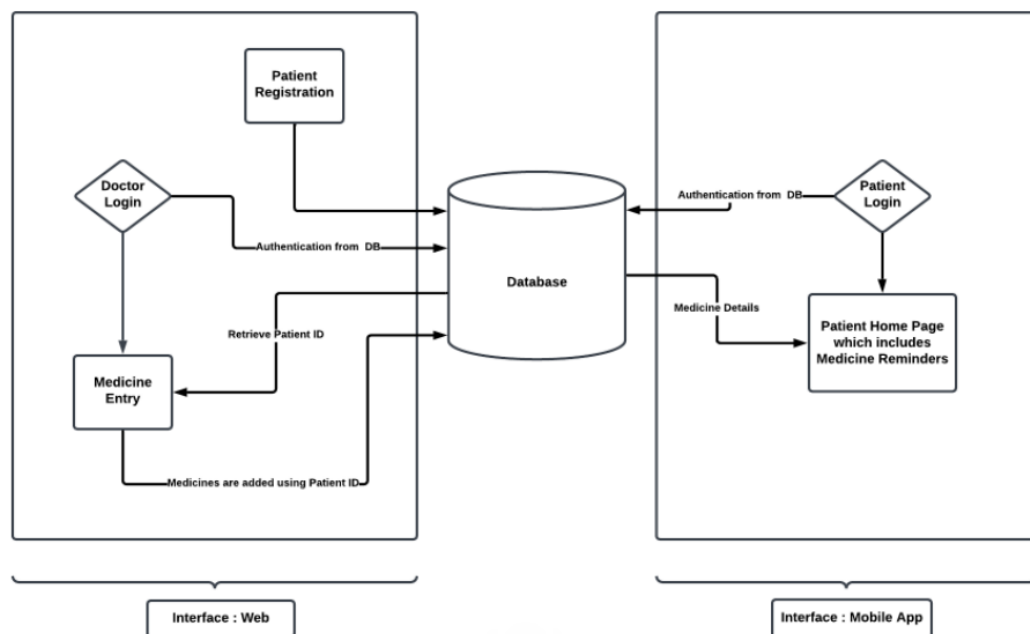


Fig. 4.9: Application Architecture

## CHAPTER 5

### IMPLEMENTATION

#### 5.1 main.dart

The main.dart file serves as the entry point for a Flutter application. It typically contains the main function, which is the starting point of the application, and defines the root widget of the app. Here's a breakdown of the key components typically found in a main.dart file.

**1. Imports:** Import necessary packages and libraries required for the application. In Flutter, commonly imported packages include flutter/material.dart, which provides material design widgets, and other custom packages or dependencies used in the project.

**2. Main Function:** The main() function is the entry point of the Flutter application. It calls the runApp() function and passes the root widget of the application. This function initializes the Flutter framework and starts the app's execution.

**3. Root Widget (MyApp):** The MyApp class is typically a StatelessWidget or StatefulWidget that represents the root of the application's widget tree. It usually contains the MaterialApp widget, which configures the overall theme, navigation, and other global properties of the app.

**4. MaterialApp:** MaterialApp is a widget provided by Flutter that implements the material design specifications. It configures the top-level MaterialApp for the application and sets properties such as the app's title, theme, initial route, and home page.

**5. Home Page Widget:** Within MaterialApp, the home property specifies the widget that represents the home page of the application. In the provided example, Medication Home Page is set as the home page widget.

**6. Medication Home Page Widget:** This is a StatelessWidget or StatefulWidget representing the home page of the application. It defines the layout and content displayed on the screen when the app is launched.

**7. Scaffold Widget:** Scaffold is a material design widget that provides a framework for implementing basic layout structure and components such as app bars, drawers

and bottom navigation. It typically serves as the top-level container for the home page widget and defines the overall layout structure.

**8. AppBar:** AppBar is a material design widget used to display a navigation bar at the top of the screen. It usually contains a title and optional actions or icons for navigation and interaction.

**9. Body Content:** The body of the Scaffold contains the main content of the home page. It typically consists of one or more widgets arranged in a column, row, or other layout structures to display content to the user.

**10. Button Widget:** In the provided example, a RaisedButton widget is used as a button component within the home page content. It triggers an action when pressed, such as navigating to another screen or performing a specific task.

Overall, the main.dart file as shown in the Fig. 5.1 sets up the basic structure and configuration of a Flutter application, defining the entry point, root widget, theme, and initial user interface components. Developers can further customize and extend this file to implement the desired functionality and user experience of their application.

