

LAB REPORT

Submitted by

CHARVEE RATHOD [RA201100301121]

Under the Guidance of

Dr. Aruna M

Assistant professor, Department of Computing Technologies

In partial satisfaction of the requirements for the degree of

**BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING**



**SCHOOL OF COMPUTING
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR - 603203**

JUNE 2022



SRM INSTITUTION OF SCIENCE AND TECHNOLOGY KATTANKULATHUR-603203

BONAFIDE CERTIFICATE

Certified that this lab report titled MedCare application is the bonafide work done by Charvee Rathod (RA2011003011121) who carried out the lab exercises under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.

SIGNATURE

Dr. Aruna M
SEPM – Course Faculty

Assistant professor
Department of Computing Technologies

ABSTRACT

A pill reminder app notifies a person when to take specific medications. A person can download it on their smartphone or some other devices. Taking medication at the right time improves how well it works and how safe it is. The application allows the user to store pill objects and multiple alarms for those pills. Alarms have one time of day and can occur on multiple days of the week. The user is able to view their pills in a today view and can select the date to view medicines. In addition, the application stores the history of when each medication was taken; this will aid the user in keeping track of their medication usage.

INDEX

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	
	LIST OF TABLES	
	LIST OF FIGURES	
1	PROBLEM STATEMENT	1
2	PROCESS MODEL AND STAKEHOLDERS	3
3	IDENTIFYING REQUIREMENTS	5
4	PROJECT PLAN AND EFFORT	6
5	RISK ANALYSIS AND WORK BREAKDOWN STRUCTURE	10
6	SYSTEM ARCHITECTURE, USE CASE AND CLASS DIAGRAM	14
7	ENTITY RELATIONSHIP DIAGRAM	16
8	DATA FLOW DIAGRAM	17
9	SEQUENCE AND COLLABORATION DIAGRAM	18
10	DEVELOPMENT OF TESTING FRAMEWORK/USER INTERFACE	20
11	TEST CASES MANUAL	25
12	TEST CASE REPORT	28
13	ARCHITECTURE DESIGN/FRAMEWORK AND IMPLEMENTATION	29
	CONCLUSION	
	REFERENCES	

LIST OF TABLES

Table No.	Title	Page No.
2.1	Stakeholders of the project	3
4.1	Project management plan	6
4.2.1	Effort and Cost estimation	7
4.2.2	Infrastructure /Resource cost	8
4.2.3	Maintenance and support cost	8
4.3.2	Identification of team members	8
4.3.2	Responsibility assignment matrix	8
5.1	Risk management	10
5.2	Risk Appetite table	11
10.1	Test plan	23
11.1	Functional test cases	25
11.2	Non-functional test cases	26
12.1	Test case progress table	28
12.2	Test case coverage table	28

LIST OF FIGURES

Figure No.	Title	Page No.
5.1	Gantt chart	12
5.2	SWOT Analysis	13
5.3	Work breakdown structure	13
6.1	System architecture requirements	14
6.2	Use-case diagram	15
6.3	Class diagram	15
7.1	Entity relationship diagram	16
8.1	Data flow diagram	17
9.1	Sequence diagram	18
9.2	Collaboration diagram	19

CHAPTER 1

PROBLEM STATEMENT

Problem Statement

To develop an application that will give regular reminders to its user for taking their medications

Project Description

It is designed for users who need a little help keeping track of their medication schedule and who are dedicated to keeping the schedule. The application allows the user to store pill objects and multiple alarms for those pills. Alarms have one time of day and can occur on multiple days of the week. The user is able to view their pills in a today view and can select the date to view medicines. In addition, the application stores the history of when each medication was taken; this will aid the user in keeping track of their medication usage.

ONE PAGE BUSINESS CASE TEMPLATE



DATE	16 March 2022
SUBMITTED BY	Charvee Rathod
TITLE / ROLE	Team member

THE PROJECT

In bullet points, describe the problem this project aims to solve or the opportunity it aims to develop.

- It is meant to aid the forgetful and busy with remembering to take their daily medications.
- It is designed for users who need a little help keeping track of their medication schedule and who are dedicated to keeping the schedule.

THE HISTORY

In bullet points, describe the current situation.

- Everyone leads a very busy lifestyle and in this era of rushing they often forget to take their meds and take care of their health, which we believe is the most important thing.

LIMITATIONS

List what could prevent the success of the project, such as the need for expensive equipment, bad weather, lack of special training, etc.

- A lot many people in our target audience are not that tech savvy(but we will try our best to make the application as user friendly as possible)

APPROACH

List what is needed to complete the project.

- The application allows the user to store pill objects and multiple alarms for those pills.
- Alarms have one time of day and can occur on multiple days of the week. The user is able to view their pills in a today view and can select the date to view medicines.
- In addition, the application stores the history of when each medication was taken; this will aid the user in keeping track of their medication usage.

BENEFITS

In bullet points, list the benefits that this project will bring to the organization.

- It will improve the health of our users
- It will make doctor appointments even more convenient
- Buying medicines would become easier as the user will get reminder on when to restock

CHAPTER 2

PROCESS MODEL AND STAKEHOLDERS

Agile Methodology

Agile methodology will be used in the development of MedCare, medicine reminder app. The Agile methodology is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continuous improvement at every stage. Once the work begins, teams cycle through a process of planning, executing, and evaluating.

Agile is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches. Instead of betting everything on a "big bang" launch, an agile team delivers work in small, but consumable, increments. Requirements, plans, and results are evaluated continuously so teams have a natural mechanism for responding to change quickly.

Scrum is a subset of Agile. It is a lightweight process framework for agile development, and the most widely-used one.

Table 2.1 stakeholders of the project

Stakeholder Name	Activity/ Area /Phase	Interest	Influence	Priority (High/ Medium/ Low)
Developer	Achieve targets, help improve the health of people, updating the app frequently, removing bugs	High	High	High
Consumer	Providing feedback	High	Medium	High
Owner	Manages	High	Low	Medium

	successful project implementation			
Business analyst	Analyze and expand the business proposals	High	Medium	High
Tester	Checks the application by using various inputs	High	High	High
Technical Lead	Coding and fixing bugs	High	High	High

CHAPTER 3

IDENTIFYING REQUIREMENTS

System Requirements:

- It will work on android as well as IOS devices
- we will use firebase to store the user's data

Functional Requirements:

- Users will be able to log in
- The application will display all the medicines with its reminders and available medicines quantity
- In case the user loses or uninstalls the application, the user can again install it and log in and use the application like before as the data will be stored in the cloud
- Users will be able to store data about the medical medicine timings
- Users can set reminders and snooze times
- It will alert the users if the medicine is going to get over

Non-Functional Requirements:

- add multiple colors to the application so that it looks attractive to use
- it will be user friendly
- add SMS systems, if the user forgets to take medicines 3 consecutive times, an SMS alert will go to another person who is set in close people
- This application will be secured as it will have login credentials

CHAPTER 4

PROJECT PLAN AND EFFORT

4.1 Project Management Plan

Focus Area	Details
Integration Management	Governance Framework Project Team Structure Roles & Responsibilities of Team Change Management (Change Control, Issue Management) Project Closure
Scope Management	Scope Statement Requirement Management (Gathering, Control, Assumption, Constraint Stakeholder) Define Deliverable Requirement Change Control Activities and Sub-Tasks
Schedule Management	Define Milestones Schedule Control
Cost Management	Estimate Effort Assign Team Budget Control
Quality Management	Quality Assurance: Quality assurance will be managed including governance, roles and responsibilities, tools and techniques and reporting Quality Control: Specify the mechanisms to be used to measure and control the quality of the work products

