MedDeliver

A Medicine Delivery and Scheduling Android Application

ومن يؤت الحكمة
فقد أوتي خيرا كثيرا

QUAID-I-AZAM UNIVERSITY
ISLAMABAD

By

Natasha Javed

Supervised By

Mr. Syed Muhammad Naqi

Department of Computer Science,

Quaid-i-Azam University, Islamabad, Pakistan.

Session (2015 – 2019)

QUAID-I-AZAM UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

Dated: Oct 07, 2019

FINAL APPROVAL

This is to certify that we have read the project report submitted by Ms. Natasha Javed it is our judgment that this report is of sufficient standard to warrant its acceptance by the Quaid-i-Azam University, Islamabad for the degree of the Bachelors of Science in Computer Sciences.

COMMITTEE

1. EXTERNAL EXAMINER

Dr. Amir Qayyum
Professor
Capital University of Science &
Technology (CUST)
Islamabad

2. SUPERVISOR

Syed M. Naqi
Lecturer
Deptt. of Computer Sciences
Quaid-i-Azam University
Islamabad.

3. Chairperson

Dr. Onaiza Maqbool
Associate Professor
Deptt. of Computer Sciences
Quaid-i-Azam University
Islamabad.

ACKNOWLEDGEMENT

All the praises, thanks and acknowledgments are for the creator Allah Almighty, the most Beneficent, the most Merciful, who gave me strength and enabled me to undertake and execute this task. Countless salutations upon the Holy Prophet Hazrat Muhammad (S.A.W), source of knowledge for enlightening with the essence of faith in Allah and guiding the mankind, the true path of life. I would like to express my gratitude to my Project Supervisor, Mr. Syed Muhammad Naqi Co-Supervisor Mr. Hanny Tufail for their constant guidance and motivation throughout the development of the project. I am sincerely grateful to them for always being there to solve issues related to the project.

In addition, I would also like to express my gratitude to my family for their enormous support, to all my teachers who made me who I am today and my friends for always encouraging and believing in me.

Natasha Javed

Abstract

MedDeliver is an android application which will facilitate people by providing them genuine medicines from authentic pharmacies at their doorstep. They will be able to make an order for a prescription once and get it delivered regularly by setting a schedule on it like weekly, fortnight or monthly. This scheduling will primarily be helpful for people with diseases like Blood Pressure, Diabetes for whom it's crucial to take medicines regularly and on time. MedDeliver will not only deliver medicines on time but will remind people to take them on time and regularly as well, they will be able to do so by setting an alarm for different dosages of medicines.

Additionally, this app will allow patients to log their condition after every dosage. This feature will bridge communication between patient and doctor by letting the doctor know about how you felt after every dosage, making it easier for him to see how you've been doing, what effects or side effects the medicine had on you and what the next move in your treatment should be.

MedDeliver will be your partner helping you through your treatment and recovery so you go on to live a healthy life.

Table of Contents

List of Figures

List of Tables:

Chapter1

Software Project Management Plan

1.1 Introduction

This chapter describes roles, responsibilities, processes and schedules for managing the software development process, in addition to outlining the tools, methods, and procedures to be used.

1.1.1 Project Overview

The system under development will be an android app. MedDeliver will facilitate people by providing them genuine medicines from authentic pharmacies at their doorstep. They will be able to make an order for a prescription once and get it delivered regularly by setting a schedule on it like weekly, fortnight or monthly. This scheduling will primarily be helpful for people with diseases like Blood Pressure, Diabetes for whom its crucial to take medicines regularly and on time. MedDeliver will also try and ensure that not only medicines reach the patient in time but they take it on time as well, for this they will be able to set an alarm for every dosage and then be reminded of it. Additionally, this app will allow patients to log their condition after every dosage. This feature will bridge communication between patient and doctor by letting the doctor know about how you felt after every dosage, making it easier for him to see how you've been doing, what effects or side effects the medicine had on you and what the next move in your treatment should be. MedDeliver will try and be your partner helping you through your treatment and recovery so you go on to live a healthy llife.

1.1.2 Project Deliverables

Deliverable is a tangible output of human effort provided by a developer to a customer. These deliverables are delivered to the project supervisor. The deliverables for the project "MedDeliver" are as follows:

Deliverables

- 1st Phase SPMP,SRS
- 2nd Phase SDD, STD
- 3rd Phase Application

1.2 Project Organization

This section details the architecture of the project, including: the process model, organizational structure, interfaces, and project responsibilities.

1.2.1 Software Process Model

In this project, Agile Software Development model with Prototyping will be used because

- Requirements are not fixed and are likely to change and evolve, agile model is adaptive to such change.
- When requirements are likely to change or an idea is new it's a better approach to visualize it somehow that's where prototyping comes in role, allowing customers to see what the product might actually be.
- When requirements will change it will be easier to incorporate them into prototype first then moving into development.

1.2.2 Roles and Responsibilities

In this Project, I am responsible for developing the Project Plan and responsible for communication, status reporting, developing the software product, and, in general, making sure the project is delivered on schedule and within scope. The Project supervisor guides in major issues, problems, and policy conflicts and removes obstacles.

1.2.3 Tools and Techniques

Following are the tools and techniques used for this project.