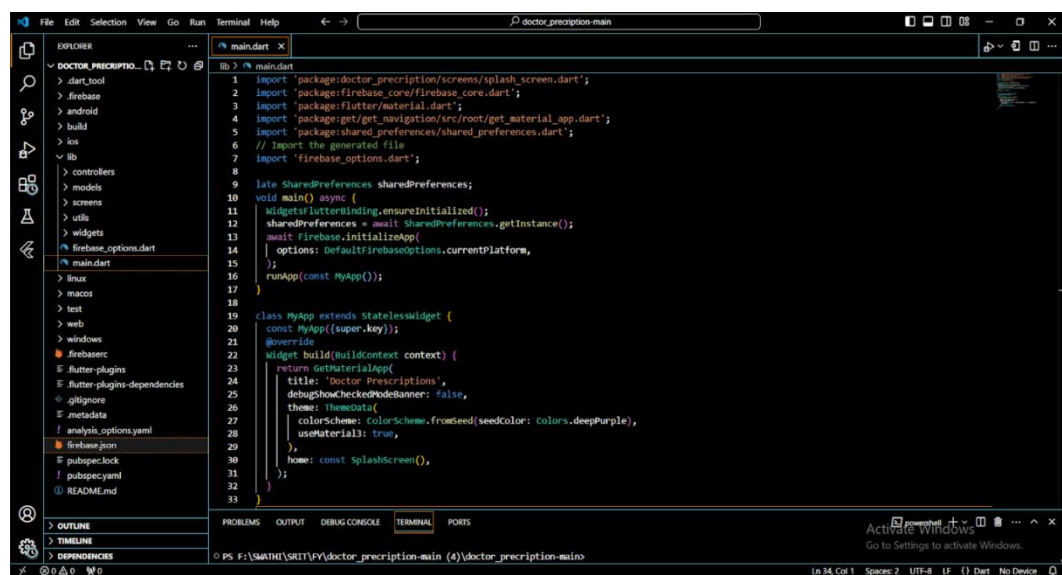


Fig. 5.1: main.dart

## 5.2 pubspec.yaml

The pubspec.yaml file in Flutter serves as a vital configuration file for defining project metadata, dependencies, and assets without code. Here's an overview of its key components:

- 1. Name and Description:** Specifies the name and a brief description of the Flutter project.
- 2. Version:** Indicates the version of the Flutter project, following semantic versioning conventions.
- 3. Environment:** Specifies the minimum Flutter SDK version required to run the project.
- 4. Dependencies:** Lists external packages and libraries required by the Flutter project, along with their versions.
- 5. Dev Dependencies:** Optionally, lists packages required only during development, such as testing frameworks or code generators.
- 6. Flutter SDK Constraints:** Specifies constraints for the Flutter SDK version used in the project.
- 7. Assets:** Defines the assets (e.g., images, fonts, JSON files) included in the Flutter project.
- 8. Flutter Assets:** Declares Flutter-specific assets like fonts and images under the flutter section, along with additional configuration options.

Thus, the pubspec.yaml file as shown in the Fig. 5.2 provides a structured way to manage project dependencies, assets, and metadata, ensuring consistent and reproducible builds across different environments in Flutter projects.

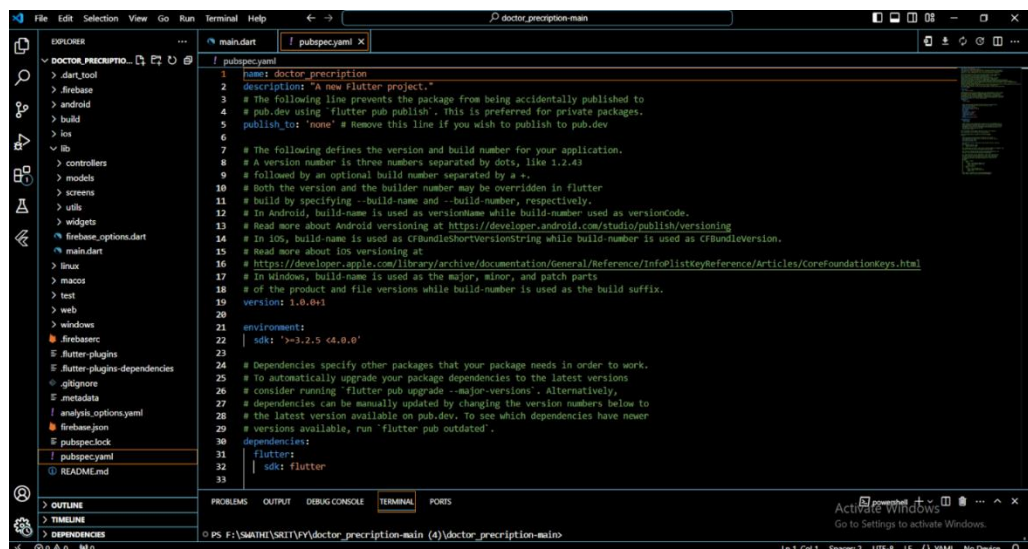


Fig. 5.2: pubspec.yaml



### 5.3 Firebase Cloud Firestore

Firebase Cloud Firestore is a NoSQL, document-oriented database service provided by Google as part of the Firebase suite. It offers real-time synchronization, scalability, offline support, and fine-grained security rules. Firestore organizes data into documents stored in collections and seamlessly integrates with other Firebase services. It provides SDKs for various platforms, making it easy to build real-time and scalable applications for mobile and web.