Resource Management	Estimate and Manage the need People: People & Skills Required Finance: Budget Required Physical: Facilities, IT Infrastructure
Stakeholder	Identifying, Analyzing, Engaging Stakeholders
Communication Management	Determine communication requirements, roles and responsibilities, tools and techniques. [Type of Communication, Schedule, Mechanism Recipient]
Risk Management	Identifying, analysing, and prioritizing project risks
Procurement Management	Adhering to organization procurement process

Estimation

4.2.1 Effort and Cost Estimation

Activity Description	Sub-Task Description	Effort (in hours)	Cost in INR
Android-Application Development	To create an android-based application	20	30,000
Research and development	On the input given in the database and the way of password gets generated	4	5000
Android-Application Maintenance	Weekly application A maintenance check of data.	6	3000
Marketing	Advertisements, Public Relations	10	5000
Testing	Simultaneously testing the App application.	5	3000

4.2.2 Infrastructure/Resource Cost

Infrastructure Requirement	Qty	Cost per qty
Laptops/Machines	2	50,000
Data Infrastructure	1	Pay-As-You-Go
Web Page Domain	1	999/year

4.2.3 Maintenance and Support Cost [OpEx]

Category	Details	Qty	Cost per qty per annum	Cost per item
People	Network, System, Database, website	3	20,00,000	60,00,000
License	Operating System Database Middleware IDE	10	10,000	1,00,000
Infrastructures	Data and Network	20	20,000	4,00,000

Project Team Formation

4.3.1 Identification Team members

Name	Role	Responsibilities
Chetas Shree	Key Business User (Product Owner)	Provide clear business and user requirements
Khushi	Project Manager	Manage the project
Charvee	Business Analyst	Discuss and Document Requirements
Chetas Shree	Technical Lead	Design the end-to-end architecture
Khushi	UX Designer	Design the user experience
Chetas Shree	Frontend Developer	Develop user interface
Chetas Shree	Backend Developer	Design, Develop and Unit Test Services/API/DB
Charvee	Database Architect	Design the cost effective, highly available and scalable architecture
Khushi	Database Operations	Provision required Services
Charvee	Tester	Define Test Cases and Perform Testing

4.3.2 Responsibility Assignment Matrix

RACI Matrix		Team Members		
Activity	Charvee (BA)	Chetas Shree (Developer)	Khushi (Project Manager)	

User Requirement Documentation	A	C/I	C
Stakeholder Analysis	R	A	R
Requirement Analysis	A	R	I
Cost/Effort Estimation	A	I	R
Project Statement	C	I	A

A	Accountable
R	Responsible
C	Consult
I	Inform

CHAPTER 5

RISK ANALYSIS AND WORK BREAKDOWN STRUCTURE

Risk Analysis:

Table 5.1 Risk Management

Risk Description	Impact Description
Lack of top management commitment to the project	Whether top management was involved in Lean implementation, spent time in the workplace to supervise the process as part of their support and provided the necessary resources to implement Lean in the workplace.
Failure to gain user commitment	Lack of due diligence
Lack of adequate user involvement	Without user involvement, nobody in the business feels committed to a system, and many may even be hostile to it.
Failure to manage end user expectation	User involvement is making sure the voices of people are heard and they are able to actively shape and improve the services they use and influence local and national policy.
Crashing of Applications	Applications crashing means that something went wrong. There are a few different ways of how an application can crash, including

	code error and expired domain, among others.
Introduction of new Technology	<p>Not implementing the right system procedure.</p> <p>Failing to win staff over technology.</p> <p>Not training staff adequately in the utilization of the new technology.</p>

4.2 Risk appetite table

Risk Appetite [Accept/ Mitigate/ Transfer/Avoid]	Action
Accept	Team leader should give more time towards the project
Accept	<p>1. Act quickly</p> <p>2. Solve the problem as fast you can</p> <p>3. Follow up with the customer</p>
Avoid/Mitigate	Stakeholders meeting, product owner, sprint demos, design sessions
Mitigate	Identify the resources, create a schedule, organize your development with your team.

Accept	Limit the damage, communicate the fix, implement the escalation plans, check for any safety issue
Transfer/Mitigate	Training the developers all the latest technologies related to their project