Tools and Techniques

- MS Word 2013 → Used for documentation purposes
- ArgoUML v.33.4 → Used for making diagrams
- Java SDK → Required for android studio
- Android Studio Kotlin→ Used for coding of client-side application

Firebase Database used for storage purposes. ProjectLibre 1.7 Used for making project plan

1.3 Project Management Plan

This section describes how the project will be manage, what are its tasks, deliverables, milestones etc.

1.3.1 Tasks

1.3.1.1 Problem Understanding

- **Description:**

  First problem definition is must.

- **Deliverables and Milestones:**

  None

- **Resources Needed:**

  The following table shows the resources required to complete the task:

People	Natasha Javed Supervisor

- **Dependencies and Constraints:**

  None

- **Risks and Contingencies**

  None

1.3.1.2 Software Project Management Plan

- **Description:**

  Secondly software approach and milestones are identified.

- **Deliverables and Milestones:**

  None

- **Resources Needed:**

  The following table shows the resources required to complete the task:

People	Natasha Javed Supervisor

Software	MS Word Project Libre
Hardware	Laptop

- **Dependencies and Constraints:**

  Problem Understanding

- **Risks and Contingencies:**

  None

1.3.1.3 Software Requirement Specification

- **Description:**

  Thirdly analysis on how the requirements will meet is included.

- **Deliverables and Milestones:**

  SPMP and SRS Document.

- **Resources Needed:**

  The following table shows the resources required to complete the task:

People	Natasha Javed Supervisor
Software	MS Word
Hardware	Laptop

- **Dependencies and Constraints:**

  SPMP

- **Risks and Contingencies:**

  None

1.3.1.4 Software Design Description

- **Description:**

  Fourthly detailed design and interface design will be included.

- **Deliverables and Milestones:**

  None

- **Resources Needed:**

  The following table shows the resources required to complete the task:

People	Natasha Javed Supervisor
Software	MS Word
Hardware	Laptop

- **Dependencies and Constraints:**

  Analysis and Requirement.

- **Risks and Contingencies:**

  None

1.3.1.5 Software Test Documentation

- **Description:**

  In this part test plans and test case to verify and validate the system and their results will be included

- **Deliverables and Milestones:**

  Software Design Documentation and Software Test Documentation

- **Resources Needed:**

People	Natasha Javed Supervisor
Software	MS Word
Hardware	Laptop

- **Dependencies and Constraints:**

  SDD

- **Risks and Contingencies:**

  None

1.3.1.6 Software Implementation

- **Description:**

  How the system will be implemented.

- **Deliverables and Milestones:**

  Complete Implementation

- **Resources Needed:**

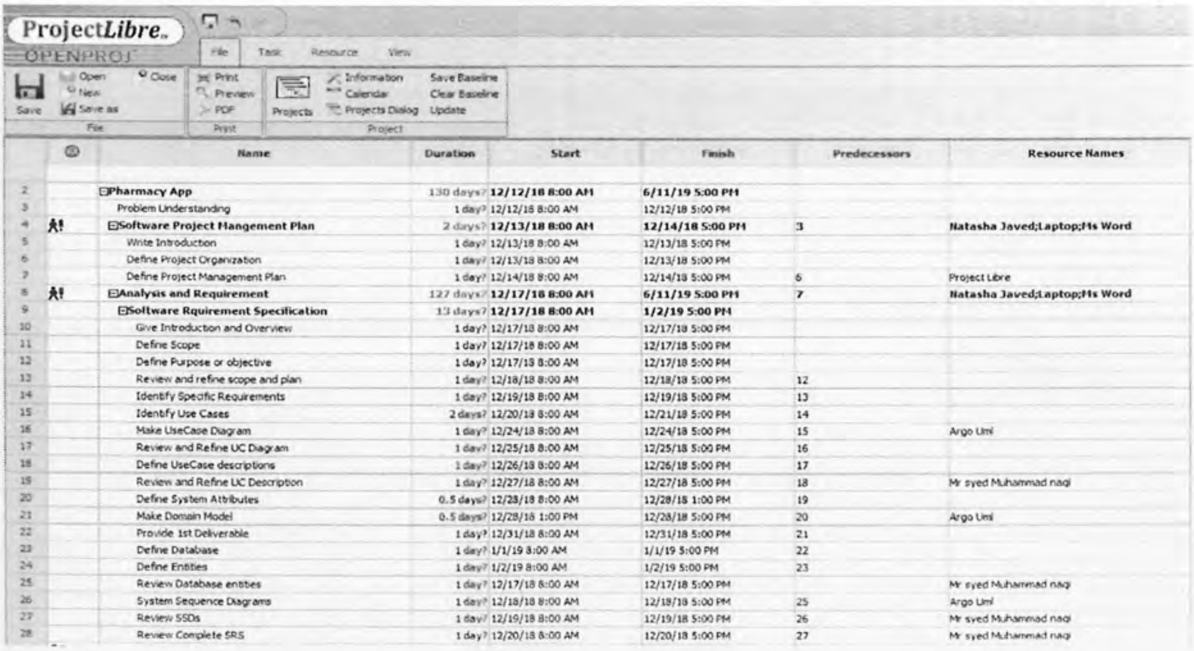
People	Natasha Javed Supervisor
Software	Android Studio, IntelliJ IDEA
Hardware	Laptop