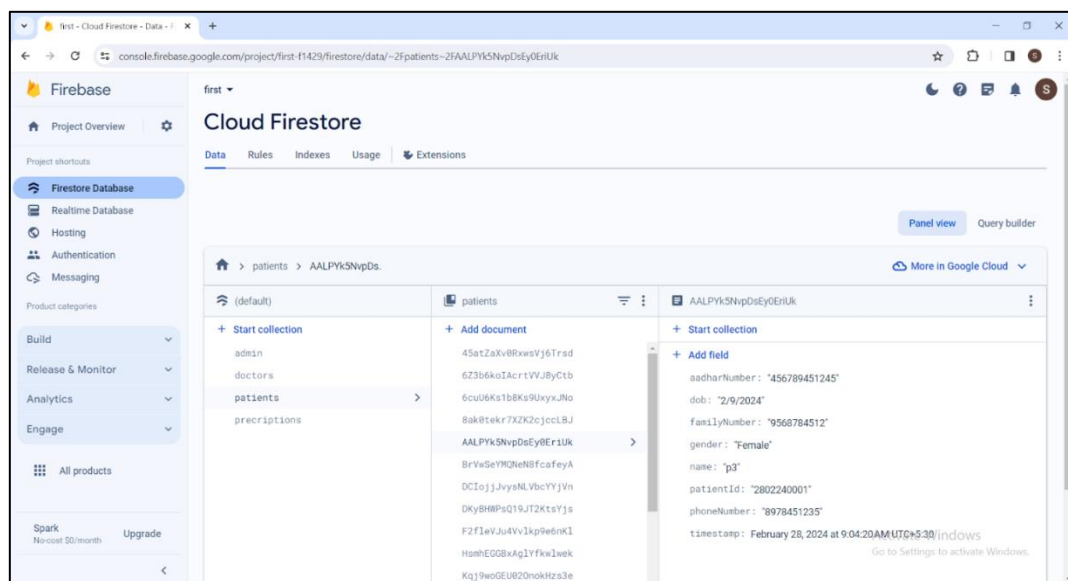


Fig. 5.3: Cloud Firestore

## CHAPTER 6

### OUTPUT SCREENS

#### 6.1 Home Screen

In Fig. 6.1, Registration of patients and doctors can login through their credentials.

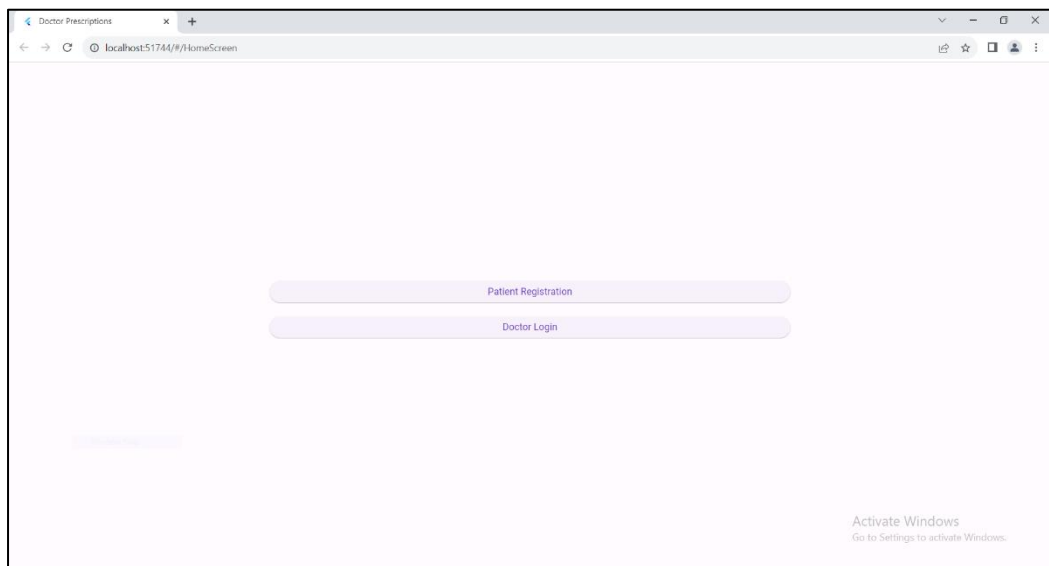


Fig. 6.1: Home Screen

#### 6.2 Patient Details Form

In Fig. 6.2, Patient details are collected to generate a Patient ID.

A screenshot of a web browser window titled "Doctor Prescriptions". The address bar shows "localhost:51744/#/RegisterFormScreen". The page has a back arrow and the title "Patient Details Form". The form contains the following fields: "Name" with the value "swathi", "Gender" with a dropdown menu showing "Female", "Date of Birth" with the value "3/12/2003", "Phone Number" with the value "9491277534", "Aadhar Number" with the value "123456789123", and "Family Member's Mobile Number" with the value "9876543210". A "Submit" button is located at the bottom center. An "Activate Windows" watermark is visible at the bottom right.

Fig. 6.2: Patient Details Form

### 6.3 Patient ID Generation

Patient ID is generated based on the date of registration, the screen is shown in the below Fig. 6.3