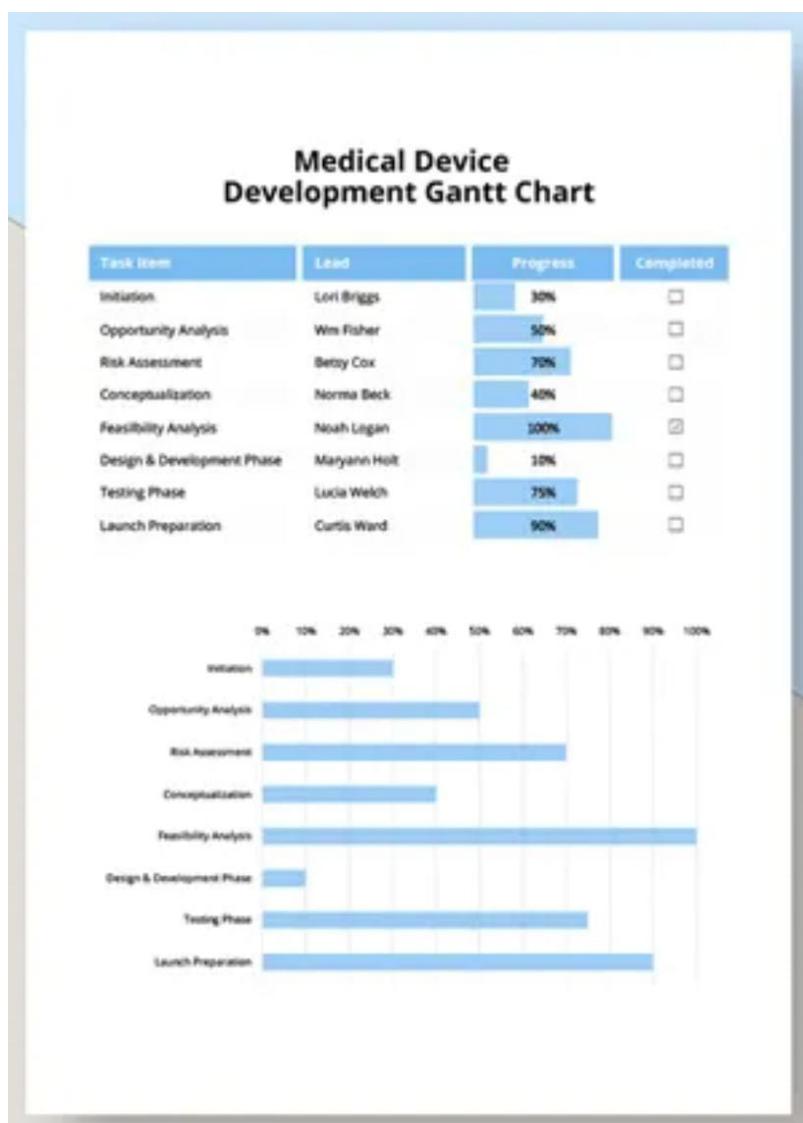


Fig 5.1 Gantt Chart

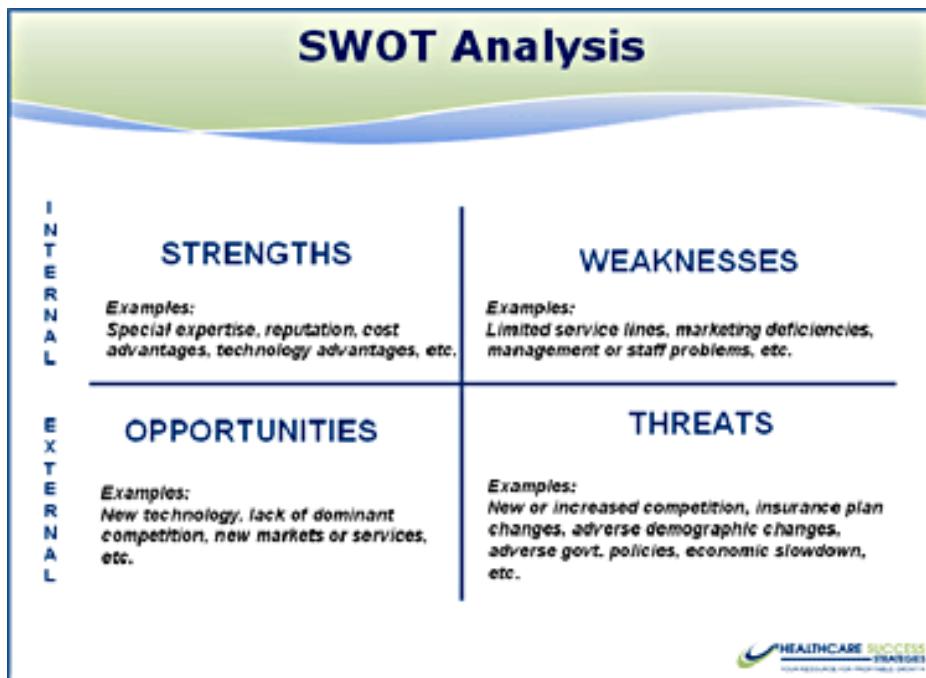


Fig 5.2 SWOT Analysis

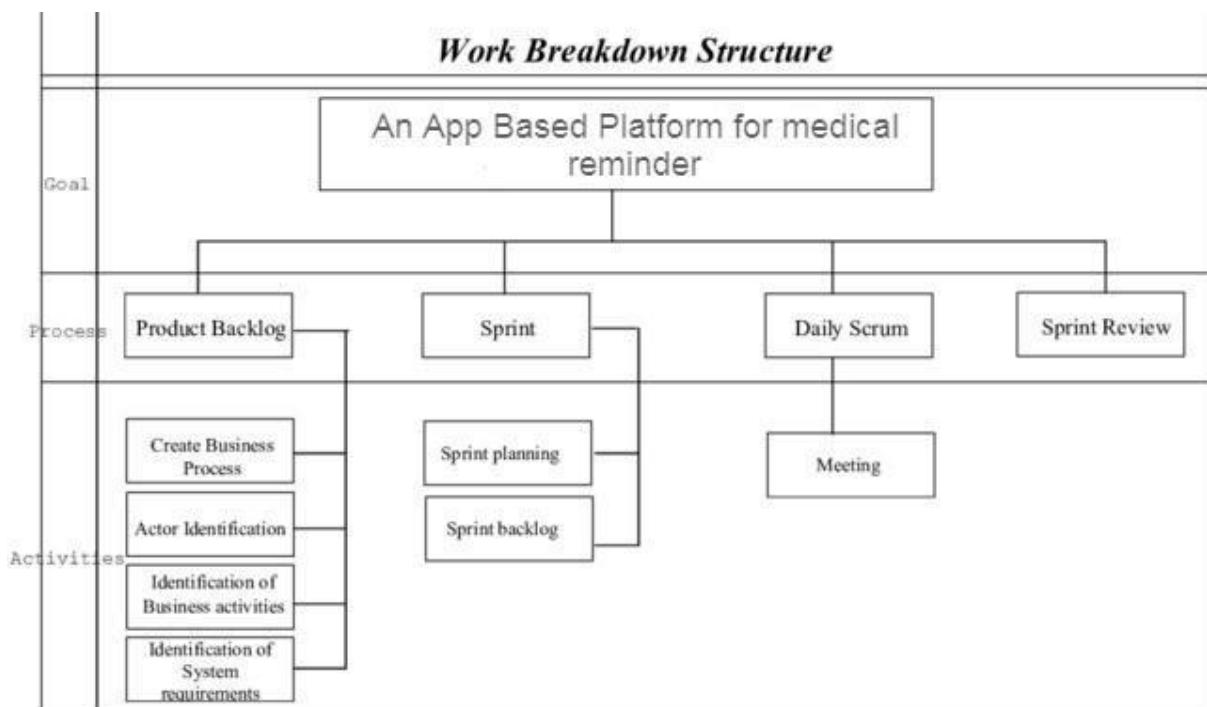


Fig 5.3 Work Breakdown Structure

CHAPTER 6

SYSTEM ARCHITECTURE, USE CASE AND CLASS DIAGRAM

Requirements:

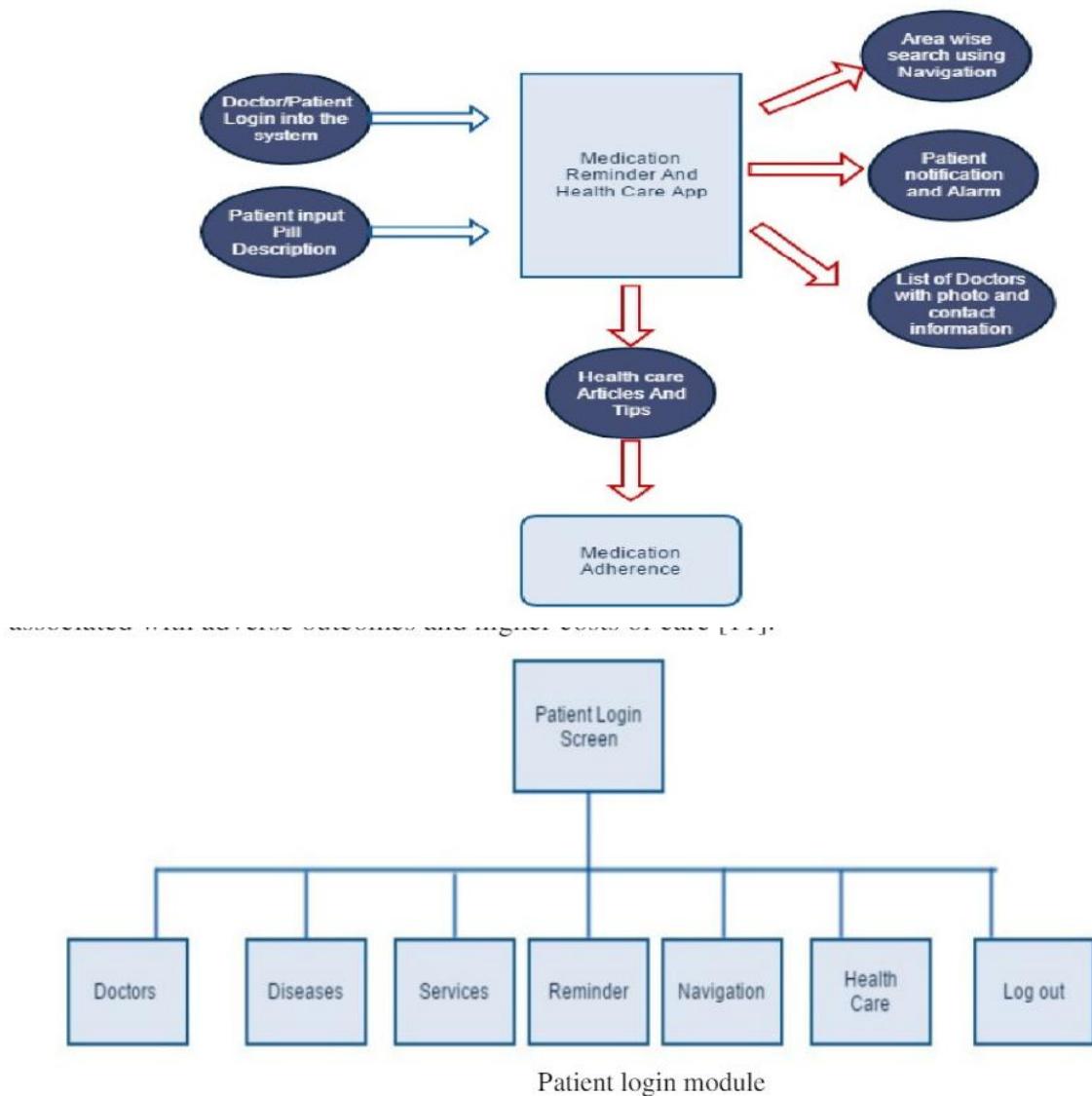


Fig 6.1 System Architecture Requirements

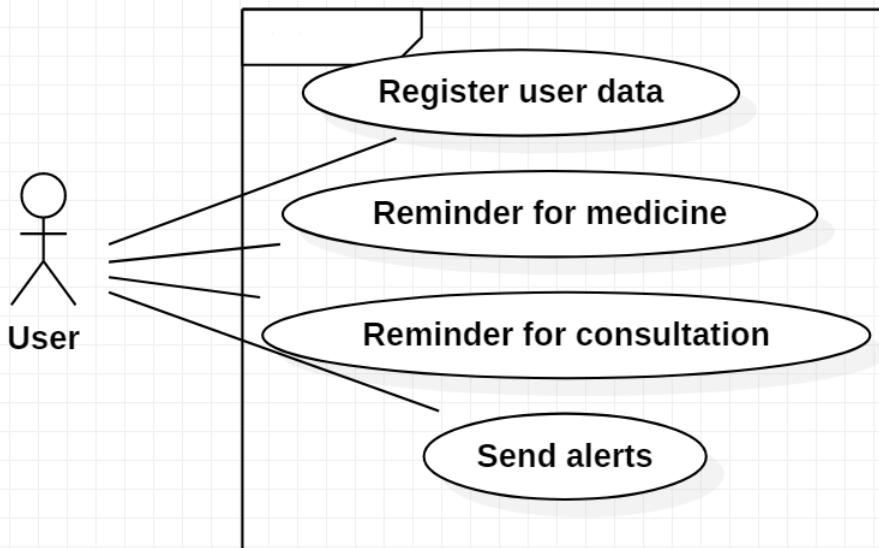


Fig 6.2 Use -Case Diagram

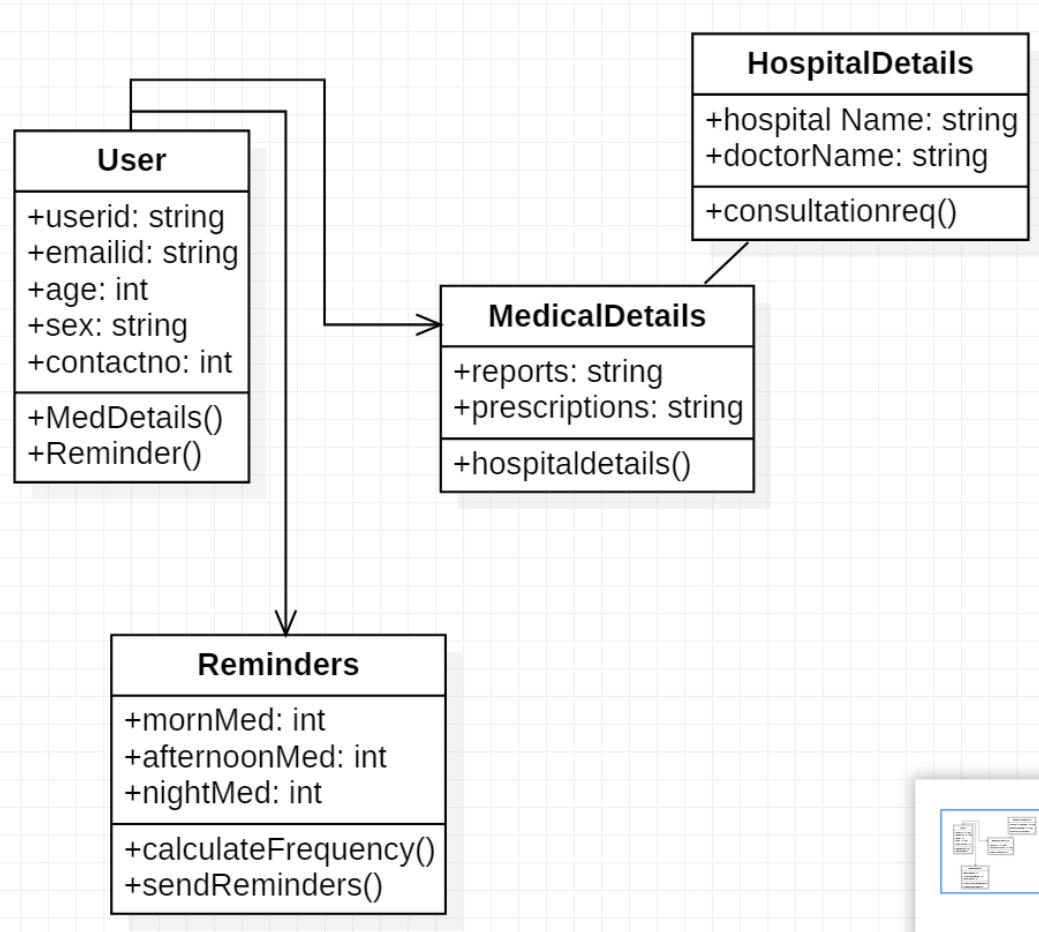


Fig 6.3 Class- diagram

CHAPTER 7

ENTITY RELATIONSHIP DIAGRAM

Entity relationship diagram describes the interrelated things of interest in domain, the diagram for the application of MedCare is as follows