- **Dependencies and Constraints:**

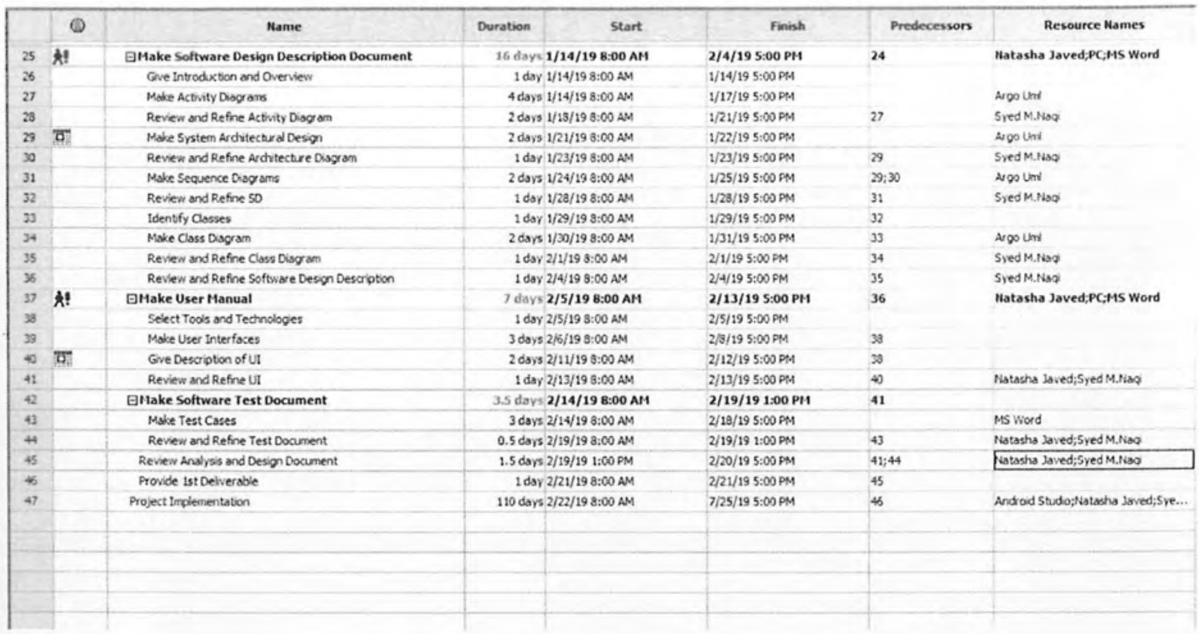
  STD

- **Risks and Contingencies:**

  None



		Name	Duration	Start	Finish	Predecessors	Resource Names
2		**⊟Pharmacy App**	130 days?	**12/12/18 8:00 AM**	**6/11/19 5:00 PM**		
3		Problem Understanding	1 day?	12/12/18 8:00 AM	12/12/18 5:00 PM		
4		**⊟Software Project Mangement Plan**	2 days?	**12/13/18 8:00 AM**	**12/14/18 5:00 PM**	3	**Natasha Javed;Laptop;Ms Word**
5		Write Introduction	1 day?	12/13/18 8:00 AM	12/13/18 5:00 PM		
6		Define Project Organization	1 day?	12/13/18 8:00 AM	12/13/18 5:00 PM		
7		Define Project Management Plan	1 day?	12/14/18 8:00 AM	12/14/18 5:00 PM	6	Project Libre
8		**⊟Analysis and Requirement**	127 days?	**12/17/18 8:00 AM**	**6/11/19 5:00 PM**	7	**Natasha Javed;Laptop;Ms Word**
9		**⊟Software Rquirement Specification**	13 days?	**12/17/18 8:00 AM**	**1/2/19 5:00 PM**		
10		Give Introduction and Overview	1 day?	12/17/18 8:00 AM	12/17/18 5:00 PM		
11		Define Scope	1 day?	12/17/18 8:00 AM	12/17/18 5:00 PM		
12		Define Purpose or objective	1 day?	12/17/18 8:00 AM	12/17/18 5:00 PM		
13		Review and refine scope and plan	1 day?	12/18/18 8:00 AM	12/18/18 5:00 PM	12	
14		Identify Specific Requirements	1 day?	12/19/18 8:00 AM	12/19/18 5:00 PM	13	
15		Identify Use Cases	2 days?	12/20/18 8:00 AM	12/21/18 5:00 PM	14	
16		Make UseCase Diagram	1 day?	12/24/18 8:00 AM	12/24/18 5:00 PM	15	Argo Uml
17		Review and Refine UC Diagram	1 day?	12/25/18 8:00 AM	12/25/18 5:00 PM	16	
18		Define UseCase descriptions	1 day?	12/26/18 8:00 AM	12/26/18 5:00 PM	17	
19		Review and Refine UC Description	1 day?	12/27/18 8:00 AM	12/27/18 5:00 PM	18	Mr syed Muhammad naqi
20		Define System Attributes	0.5 days?	12/28/18 8:00 AM	12/28/18 1:00 PM	19	
21		Make Domain Model	0.5 days?	12/28/18 1:00 PM	12/28/18 5:00 PM	20	Argo Uml
22		Provide 1st Deliverable	1 day?	12/31/18 8:00 AM	12/31/18 5:00 PM	21	
23		Define Database	1 day?	1/1/19 8:00 AM	1/1/19 5:00 PM	22	
24		Define Entities	1 day?	1/2/19 8:00 AM	1/2/19 5:00 PM	23	
25		Review Database entities	1 day?	12/17/18 8:00 AM	12/17/18 5:00 PM		Mr syed Muhammad naqi
26		System Sequence Diagrams	1 day?	12/18/18 8:00 AM	12/18/18 5:00 PM	25	Argo Uml
27		Review SSDs	1 day?	12/19/18 8:00 AM	12/19/18 5:00 PM	26	Mr syed Muhammad naqi
28		Review Complete SRS	1 day?	12/20/18 8:00 AM	12/20/18 5:00 PM	27	Mr syed Muhammad naqi