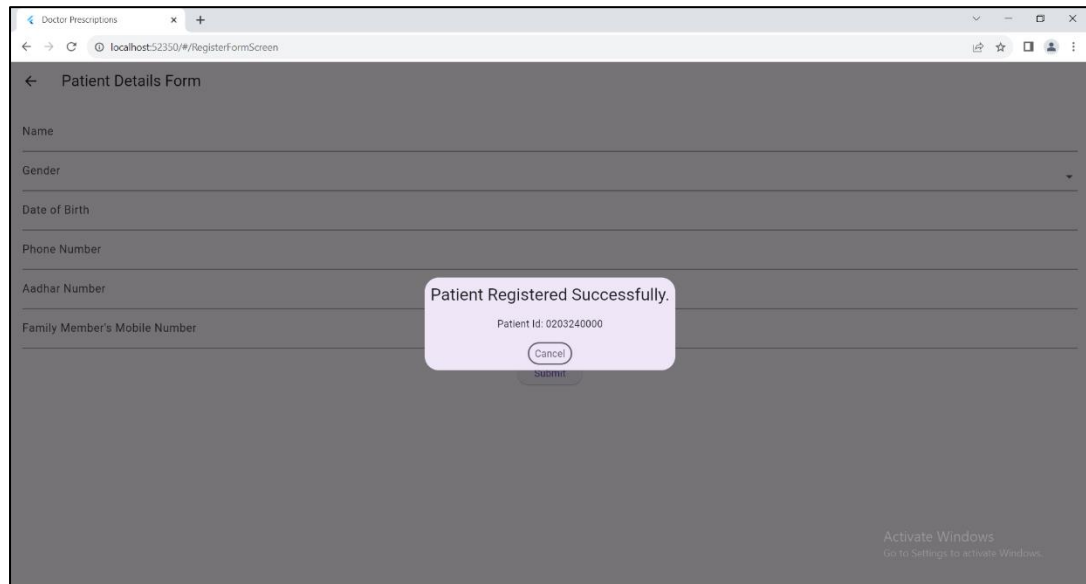


Fig. 6.3: Patient ID Generation

### 6.4 Doctor Login Page

In Fig. 6.4, doctor logs into the application using email and password.

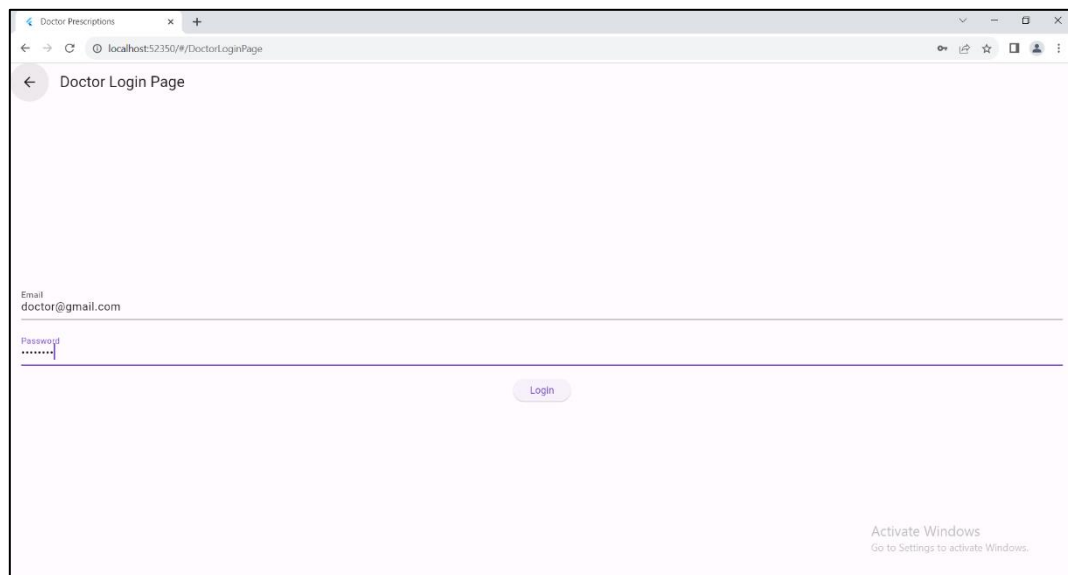


Fig. 6.4: Doctor Login Page

## 6.5 Prescribe Medicines

In Fig. 6.5, doctor prescribes the medicines which are mapped using the Patient ID.

The screenshot shows a web browser window titled 'Doctor Prescriptions' with the URL 'localhost:53535/#/DoctorPrescribeMedicinesScreen'. The page is titled 'Prescribe Medicines' and features a search bar for 'Patient ID' with the value '0203240000'. Below this is a 'Patient Details' section with the following information: ID: 0203240000, Name: swathi, Gender: Female, DOB: 3/12/2003, and Mobile: 9491277534. The main area contains two rows for prescribing medicines. The first row, 'med1', has 'noOfDays' set to 5, frequency selected as 'morning' and 'night', and 'Dosage Instructions' set to 'after food'. The second row, 'med2', has 'noOfDays' set to 3, frequency selected as 'afternoon', and 'Dosage Instructions' set to '-'. At the bottom, there are buttons for 'Save', 'Add Prescription', and 'Reset'. An 'Activate Windows' watermark is visible in the bottom right corner.

Fig. 6.5: Prescribe Medicines

## 6.6 Patient Login

In the Fig. 6.6, patient logs in using patient ID and phone number.

The screenshot shows a mobile application interface titled 'Patient Login Page'. It has a back arrow on the top left. The 'Patient id' field contains '0203240000'. The 'Mobile number' field contains '9491277534'. Below these fields is a purple 'Login' button. At the bottom of the screen is a numeric keypad with digits 1-9, 0, \*, and #, along with a 'Done' button. The status bar at the top shows the time as 10:27 and battery level at 73%.

Fig. 6.6: Patient Login

## 6.7 Patient Home

Here in Fig. 6.7, patient medicines are displayed which includes medicine name, timings and mark as taken options.