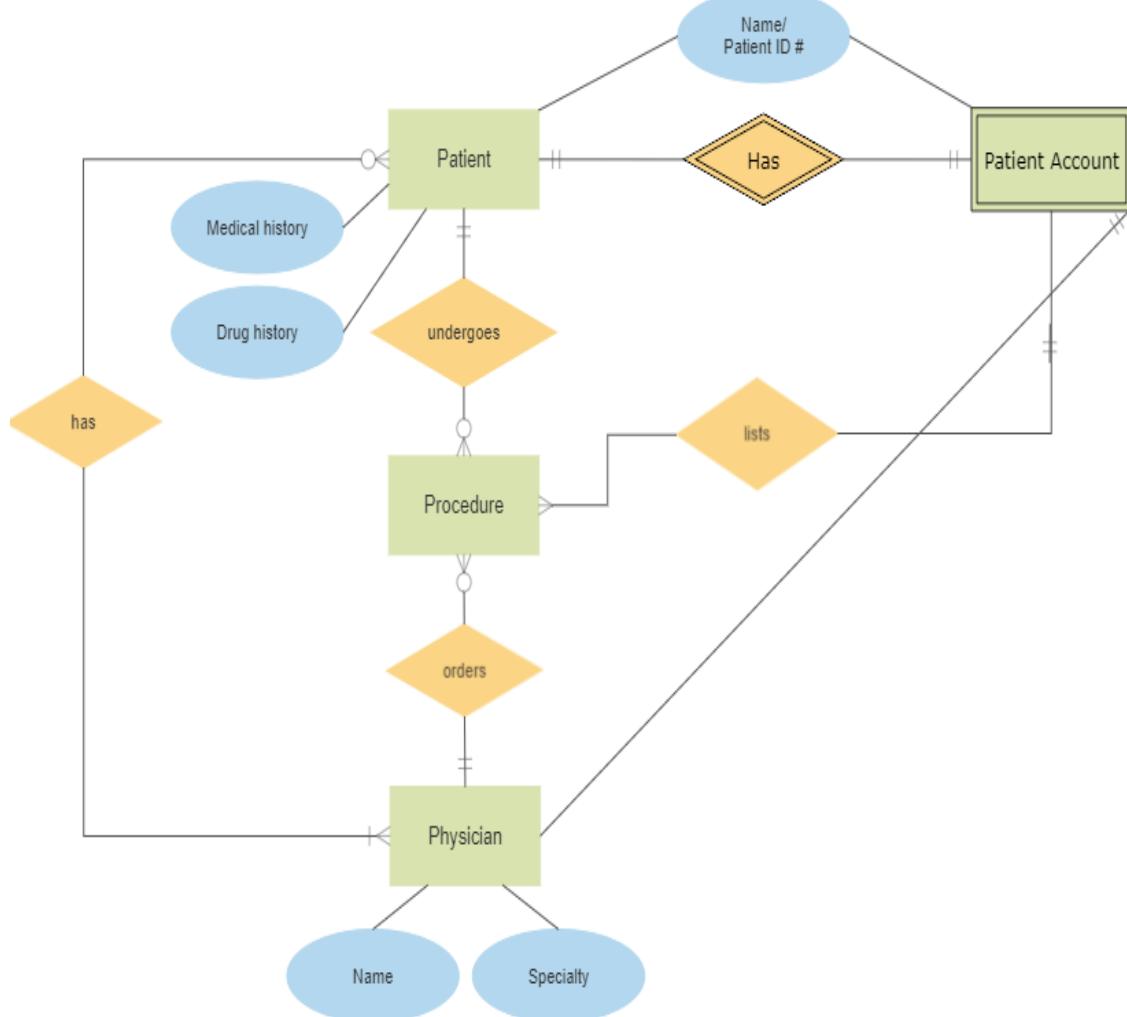


Fig 7.1 Entity Relationship diagram

CHAPTER 8

DATA FLOW DIAGRAM

Data flow diagram is a way of representing the flow of data through a process. The data flow diagram for the medicine reminder app, MedCare is as follows

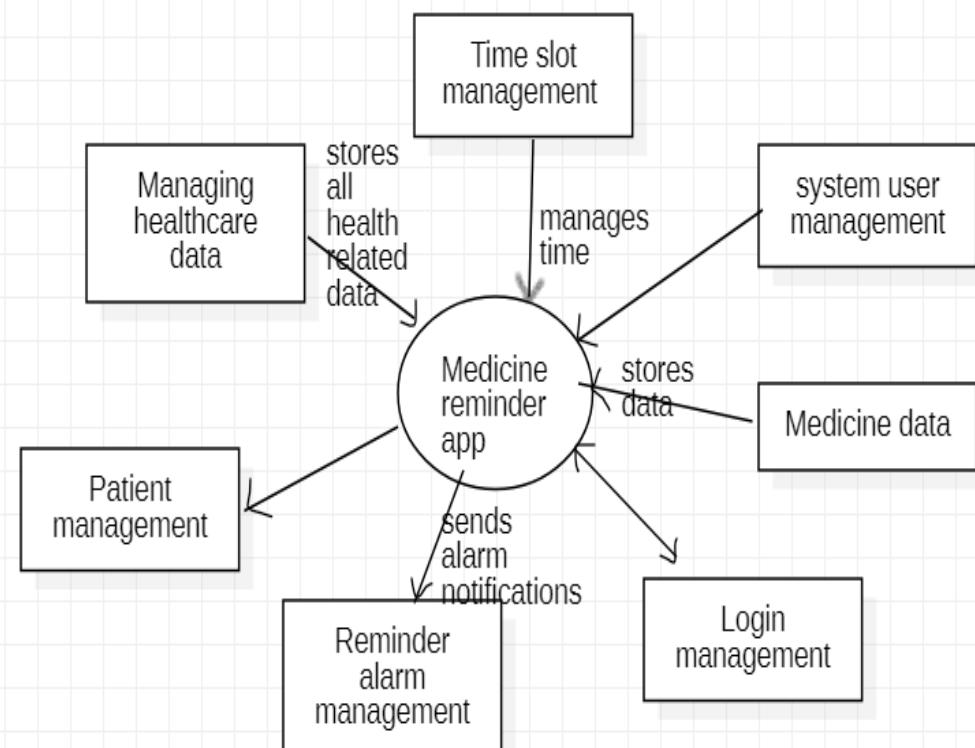


Fig 8.1 Data flow diagram

CHAPTER 9

SEQUENCE AND COLLABORATION DIAGRAM

Sequence Diagram:

Sequence diagram is a type of interaction diagram that describes how and in what order a group of objects work together. The sequence diagram for MedCare is as follows

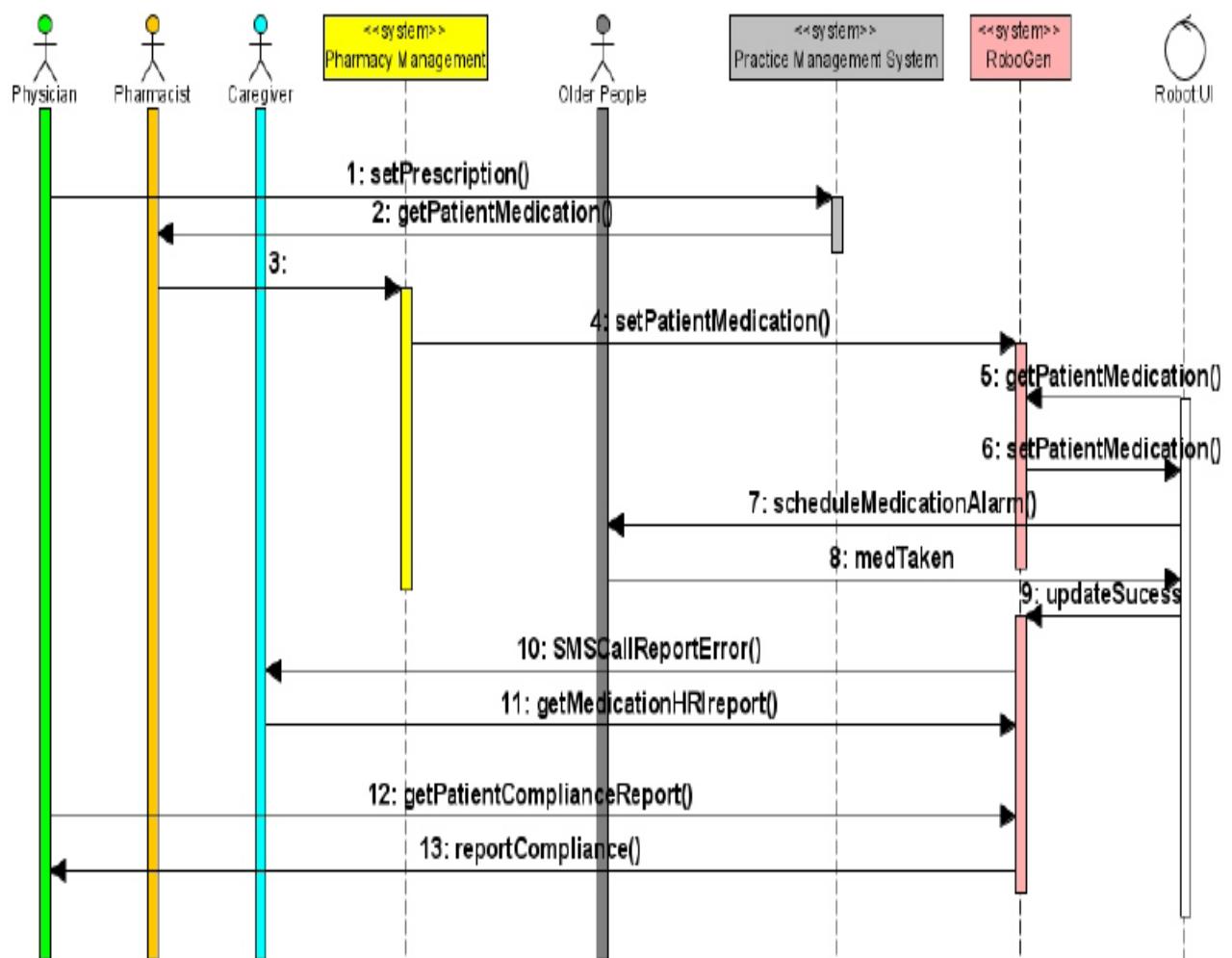


Fig 9.1 Sequence Diagram

Collaboration Diagram:

Collaboration diagram is an illustration of relationships and interactions among software objects. The collaboration diagram for MedCare is as follows

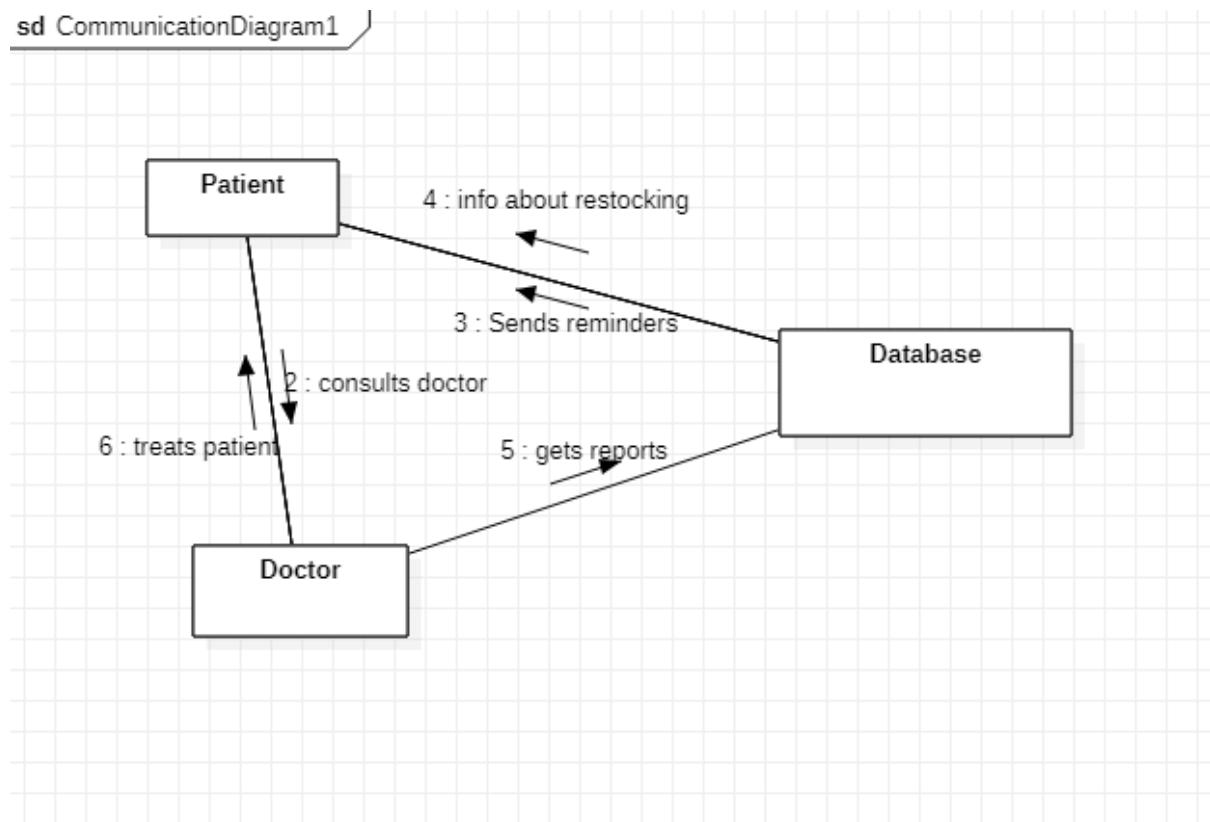


Fig 9.2 Collaboration diagram

CHAPTER 10

DEVELOPMENT OF TESTING FRAMEWORK/USER INETRFACE

Executive Summary:

Test Plan: Scope of Testing

Functional:

Functional Testing makes sure that all the applications that drive all the goodies that we use work exactly as they are supposed to. It is usually a form of Black Box Testing because generally, the testers do not really get into the internal program structure, or how the applications work. They only look at what goes in and what comes out-input and output. specific types of functional testings are:

Unit Testing

The most fundamental form of Functional Testing is Unit Testing. It evaluates the most basic elements or components of a software program-the smallest chunks of code that can be tested. Hence the name, Unit Testing. Each unit can be a module, function, method, or subroutine that takes a certain input and gives out a certain output. Test cases for Unit Testing evaluate fundamental aspects such as line, code, and method coverage. These refer to which lines or how much code or what methods were executed or "covered" during the test.

Integration Testing

Integration Testing is done when two, or generally more than two, modules or components are integrated together to develop an application. While the individual modules or components may function correctly on their own, Integration Testing ensures that the work together just as well.

User Acceptance Testing

Also known as Beta Testing, User Acceptance Testing (UAT) is a form of testing in which the application is tested by a small group of actual users or experts, representative of the actual end-users, in real-life situations. UAT provides valuable feedback to developers on whether or not the functionalities they create really meets users' needs and to what extent, what gaps or drawbacks they face, and what extra features they need to add to the application.

System Testing

In terms of hierarchy, System Testing generally follows Integration and precedes Beta Testing or UAT. In System Testing, the entire, fully integrated system is tested from end to end.

Non-Functional:

Non functional Testing is equally critical when it comes to meeting end-users' requirements because not only do applications need to "function" but they also need to "perform."

This takes into account factors such as performance under high loads, e.g., when thousands or even millions of users concurrently log into a system. It also considers the ability of the application to handle various forms of stress, environments, disasters, and so on.

The most common forms of non functional Testing include:

Load Testing

Load testing validates that the application responds as required even when a huge number of concurrent users access it simultaneously, such as in real-life situations. It is usually performed on dedicated servers that simulate actual usage environments.

Stress Testing

Stress Testing evaluates the performance of applications in crunch situations, e.g., under low memory/hard disk space conditions. In such environments, it is possible to detect defects that would not have been discovered in normal situations.

Recovery Testing

This checks whether applications recover gracefully when inputs are not as expected or when the environment fails. For instance, when a user types an invalid input that causes a database process to abort, or when systems shut down abnormally due to a power failure, and so on.

Security Testing

This simply checks whether an application has no flaws or vulnerabilities that can be exploited to compromise the system and lead to loss of data or theft. It focuses on testing authentication, access control, authorization, and other such sensitive processes.

Scalability Testing

Scalability Testing checks whether the application can handle an increase in user traffic, number of

transactions, processes, or volume of data. The application needs to scale up to meet such increases in demands.

Endurance Testing

Endurance Testing, also known as Soak Testing, checks whether the application can bear a sustained load over a long duration of time. Typically it is used to test for memory leaks in a system.

Reliability Testing

This form of testing can be used to check whether an application delivers the same output consistently over a specified duration. Reliability Testing is extremely vital in mission-critical applications such as aircraft systems, nuclear plant processes, and medical equipment, among others.

Objective to test software application

Software Testing has different goals and objectives. The major objectives of Software testing are as follows:

- Finding defects which may get created by the programmer while developing the software.
- Gaining confidence in and providing information about the level of quality
- To prevent defects.
- To make sure that the end result meets the business and user requirements
- To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications
- To gain the confidence of the customers by providing them a quality product.

Software testing helps in finalizing the software application or product against business and user requirements. It is very important to have good test coverage in order to test the software application completely and make it sure that it's performing well and as per the specifications.

Approach to test the software application:

Type of testing

Unit testing, Integration testing, Functional testing, System testing, Stress testing, Performance testing, Usability testing, Acceptance testing, Regression testing, Beta testing.

- Unit test - Verification of single classes.
- Module test - Testing interaction of groups of classes (package).
- Integration test - Verification of interactions between modules/components.