	ⓘ	Name	Duration	Start	Finish	Predecessors	Resource Names
25		**⊟Make Software Design Description Document**	16 days	**1/14/19 8:00 AM**	**2/4/19 5:00 PM**	**24**	**Natasha Javed;PC;MS Word**
26		Give Introduction and Overview	1 day	1/14/19 8:00 AM	1/14/19 5:00 PM		
27		Make Activity Diagrams	4 days	1/14/19 8:00 AM	1/17/19 5:00 PM		Argo Uml
28		Review and Refine Activity Diagram	2 days	1/18/19 8:00 AM	1/21/19 5:00 PM	27	Syed M.Naqi
29		Make System Architectural Design	2 days	1/21/19 8:00 AM	1/22/19 5:00 PM		Argo Uml
30		Review and Refine Architecture Diagram	1 day	1/23/19 8:00 AM	1/23/19 5:00 PM	29	Syed M.Naqi
31		Make Sequence Diagrams	2 days	1/24/19 8:00 AM	1/25/19 5:00 PM	29;30	Argo Uml
32		Review and Refine SD	1 day	1/28/19 8:00 AM	1/28/19 5:00 PM	31	Syed M.Naqi
33		Identify Classes	1 day	1/29/19 8:00 AM	1/29/19 5:00 PM	32	
34		Make Class Diagram	2 days	1/30/19 8:00 AM	1/31/19 5:00 PM	33	Argo Uml
35		Review and Refine Class Diagram	1 day	2/1/19 8:00 AM	2/1/19 5:00 PM	34	Syed M.Naqi
36		Review and Refine Software Design Description	1 day	2/4/19 8:00 AM	2/4/19 5:00 PM	35	Syed M.Naqi
37		**⊟Make User Manual**	7 days	**2/5/19 8:00 AM**	**2/13/19 5:00 PM**	**36**	**Natasha Javed;PC;MS Word**
38		Select Tools and Technologies	1 day	2/5/19 8:00 AM	2/5/19 5:00 PM		
39		Make User Interfaces	3 days	2/6/19 8:00 AM	2/8/19 5:00 PM	38	
40		Give Description of UI	2 days	2/11/19 8:00 AM	2/12/19 5:00 PM	38	
41		Review and Refine UI	1 day	2/13/19 8:00 AM	2/13/19 5:00 PM	40	Natasha Javed;Syed M.Naqi
42		**⊟Make Software Test Document**	3.5 days	**2/14/19 8:00 AM**	**2/19/19 1:00 PM**	**41**	
43		Make Test Cases	3 days	2/14/19 8:00 AM	2/18/19 5:00 PM		MS Word
44		Review and Refine Test Document	0.5 days	2/19/19 8:00 AM	2/19/19 1:00 PM	43	Natasha Javed;Syed M.Naqi
45		Review Analysis and Design Document	1.5 days	2/19/19 1:00 PM	2/20/19 5:00 PM	41;44	Natasha Javed;Syed M.Naqi
46		Provide 1st Deliverable	1 day	2/21/19 8:00 AM	2/21/19 5:00 PM	45	
47		Project Implementation	110 days	2/22/19 8:00 AM	7/25/19 5:00 PM	46	Android Studio;Natasha Javed;Sye...