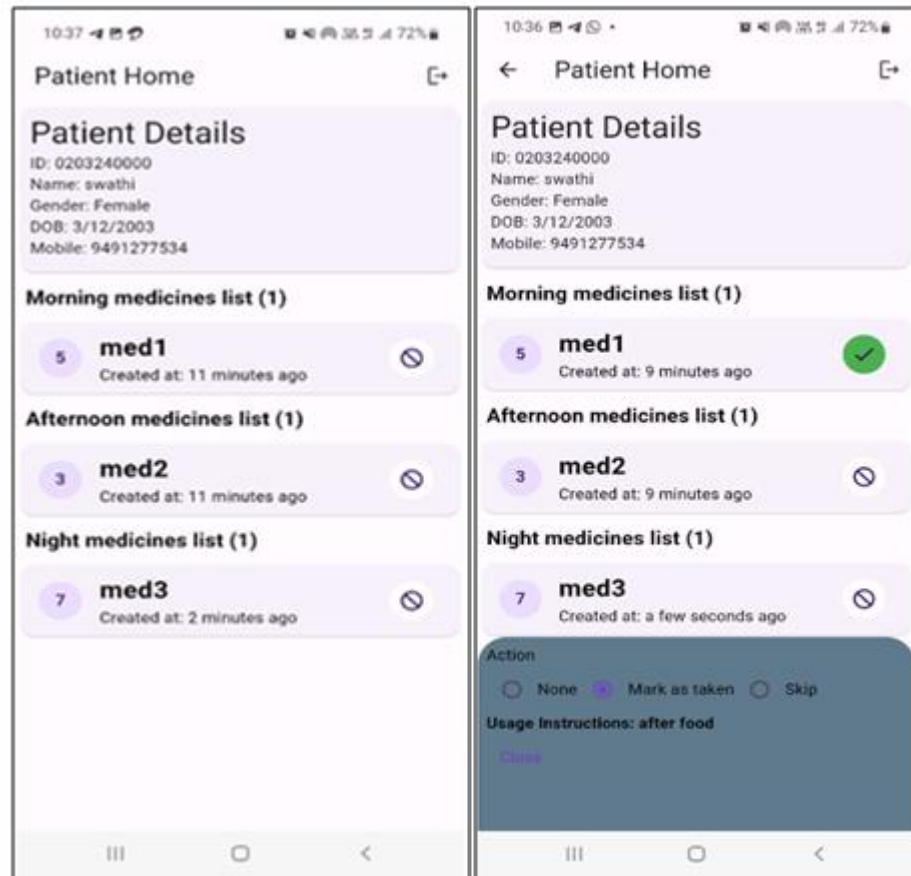


Fig. 6.7: Patient Home Screen

## **CHAPTER 7**

### **TESTING**

#### **7.1 Unit Testing**

Unit testing involves the testing of each unit or an individual component of the software application. It is the first level of functional testing. The aim behind unit testing is to validate unit components with its performance. A unit is a single testable part of a software system and tested during the development phase of the application software.

The purpose of unit testing is to test the correctness of isolated code. Unit component is an individual function or code of the application. White box testing approach is used for unit testing and usually done by the developers.

Whenever the application is ready and given to the Test engineer, he/she will start checking every component of the module or module of the application independently or one by one, and this process is known as Unit testing or components testing.

- Unit testing helps testers and developers to understand the base of code that makes them able to change defect causing code quickly.
- Unit testing helps in the documentation.
- Unit testing fixes defects very early in the development phase that's why there's a possibility to occur a smaller number of defects in upcoming testing levels.
- It helps with code reusability by migrating code and test cases.

#### **7.2 Integration Testing**

Integration testing is the second level of the software testing process after unit testing. In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units.

Integration testing is the second level of the software testing process after unit testing. In this testing, units or individual components of the software are tested in a

group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units.

### **Guidelines for Integration Testing**

- We go for the integration testing only after the functional testing is completed for each module of the application.
- We always do integration testing by picking module by module so that a proper sequence is followed, and also we do not miss out on any integration scenarios.
- First, determine the test case strategy through which executable test cases can be prepared according to test data.
- Examine the structure and architecture of the application and identify the crucial modules to test them first and also identify all possible scenarios.
- Design test cases to verify each interface in detail.
- Choose input data for test case execution. Input data plays a significant role in testing.
- If we find any bugs then communicate the bug reports to developers and fix defects and retest.
- Perform positive and negative integration testing.
- Here positive testing implies that if a query is sent to the admin through contact form and if it reaches the admin it is positive testing or else it is negative testing.

## **7.3 Functional Testing**

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

Functional testing is centred on the following items:

1. Valid Input: identified classes of valid input must be accepted.
2. Invalid Input: identified classes of invalid input must be rejected.
3. Functions: identified functions must be exercised.
4. Output: identified classes of application outputs must be exercised.

5. Systems/Procedures: interfacing systems or procedures must be invoked.

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

## **7.4 Validation Testing**

- The process of evaluating software during the development process or at the end of the development process to determine whether it satisfies specified business requirements.
- Validation Testing ensures that the product actually meets the client's needs. It can also be defined as to demonstrate that the product fulfils its intended use when deployed in an appropriate environment.

### **Workflow**

- Validation testing can be best demonstrated using V-Model. The Software/product under test is evaluated during this type of testing.
- Validation testing is testing where testers performed functional and functional testing. Here functional testing includes Unit Testing (UT), Integration Testing (IT) and System Testing (ST), and non-functional testing includes User acceptance testing (UAT).
- Validation testing is also known as dynamic testing, where we are ensuring that "we have developed the product right." And it also checks that the software meets the business needs of the client.

### **White box testing**

White Box Testing is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to testers so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing and Glass box testing.