- Functional test - Verification of external behaviors of modules, components and system.
- System test - Testing of the system against objectives.
- Acceptance test - Validation of application against user requirements.
- Regression test - Re-running all tests on the system when it is changed.

Table 10.1 Test Plan

Requirements	Typical components	Detailed description
Introduction	Test strategy and approach	<ol style="list-style-type: none"> 1. Analyze the product or feature you're testing 2. Design the test strategies (and approach) you're going to use 3. Define the test objectives and pass/fail criteria 4. Plan the test environment 5. Execute your test plan and track progress in your project management tool
Walkthrough	Defects discovered and corrected	<p>Following are the types of defects found by the tools during static analysis:</p> <p>A variable with an undefined value</p> <p>Inconsistent interface</p>

		<p>between modules and components</p> <p>Variables that are declared but never used</p> <p>Unreachable code (or) Dead Core</p> <p>Programming standards</p> <p>Security vulnerabilities</p> <p>Syntax violations</p>
	Improved Ideas	<p>three best practices for effective use of a static code analysis tool to improve your output, including:</p> <ol style="list-style-type: none"> 1. How to effectively achieve thorough coverage with a static code analysis tool. 2. How to best use a static code analysis tool with mission-critical systems. 3. How to achieve gains through effective, automated

CHAPTER 11

TEST CASES MANUAL

Test Case Manual:

Functional test case manual

Non -Functional test case manual

Table 11.1 Functional Test Cases

Test ID (#)	Test Scenario	Test Case	Execution Steps	Expected Outcome	Status	Remarks
	Verify User Registration from India	Accept Valid India Mobile Number on the Page#1	1. User clicks on User Registration link 2. Enter the mobile Number on the text box 3. Click Register button	User should be taken to the next page for entering more user details	Pass	success
	Verify User Registration from India	Don't Accept Non IndianMobile Number on the Page#1	1. User clicks on User Registration link 2. Enter the mobile Number on the text box	Users should be told to enter a valid mobile number.	Pass	Success

	Email Verification	Accept the email id from the user if they allow to share it.	<ol style="list-style-type: none"> 1. Users get an option to share their email ids if they wish to receive reminders there as well. 2. They enter the email id and verify it. 3. Click on the register button. 	User should be taken to the next page for entering more user details	Pass	Success
	Verify that the user entered 10-digit mobile number.	Accept Valid Mobile Number on the Page#1	<ol style="list-style-type: none"> 1. User clicks on User Registration link 2. Enter the mobile Number on the text box 3. Click Register button 	User should be taken to the next page for entering more user details	Pass	Success

Table 11.2 Non-Functional Test Cases

Test ID (#)	Test Scenario	Test Case	Execution Steps	Expected Outcome	Status	Remarks
	Load Testing	All pages of the app should load within 3 seconds	All the pages are getting loaded within 3 seconds.	Pages are being loaded in 3 seconds	Pass	Success
	Usability testing	App should be easy to use	All the features of the app should be easy to use	App is user friendly	Pass	Success
	Features Testing	Live Help	The app should have an option	The app has a platform to ask	Pass	Success

			for the user to raise queries.	questions which are answered from time to time.		
	Features testing	Contact to doctors	The app should have the details of the doctor the person contacts.	The doctor details available on the app makes the person contact him directly without hustle.	Pass	Success
	Features Testing	Timely Reminders	The app should be linked with the clock and should provide timely reminders.	The app gives reminders to the customers whenever required as fed by them.	Pass	Success

CHAPTER 12

TEST CASE REPORTING

Table 12.1 Test case progress

Category	Progress Against Plan	Status
Functional Testing	Green	Completed
Non-Functional Testing	Green	Completed

Table 12.2 Test case coverage

Functional	Test Case Coverage (%)	Status
adding medicine details	100%	Completed
setting reminders	100%	Completed
planner	100%	Completed
user medicine data	80%	In-progress
user detail page(profile)	90%	In-progress

Some Obstacles:

In the user medicine data module, in some cases there is an error in integration when new medicines are entered to the time slot allotted for already existing ones.

In the user detail page, i.e. the profile some newly added features are yet to be tested.

CHAPTER 13

ARCHITECTURE DESIGN/FRAMEWORK AND IMPLEMENTATION

Object Oriented Architecture

In OOD, the technology-independent concepts in the analysis model are mapped onto implementing classes, constraints are identified, and interfaces are designed, resulting in a model for the solution domain. In a nutshell, a detailed description is constructed specifying how the system is to be built on concrete technologies.

The stages for object-oriented design can be identified as –

- Definition of the context of the system
- Designing system architecture
- Identification of the objects in the system
- Construction of design models
- Specification of object interfaces

Object-oriented system design involves defining the context of a system followed by designing the architecture of the system.

- Context – The context of a system has a static and a dynamic part. The static context of the system is designed using a simple block diagram of the whole system which is expanded into a hierarchy of subsystems. The subsystem model is represented by UML packages. The dynamic context describes how the system interacts with its environment. It is modelled using use case diagrams.
- System Architecture – The system architecture is designed on the basis of the context of the system in accordance with the principles of architectural design as well as domain knowledge. Typically, a system is partitioned into layers and each layer is decomposed to form the subsystems.

After the hierarchy of subsystems has been developed, the objects in the system are identified and their details are designed. Here, the designer details out the strategy chosen during the system design. The emphasis shifts from application domain concepts toward computer concepts. The objects identified during analysis are etched out for implementation with an aim to minimize execution time, memory consumption, and overall cost.

The screenshots of the application created:

The screenshots show a mobile application interface for managing medicine intake, divided into several sections:

- Top Left:** A circular icon featuring a blue background with various colored pills and a small clock.
- Top Middle:** A screen titled "New Medicine" showing fields for "Medicine Name" (paracetamol), "Dosage" (2.0 Tablet), "Start Date" (6/17/2022), "Duration" (5 Days), and "Frequency" (Daily). It includes a "Medicine Reminder" section with a plus sign and a reminder time of 10:07 AM.
- Top Right:** Another circular icon with pills and a clock.
- Middle Left:** A "Set Reminders" screen with instructions to never forget medicine times and mentions of family notifications. It has a "Next >" button.
- Middle Middle:** A "Planned" screen showing an "Upcoming" calendar for March 2022. The 13th is highlighted in green. Below the calendar, it says "Days to take pills".
- Middle Right:** A "Profile" screen for a user named "Chetas" (Patient) with options to "Add a care taker" and "Notification". It includes a "Log Out" button.
- Bottom Left:** A "My Pills" screen with a message "Hi Chetas" and "Keep it up! You are on the right track.", followed by a "Todays Medicines" section with a plus sign.
- Bottom Middle:** A "Report" screen with navigation icons for Home, Planned, Report, and Profile.
- Bottom Right:** A "New Medicine" screen identical to the one in the top middle, showing the same input fields and reminder settings.

Source Code

The screenshot shows two instances of Visual Studio Code side-by-side, both displaying Dart code for a Flutter application named "FLUTTER_PILL_Reminder_APP".