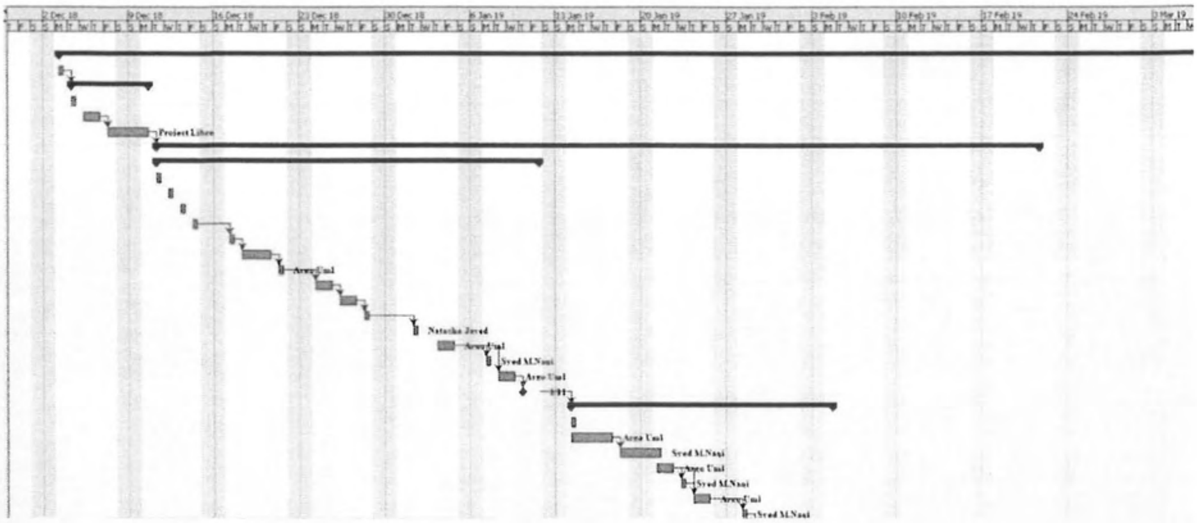


Figure 1.1 Gantt Chart

Chapter 2
Software Requirement Specification

2.1 Introduction

This chapter aims at defining the overall Software Requirements for "MedDeliver". This document depicts the features the system is to have.

2.1.1 Purpose

Primarily, the purpose of MedDeliver is to automate the process of medicine delivery as per patient's prescription and ensures genuine medicines reach the patient directly from authentic pharmacy in timely manner, this will save people time and commute to pharmacies. Secondly, people can benefit from MedDeliver by logging their condition. Whenever a patient starts new medication he/she faces some effects and he/she has to report the effects to the doctors but there is a chance you might miss some details or don't remember exactly when you felt a certain symptom which could be crucial in some cases. MedDeliver will automate this process for patients by allowing them to log their condition after every medicine dosage so a complete track is maintained throughout their medicine course. In this way not only they can show this to their doctor but can also self-track their condition.

2.1.2 Scope

MedDeliver is a combination of two applications where the admin will be using the android tablet application whereas customer will be using smartphone android application.

Customer:

- Allowing the user to upload their prescription.
- Users can schedule delivery of medicines.
- User can set alarms for medicine intake time(s).
- Allowing user to log their condition after medicine intake
-

Admin:

- Can view schedules of all orders and can search them by date.
- Admin then can update status of orders such as Confirmed, Delivered.
- Admin can view all the orders delivered.

2.1.3 Overview

This Chapter focuses on the requirements of system, their elicitation and analysis, which requirements are functional and which requirements are non-functional. In addition to this, specifying any constraints that the system is to follow.

2.2 Overall Description

This section gives an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It also describes what type of stakeholders use the system and what functionality is available for each type.

2.2.1 Product Perspective

MedDeliver is an android application that is intended to run on an android platform. It has two applications one for admin one for users.

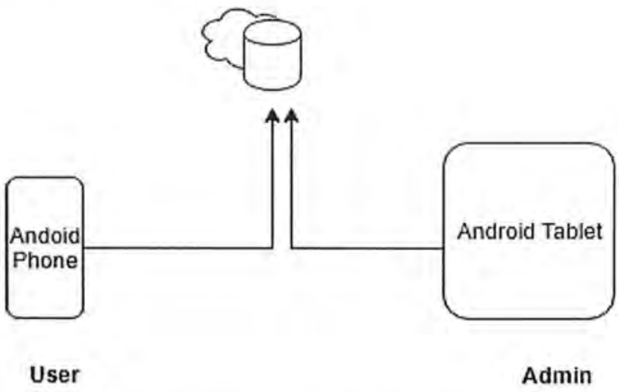


Figure 2. 1 System Block Diagram

2.2.2 User Characteristics

- Users must know how to use a smartphone.
- Users should be able to use system functionalities.

2.2.3 General Constraints

- Application is to be installed in a smartphone
- Only registered users can use the application
- Application should be easy to use and understand

2.2.4 Assumptions and Dependencies

- System is dependent on internet in order to update to update database.
- Adequate knowledge of English as interface is in English language.

2.3 External Interfaces Requirements

2.3.1 User Interfaces

- Interfaces follow HCI rules such consistency, visibility etc.
- Interfaces are self-explanatory an easy to interact with.
- There are options like prescription, payment etc. and clicking each will take you to respective screen.

2.3.2 Hardware Interfaces

- Application will run on smartphone
- Internet is needed for prescription and delivery purpose.

2.3.3 Software Interfaces

Platform- Android Studio 3.1.2 API 28 version 4

Emulator-Nexus 5X API 28, 10.1 WXGA API 28

Technologies used: Kotlin

Database: Firebase

2.3.4 Communication Interfaces

Http protocol be used as communicational interface.

2.4 Software Product Features

Register account:

This function allows admin as well as Customers to register their accounts.

Login account:

This function allows admin and Customers to login accounts.

Upload Prescription:

This function allows entering of medicine names, type, their dosage, their gramage, per day intake, and medicine course period.

Scheduling of delivery:

This function allows specifying the period after which medicine is to be delivered by specifying a duration i.e. weekly, fortnight, monthly.

Set alarm for medicine taking time:

This function will allow setting alarms for medicine intake time e.g. 9 am, 2pm, 7pm depending upon your prescription.