## **7.5 Acceptance Testing**

User Acceptance Testing (UAT) 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. UAT is done in the final phase of testing after functional, integration and system testing is done.

## **7.6 Test Results**

This program has been successfully computed and tested. It is user-friendly and contains the necessary options for the user to accomplish the requested activities. To a large extent, application software satisfies the defined information requirements. The system was created with current and future requirements in mind, and it is extremely versatile. The system achieves the following objectives: instant access, increased medicine adherence and efficient medicine management.

## **CONCLUSION AND FUTURE WORK**

The proposed Mobile Application tackles the challenge of medication non-adherence by providing an interface for doctors to prescribe medications with automated reminders synced to patients. The Application also provides a user-friendly interface. This integration ensures real-time synchronization with the patient application, allowing for automated reminders set by healthcare professionals. The doctor-set reminders eliminate confusion, and patients can involve family members for added support. With features like customizable reminders, this application merges technology and healthcare seamlessly, to enhance medication adherence and overall patient care.

Additional functionalities like doctor availability, booking doctor appointments, lab reports and video consultation with doctors can also be implemented.

## REFERENCES

- [1] S.A.Patil, Monika Bhanuse, Snehal Mali and Vishaka Swami , “Review on Mobile application for Medicine Reminder”, International Research Journal of Modernization in Engineering Technology and Science, Volume:03/Issue:02/February-2021.
- [2] Samir V.Zanjal and Girish. R. Talmale, “Medicine Reminder and Monitoring System for Secure Health Using IOT”, International Conference on Information Security and Privacy, December 2020.
- [3] KeeHyun Park and Seung Hyeon Lim, “Construction of a Medication Reminder Synchronization System Based on Data Synchronization”, International Journal of Bio-Science and Bio-Technology Vol. 4, No. 4, December, 2012.
- [4] Lindsey Dayer, “Smartphone medication adherence apps: Potential benefits to patients and providers”, National Library, DOI: 10.1331/JAPhA.2013.12202.
- [5] Bob Tedeschi, “Healthful Reminders for Medications, Beyond an Apple a Day”, Sept. 29, 2010.

---

**ENHANCING MEDICATION ADHERENCE THROUGH MOBILE APPLICATION****Mr. C. Sudheer Kumar<sup>\*1</sup>, Ms. M. Swathi Reddy<sup>\*2</sup>, Ms. A. Leena Sri<sup>\*3</sup>,****Ms. C. Soumya<sup>\*4</sup>, Ms. M. Parimala<sup>\*5</sup>**

<sup>\*1</sup>Assistant Professor, Department Of Computer Science And Engineering, Srinivasa Ramanujan Institute Of Technology, Anantapur, Andhra Pradesh, India.

<sup>\*2,3,4,5</sup>Student, Department Of Computer Science And Engineering, Srinivasa Ramanujan Institute Of Technology, Anantapur, Andhra Pradesh, India.

DOI : <https://www.doi.org/10.56726/IRJMETS50524>

---

**ABSTRACT**

Human forgetfulness poses a significant challenge in the context of medication adherence, leading to adverse health consequences and, in some instances, premature deaths. Recognizing the importance of addressing this issue, our project involves the development of a specialized mobile application aimed at improving medication management. This innovative application is designed to streamline the medication adherence process by seamlessly integrating doctor's prescriptions with patient's daily routines. The core objective of the proposed mobile application is to provide automatic medication reminders. Through this platform, healthcare professionals can prescribe medications directly, with the data seamlessly synchronized in real-time with the patient's interface. Patients can include their family members in the app, so if they forget to take their medications despite reminders, their family can help remind them. This dynamic approach ensures that patients receive accurate and timely updates regarding their medication schedules, promoting a proactive and informed approach to treatment. The proposed mobile application not only serves as a reliable tool for patients but also facilitates healthcare providers to track the treatment. By using technology to connect the prescription with the patient's adherence, our suggested solution aims to reduce the harmful health effects that are linked to medication non-adherence.

**Keywords:** Mobile Application, Medication Adherence, Reminder System, Automatic Alarm.

---

**I. INTRODUCTION**

In today's world, the stress and busy routines we face can harm our health and make us prone to getting sick. So, it's really important for us to make sure we take care of ourselves and stay fit. We need to deal with health problems early on to stay healthy and strong. That's the reason why we should create ways to help people manage their health better. To meet this need, developing a mobile application that serves as a vital tool in healthcare management is important.

This innovative application empowers doctors to prescribe medications seamlessly, incorporating a feature for setting automatic reminders directly synchronized with the patient's interfaces. Moreover, recognizing the importance of familial support in healthcare, the platform allows patients to include family members. This inclusion facilitates family members in reminding patients to adhere to their medications. With a primary focus on individuals prone to forgetting medication schedules, the proposed mobile application helps to enhance medication adherence in the middle of our fast-paced lives.

**II. LITERATURE REVIEW**

In this modern world, the prevalence of medicine reminder applications underscores the significance of technology in facilitating medication adherence among patients. Various applications have been introduced, each incorporating distinct ideas and methodologies to address the crucial issue of timely medication intake. Despite many applications that have been already implemented in this domain, certain concerns still persist, indicating areas that necessitate further exploration and enhancement.

A notable concern revolves around the potential errors associated with individualized reminder settings by patients themselves. While many applications empower users to set reminders based on their treatment plans, the risk of error prone entries remains a valid concern. The consequences of such inaccuracies can have a huge impact, potentially decreasing the efficiency of the overall treatment plan.