Left Window (patient_user.dart):

```
lib> models > patient_user.dart > ...
1 import 'dart:convert';
2
3 import 'package:cloud_firestore/cloud_firestore.dart';
4 import 'package:flutter/foundation.dart';
5
6 import 'package:flutter_damnakam_app/models/caretaker_user.dart';
7 import 'package:flutter_damnakam_app/models/medicine.dart';
8
9 class PatientUser {
10   String uid;
11   String name;
12   List<Medicine> medicinesList;
13   List<CaretakerUser> caretakerList;
14   List<String> tokens;
15   bool isActive;
16   DocumentReference? reference;
17   // fromSnapshot method to get data and reference.
18   factory PatientUser.fromSnapshot(DocumentSnapshot snapshot) {
19     PatientUser patientUser =
20       PatientUser.fromMap(snapshot.data() as Map<String, dynamic>);
21     patientUser.reference = snapshot.reference;
22     return patientUser;
23   }
24   PatientUser copyWith({
25     String? uid,
26     required String name,
27     required List<Medicine> medicinesList,
28     required List<CaretakerUser> caretakerList,
29     required List<String> tokens,
30     required bool isActive,
31     DocumentReference? reference,
32   });
33
34   PatientUser copyWith({
35     String? uid,
36     String? name,
37     List<Medicine>? medicinesList,
38     List<CaretakerUser>? caretakerList,
39   });
40 }
```

Right Window (anonymous_auth_service.dart):

```
lib> services > anonymous_auth_service.dart > ...
14
15 PatientUser? _patientUser;
16 String? _patientName;
17 String? _deviceToken;
18
19 setTheUID(String? theUID) {
20   _uid = theUID;
21   notifyListeners();
22 }
23
24 setTheName(String name) {
25   _patientName = name;
26   notifyListeners();
27 }
28
29 setTheUser(User? theUser) {
30   _user = theUser;
31   notifyListeners();
32 }
33
34 setTheToken(String token) {
35   _deviceToken = token;
36   notifyListeners();
37 }
38
39 Future signInAnonymously() {
40   get_DeviceToken();
41   return _firebaseAuth.signInAnonymously().then((theCredential) {
42     print('User signed in anonymously');
43     setTheUID(theCredential.user!.uid);
44     setTheUser(theCredential.user!);
45     get_DeviceToken();
46     savePatientToFirestore();
47   });
48 }
49
50 get_DeviceToken() {
51   Name non-constant identifiers using lowerCamelCase.
52   _firebaseMessaging.getTokens().then((tokens) {
53 }
```

Visual Studio Code Screenshot 1:

File Edit Selection View Go Run Terminal Help

add_medicine_screen.dart - flutter_pill_reminder_app - Visual Studio Code

```

23 class AddMedicineScreenState extends State<AddMedicineScreen> {
24   TextEditingController medicineNameController = TextEditingController();
25   TextEditingController durationNumberController = TextEditingController();
26   List<TimeOfDay> reminders = [];
27   TimeOfDay selectedTime = TimeOfDay.now();
28   bool isSelected = false;
29   bool isUnselected = false;
30   bool isNoneSelected = false;
31   bool isOneSelected = false;
32   bool isTwoSelected = false;
33   bool isThreeSelected = false;
34   bool isFourSelected = false;
35   bool isFiveSelected = false;
36   options.radioButtonValue = options.daily;
37   String dropdownValue = 'Days';
38   DateTime? selectedDate;
39   double numberOffTablets = 1;
40   @override
41   Widget build(BuildContext context) {
42     Size _screenSize = MediaQuery.of(context).size;
43     return Scaffold(
44       backgroundColor: whiteColor,
45       appBar: AppBar(
46         backgroundColor: whiteColor,
47         centerTitle: true,
48         elevation: 1,
49         title: Text(
50           'New Medicine',
51           style: Theme.of(context).textTheme.headline5!.copyWith(
52             color: darkGreen,
53             fontWeight: FontWeight.bold,
54           ),
55         ), // text
56       ), // AppBar
57       body: SafeArea(
58         child: SingleChildScrollView(
59           child: Padding(

```

Type here to search

File Edit Selection View Go Run Terminal Help

notify_helper.dart - flutter_pill_reminder_app - Visual Studio Code

```

1 Report: 'package:cloud_firestore/cloud_firestore.dart';
2 import 'package:flutter/material.dart';
3 import 'package:flutter_darmanakone_app/models/days.dart';
4 import 'package:flutter_darmanakone_app/models/frequencyType.dart';
5 import 'package:flutter_darmanakone_app/models/medicine.dart';
6 import 'package:flutter_darmanakone_app/screens/patientScreens/bottom_nav_handler.dart';
7 import 'package:flutter_local_notifications/flutter_local_notifications.dart';
8 import 'package:flutter_native_timezone/flutter_native_timezone.dart';
9 import 'package:rxdart/rxdart.dart';
10 import 'package:timezone/data/latest.dart' as tz;
11 import 'package:timezone/timezone.dart' as tz;
12
13 class NotifyHelper {
14   FlutterLocalNotificationsPlugin flutterLocalNotificationsPlugin =
15     FlutterLocalNotificationsPlugin();
16
17   String selectedNotificationPayload = '';
18
19   final BehaviorSubject<String> selectNotificationSubject =
20     BehaviorSubject<String>();
21   initializeNotification() async {
22     tz.initializeTimeZones();
23     // configureSelectNotificationSubject();
24     await _configureLocalTimeZone();
25     // await requestUserPermissions(FlutterLocalNotificationsPlugin());
26     final InitializationSettings initializationSettingsIOS =
27       InitializationSettingsIOS(
28         requestSoundPermission: false,
29         requestBadgePermission: false,
30         requestAlertPermission: false,
31         onDidReceiveLocalNotification: onDidReceiveLocalNotification,
32       );
33
34   const AndroidInitializationSettings initializationSettingsAndroid =
35     AndroidInitializationSettings('icon');
36
37   final InitializationSettings initializationSettings =
38     InitializationSettings(

```

Type here to search

The screenshot shows the Visual Studio Code interface with the file `OTP_verification_screen.dart` open. The code implements a stateless widget `OTPVerificationScreen` that extends `StatefulWidget`. It takes a `key` and a `required this.photo` parameter. The `build` method creates a `Scaffold` with a `body` containing a `SafeArea`, a `SingleChildScrollView`, and a `Column` with a centered `Text` field. The code includes several imports from the `flutter` and `flutter_damandokm` packages.

```
1 import 'package:flutter/material.dart'; // Name source files using "lowercase_with_underscores".
2 import 'package:flutter_damandokm/app/constants.dart';
3 import 'package:flutter_damandokm/app/screens/caretakerscreens/caretakers_invitation_screen.dart';
4 import 'package:flutter_damandokm/app/screens/patientscreens/bottom_nav_handler.dart'; unused imports: "package:flutter_damandokm/app/screens/patientscreens/bottom_nav_handler.dart"
5 import 'package/flutter_code_fields/pin_code_fields.dart';
6 import 'package:provider/provider.dart';
7
8 import '../../../../../services/authentication_service.dart';
9
10 class OTPVerificationScreen extends StatelessWidget {
11   const OTPVerificationScreen({Key? key, required this.photo})
12   : super(key: key);
13   final String phone;
14
15   @override
16   State<OTPVerificationScreen> createState() => _OTPVerificationScreenState();
17 }
18
19 class _OTPVerificationScreenState extends State<OTPVerificationScreen> {
20   TextEditingController _msController = TextEditingController(); Private field could be final.
21
22   @override
23   Widget build(BuildContext context) {
24     Size _screenSize = MediaQuery.of(context).size;
25     return Scaffold(
26       body: SafeArea(
27         child: SingleChildScrollView(
28           child: Column(
29             mainAxisAlignment: MainAxisAlignment.center,
30             mainAxisSize: MainAxisSize.min,
31             children: [
32               Text(
33                 "Enter code",
34                 style: TextStyle(
35                   color: Colors.black,
36                   fontSize: 18,
37                   fontWeight: FontWeight.w500,
38                 ),
39               ),
40             ],
41           ),
42         ),
43       ),
44     );
45   }
46 }
```

CONCLUSION

Hence, a medical reminder mobile application was made which gives timely reminders mostly helpful for the old aged population and the people who forget to take medicines on time.

REFERENCES

<https://www.geeksforgeeks.org/project-idea-meditrack/>

<https://www.onlinedoctor.com/best-medicine-reminder-apps/>

<https://www.healthline.com/health/best-medication-reminders>

http://www.cs.binghamton.edu/~steflik/wiki/index.php/Medicine_Reminder_App