Log condition of Customer after medicine intake:

A function allowing Customer to log his/her condition after medicine intake. Customer will be able to select from 5 emojis

2.5 Usecase Model

Customer:

1. Login
2. Logout

3. Register account
4. Update account
5. Delete account
6. View Account
7. Add prescription
8. Update prescription
9. Delete prescription
10. View prescription
11. Schedule delivery
12. Update schedule
13. Delete schedule
14. View Schedule
15. Set alarm for medicine intake
16. Update alarm for medicine intake
17. Delete alarm for medicine intake
18. Turn alarm off for medicine intake
19. Log condition after medicine intake
20. Track order
21. Update Status of order

Admin

1. Login
2. Logout
3. Register account
4. Update account
5. Delete account
6. Add prescription
7. Update prescription
8. Delete prescription
9. View prescription
10. Schedule delivery

11. Update schedule
12. Delete schedule
13. View Schedule
14. Update order status

Special requirements

1. Android based smartphone

2. Text must be visible to the users.

Stakeholders and interests

Admin: Admin wants to keep track of all Schedules for delivering medicines on time.

User: User wants delivery of medicine on time along with being reminded of taking medicine and then logging his/her condition after intake

2.5.1 Use case Diagram

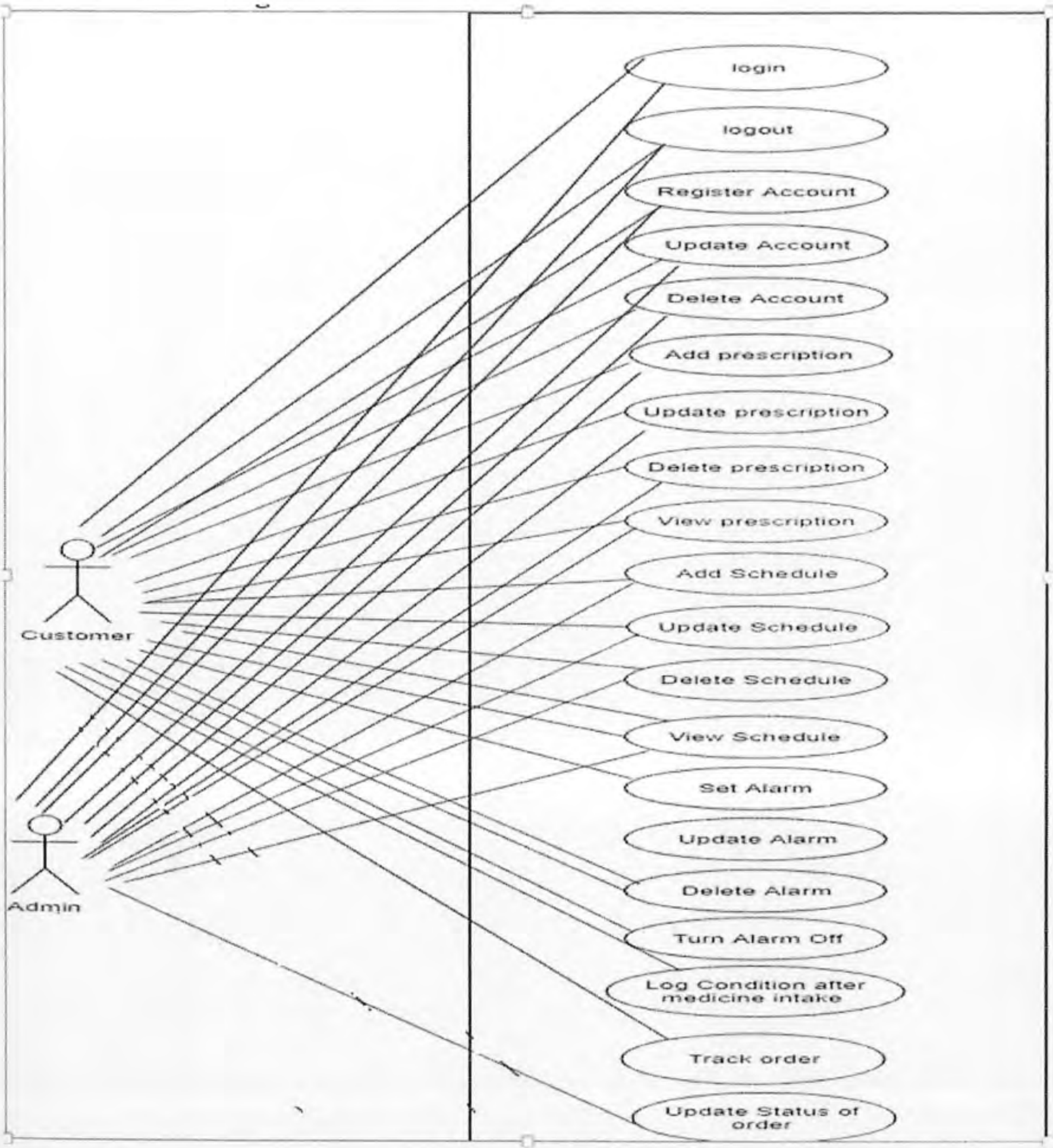


Figure 2.2 Use Case Diagram

2.5.2 Use cases Description

The full description of use cases is given below:

Table 1.1 UCD for Login

UCD - 1 Login	
Primary Actor:	Customer, Admin
Pre-Condition:	Customer, Admin must have an account
Post-Condition:	Customer, Admin has logged in successfully
Main success scenario:	1. User opens application 2. The System prompts the user for username and password or register new account. 3. The user enters username and password. 4. User presses login button. 5. The system validates the entered information then redirects the user to home page.
Alternative scenario	a. Internet is not available. b. Android OS is crashed. 1a User enters wrong credentials. 1. System prompts the user to enter correct information. 2. User re-renter's information. 2a. User submits information without filling all required fields. 1. System prompts the user to fill all required fields

Special requirements	Android phone or tablet
Technology	None
Frequency	Many times, a day.