Furthermore, financial constraints pose a significant barrier to the widespread adoption of certain advanced features, particularly those involving Internet of Things (IoT) devices. Not all individuals have the financial means to invest in such technologies, raising questions about the accessibility of IOT based reminder applications. This financial aspect extends to the hiring of personal assistants or nurses, as not everyone can afford dedicated personnel to oversee and manage their healthcare needs.

### III. METHODOLOGY

A mobile app where doctors can set medication reminders when they prescribe medicines. This information will automatically sync with the patient's mobile app. By linking prescriptions to patient IDs, this mobile application provides a user interface that enables us to directly issue prescriptions. By ensuring synchronization with the patient application, this connection sets off automated reminders that help patients take their medications on time. There are no questions about medications or dosage guidelines because the doctors themselves set the reminder. The patients can also add other family members to monitor them. Additionally, the application offers a number of features including customizable time-stamps for snooze reminders and, all of which help in keeping track of patient's drug intake.

When a patient first goes to the hospital, they need to register in the system by giving a phone number, name, age, and Aadhar number. After that, they are provided with a Patient ID which is unique. The application has mainly three parts.

#### **Patient Registration Web Page:**

1. An interface to collect Patient Details like Name, Gender, Date of birth, Phone number, Aadhar Number and Family Member Mobile number.
2. Once the details are collected in proper format, an unique ID will be generated for the patient for further use.

#### **Doctor Web Page :**

1. A login page which collects their mail id and password as authentication.
2. A Prescription page which allows doctors to prescribe medicines by entering the unique patient ID.
3. Doctors can add the medicines, time, dosage instructions, duration of the medication and next review date
4. Once the doctor submits the prescription, the information will automatically update in the patient's mobile app.

#### **Patient Mobile App:**

1. Patients can login in the app using their mobile number, Patient ID which is given during the first registration.
2. All the reminders are already set by the doctors.
3. The patients have multiple options like snooze the medicine, mark the medicine as taken or even skip the medicine. This functionality helps in tracking the medicine intake.

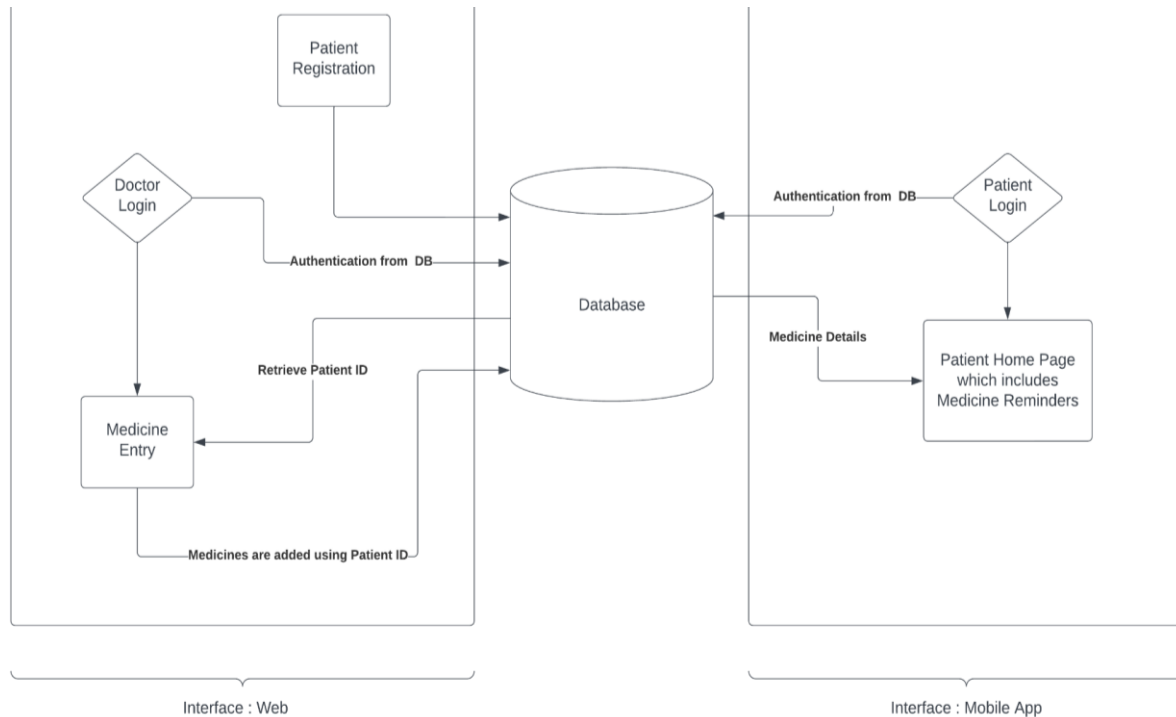
#### **ADVANTAGES**

1. The primary advantage is the enhancement of medication adherence through automated reminders, reducing the risk of negative health implications.
2. Patient ID avoids any redundancy that might occur if two or more patients have the same details.
3. Since the reminders are automatically set based on the prescription given by the doctors, there is no possibility of setting the remainders incorrectly.
4. An interface where patients can add their family members.
5. Multiple options like snooze, take or skip the medicines.

#### **DISADVANTAGES**

1. Clinics cannot bear the cost of maintaining a database for patient information.
2. Individuals without smartphones cannot utilize this application.