Table 2.2 UCD for Logout

UCD- 2 Logout	
Primary Actor:	Customer, Admin
Pre-Condition:	Customer, Admin must be logged in
Post-Condition:	Customer, Admin has logged out successfully
Main success scenario:	1. User presses logout button. 2. System displays login screen.
Alternative scenario	a. Android OS is crashed.
Special requirements	Android phone or tablet
Technology	None
Frequency	Many times, a day.

Table 2.3 UCD for Register Account

UCD-3 Register Account	
Primary Actor:	Customer, Admin
Pre-Condition:	Customer, Admin have their applications installed
Post-Condition:	Customer, Admin is registered

Main success scenario:	1. User presses the create account button. 2. System prompts for inputs name, contact number, address, username and password fields 3. User enters into the fields. 4. User presses Submit button. 5. System saves user details and displays successfully created account message.
Alternative scenario	a. Android OS is crashed. b. Internet is not available. 4a. User submits information without filling all required fields. 1. System prompts the user to fill all required fields
Special requirements	Android phone or tablet
Technology	None
Frequency	Many times a day

Table 2.4 UCD for Update Account

UCD-4 Update Account	
Primary Actor:	Customer, Admin
Pre-Condition:	Customer/Admin must be logged in.
Post-Condition:	Customer/Admin updated the account successfully
Main success scenario:	1. User selects profile 2. System open profile

	3. User edits fields of his choice and presses save button. 4. System saves the user edits.
Alternative scenario	a. Android OS is crashed. b. Internet is not available. 3a. User submits information without filling all required fields. 1. System prompts the user to fill all required fields 3b. User wants to change password 1. System asks to verify password first. 2. User enters previous password 3. System verifies password and prompts to enter new one.
Special requirements	Android phone or tablet
Technology	None
Frequency	Many times a day

Table 2.5 UCD for Delete Account

UCD-5 Delete Account	
Primary Actor:	Customer, Admin
Pre-Condition:	User is logged on to the system.
Post-Condition:	Account has been successfully deleted.
Main success scenario:	1. User selects profile. 2. System open profile. 3. User selects delete account.

	4. System prompts for confirmation. 5. User selects confirm deletion. 6. System deletes the account.
Alternative scenario	a. Internet is not available. b. Android OS is crashed. 5a User not confirms the operation
Special requirements	Android phone or tablet
Technology	None
Frequency	Many times, a day.

Table 2.6 UCD for View Account

UCD-6 View Account	
Primary Actor:	Customer, Admin
Pre-Condition:	User is logged on to the system.
Post-Condition:	Account has been successfully Viewed.
Main success scenario:	1. User selects profile. 2. System displays contents of profile such as name, contact number, address, email and password
Alternative scenario	a. Android OS is crashed.
Special requirements	Android phone or tablet
Technology	None
Frequency	Many times, a day.

Table 2.7 UCD for Add Prescription

UCD-7 Add Prescription	
Primary Actor:	Customer, Admin
Pre-Condition:	Customer is logged on to the system.
Post-Condition:	Prescription with medicine names, type, their dosage, their gramage, per day intake, and medicine course period is saved successfully
Main success scenario:	1. User selects add prescription option 2. System prompts to upload prescription picture 3. User uploads picture. 4. System opens a form with medicine name, type, their dosage, their gramage, per day intake, and medicine course period 5. User enters the fields 6. System saves the fields User repeats step 5 if more than one medicine is to be added and system repeats step 6 consequently. 7. Include Scheduling Delivery Use case 8. System generates PDF view of prescription with customer name, Estimated delivery Date, list of medicines with their per day intake, company name and phone number

Alternative scenario	a. Internet is not available. b. Android OS is crashed. 3a.User enters wrong type of value in any field 1. System rejects the entry and doesn't field contents 2. System prompts re-entry.
Special requirements	Android phone or tablet
Technology	None
Frequency	Many times, a day.

Table 2.8 UCD for Update Prescription

UCD-8 Update Prescription	
Primary Actor:	Customer, Admin
Pre-Condition:	Customer is logged on to the system.
Post-Condition:	Prescription is updated successfully
Main success scenario:	1. User selects prescription to edit 2. System opens prescription. 3. User edits fields of his choice and presses save button 4. System saves the user edits.
Alternative scenario	a. Internet is not available. b. Android OS is crashed. 3a.User enters wrong type of value in any field

	System rejects the entry and doesn't field contents 3b. User tries to update a schedule that is already confirmed 1. System prompts with message that order is confirmed. 3. System prompts re-entry.
Special requirements	Android phone or tablet
Technology	None
Frequency	Many times, a day.