# BLOCK DIAGRAM



## IV. CONCLUSION

The proposed Mobile Application tackles the challenge of medication non-adherence by providing an interface for doctors to prescribe medications with automated reminders synced to patients. The Application also provides a user-friendly interface. This integration ensures real-time synchronization with the patient application, allowing for automated reminders set by healthcare professionals. The doctor-set reminders eliminate confusion, and patients can involve family members for added support. With features like customizable reminders, this application merges technology and healthcare seamlessly, to enhance medication adherence and overall patient care.

Additional functionalities like doctor availability, booking doctor appointments, lab reports and video consultation with doctors can be implemented.

## V. REFERENCES

- [1] S. A. Patil, Monika Bhanuse, Snehal Mali and Vishaka Swami, "Review on Mobile application for Medicine Reminder", International Research Journal of Modernization in Engineering Technology and Science, Volume:03/Issue:02/February-2021.
- [2] Samir V.Zanjal and Girish. R. Talmale, "Medicine Reminder and Monitoring System for Secure Health Using IOT", International Conference on Information Security and Privacy, December 2020.
- [3] KeeHyun Park and Seung Hyeon Lim, "Construction of a Medication Reminder Synchronization System Based on Data Synchronization", International Journal of Bio-Science and Bio-Technology Vol. 4, No. 4, December, 2012.
- [4] Lindsey Dayer, "Smartphone medication adherence apps: Potential benefits to patients and providers", National Library, DOI: 10.1331/JAPhA.2013.12202.
- [5] Bob Tedeschi, "Healthful Reminders for Medications, Beyond an Apple a Day", Sept. 29, 2010.
- [6] John K. Zao, Mei-Ying Wang, Peihuan Tsai and Jane Liu, "Smart phone based medicine in-take scheduler, reminder and monitor", IEEE 01-03 July 2010.
- [7] Lars Osterberg, M.D., and Terrence Blaschke, M.D., "Adherence to Medication", The New England Journal of Medicine DOI: 10.1056/NEJMra050100.

## CERTIFICATIONS



### *International Research Journal Of Modernization in Engineering Technology and Science*

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 06/Issue 03/60300060171

Date: 21/03/2024

### *Certificate of Publication*

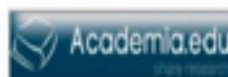
*This is to certify that author "Mr. C. Sudheer Kumar" with paper ID "IRJMETS60300060171" has published a paper entitled "ENHANCING MEDICATION ADHERENCE THROUGH MOBILE APPLICATION" in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 06, Issue 03, March 2024*

*A. Deneak*

Editor in Chief



*We Wish For Your Better Future*  
[www.irjmets.com](http://www.irjmets.com)







*International Research Journal Of Modernization  
in Engineering Technology and Science*

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 06/Issue 03/60300060171

Date: 21/03/2024

*Certificate of Publication*

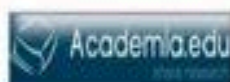
This is to certify that author "Ms. M. Swathi Reddy" with paper ID "IRJMETS60300060171" has published a paper entitled "ENHANCING MEDICATION ADHERENCE THROUGH MOBILE APPLICATION" in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 06, Issue 03, March 2024

*A. Devala*

Editor in Chief



We Wish For Your Better Future  
[www.irjmets.com](http://www.irjmets.com)







*International Research Journal Of Modernization  
in Engineering Technology and Science*

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 06/Issue 03/60300060171

Date: 21/03/2024

*Certificate of Publication*

This is to certify that author "Ms. A. Leena Sri" with paper ID  
"IRJMETS60300060171" has published a paper entitled  
"ENHANCING MEDICATION ADHERENCE THROUGH MOBILE  
APPLICATION" in International Research Journal Of Modernization  
In Engineering Technology And Science (IRJMETS), Volume 06,  
Issue 03, March 2024

*A. Deenath*

Editor in Chief



*We Wish For Your Better Future*  
[www.irjmets.com](http://www.irjmets.com)





*International Research Journal Of Modernization  
in Engineering Technology and Science*

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETSCertificate/Volume 06/Issue 03/60300060171

Date: 21/03/2024

*Certificate of Publication*

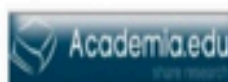
This is to certify that author "Ms. C. Soumya" with paper ID  
"IRJMETSC60300060171" has published a paper entitled  
"ENHANCING MEDICATION ADHERENCE THROUGH MOBILE  
APPLICATION" in International Research Journal Of Modernization  
In Engineering Technology And Science (IRJMETSC), Volume 06,  
Issue 03, March 2024

*A. Dhanasekhar*

Editor in Chief



We Wish For Your Better Future  
[www.irjmets.com](http://www.irjmets.com)





*International Research Journal Of Modernization  
in Engineering Technology and Science*

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETSCertificate/Volume 06/Issue 03/60300060171

Date: 21/03/2024

*Certificate of Publication*

This is to certify that author "Ms. M. Parimala" with paper ID  
"IRJMETSCertificate/Volume 06/Issue 03/60300060171" has published a paper entitled  
"ENHANCING MEDICATION ADHERENCE THROUGH MOBILE  
APPLICATION" in International Research Journal Of Modernization  
In Engineering Technology And Science (IRJMETSCertificate), Volume 06,  
Issue 03, March 2024

*A. Dhanu*

Editor in Chief



*We Wish For Your Better Future*  
[www.irjmets.com](http://www.irjmets.com)