Table 2.9 UCD for Delete Prescription

UCD-9 Delete Prescription	
Primary Actor:	Customer, Admin
Pre-Condition:	Customer is logged on to the system.
Post-Condition:	Prescription is deleted successfully
Main success scenario:	1. User selects prescription option 2. System displays prescription list. 3. User selects prescription to delete presses delete button 4. System prompts for confirmation. 5. User selects confirm deletion. 6. System deletes the prescription.

Alternative scenario	a. Internet is not available. b. Android OS is crashed. 3a User not confirms the operation 1. System prompts re-entry. 3b. User tries to update a schedule that is already confirmed 2. System prompts with message that order is confirmed.
Special requirements	Android phone or tablet
Technology	None
Frequency	Many times, a day.

Table 2.10 UCD for Scheduling of Delivery

UCD-10 Scheduling of delivery	
Primary Actor:	Customer, Admin
Pre-Condition:	User is logged on to the system.
Post-Condition:	Schedule against the prescription is added.
Main success scenario:	1. User selects from options(weekly/fortnight/monthly) 2. System then saves the schedule with specified option

Alternative scenario	a. Android OS is crashed. b. No internet connection
Special requirements	Android phone or tablet
Technology	None
Frequency	Many times for each prescription

Table 2.11 UCD for Update Schedule

UCD-11 Update Schedule	
Primary Actor:	Customer, Admin
Pre-Condition:	Customer is logged on to the system.
Post-Condition:	Prescription is edited successfully
Main success scenario:	1. User selects Schedule option 2. System displays Schedules list. 3. User selects schedule. 4. System opens schedule 5. User changes delivery option(weekly, fortnight, monthly) 6. System saves the changes
Alternative scenario	a. Internet is not available. b. Android OS is crashed. 5a. User tries to update a schedule that is already confirmed 3. System prompts with message that order is confirmed.

Special requirements	Android phone or tablet
Technology	None
Frequency	Many times, a day.

Table 2.12 UCD Delete Schedule

UCD-12 Delete Schedule	
Primary Actor:	Customer, Admin
Pre-Condition:	User is logged on to the system.
Post-Condition:	Schedule against the prescription is Deleted.
Main success scenario:	1. User selects his added Schedules 2. System displays Schedules 3. User selects schedule to delete and presses delete button 4. System prompts for confirmation. 5. User selects confirm deletion.
Alternative scenario	a. Android OS is crashed. b. No internet connection 3a User tries to cancel a schedule that is already confirmed 1. System prompts with message that order is confirmed.
Special requirements	Android phone or tablet
Technology	None
Frequency	Many times for each prescription

Table 2.13 Set Alarm

UCD -13 Set Alarm	
Primary Actor:	Customer
Pre-Condition:	User is logged on to the system.
Post-Condition:	Alarm is set for intake of medicine(s).
Main success scenario:	1. User selects Alarm option. 2. System displays Alarm screen 3. User presses add alarm. 4. System displays add alarm screen 5. User selects prescription from prescription list 6. System retrieves list of medicines for that prescription. 7. User selects medicine 8. System generates alarms depending upon the number of intakes of the medicine which are to be set by user 9. User selects time, snooze and tone for the alarm 10. System saves the alarm. User repeats steps 5-9 if there are more than one medicine against which alarm is to be set and system repeats step 10 consequently.
Alternative scenario	a. Android OS is crashed.
Special requirements	Android phone or tablet

Technology	None
Frequency	Few times for each prescription

Table 2.14 UCD for Update Alarm

UCD -14 Update Alarm	
Primary Actor:	Customer
Pre-Condition:	Customer is logged on to the system.
Post-Condition:	Alarm is updated for intake of medicine(s).
Main success scenario:	1. User select Alarm option. 2. System displays Alarm screen with saved alarms. 3. User edits time, snooze or tone for the alarm 4. System saves the alarm. User repeats steps 3 if there are more than which alarm is to be edited and system repeats step 4 consequently.
Alternative scenario	a. Android OS is crashed.
Special requirements	Android phone or tablet
Technology	None
Frequency	Few times for each prescription

Table2.15 UCD for Delete Alarm

UCD -15 Delete Alarm	
Primary Actor:	Customer
Pre-Condition:	Customer is logged on to the system.
Post-Condition:	Alarm is deleted for intake of medicine(s).
Main success scenario:	1. User select Alarm option. 2. System displays Alarm screen with saved alarms. 3. User selects already added alarm and presses delete button 4. System prompts for confirmation. 5. User selects confirm deletion. 6. System deletes the alarm User repeats steps 3-5 if there are more than one alarm which is to be deleted and system repeats step 6 consequently.
Alternative scenario	a. Android OS is crashed.
Special requirements	Android phone or tablet
Technology	None
Frequency	Few times for each prescription

Table 2.16 Logging Condition after medicine intake

UCD -16 Logging Condition after medicine intake	
Primary Actor:	User

Pre-Condition:	User is logged on to the system.
Post-Condition:	Condition of Customer is logged
Main success scenario:	1. User selects log condition 2. System displays all added prescriptions 3. User selects prescription. 4. System displays a screen with latest date with a list of all medicines and their intake time(taken from alarms)each with emojis and an optional comment field. 😫 ☹ 😐 🙂 😃 5. User selects the desired emoji and adds comments to comment field 6. System saves the emoji against that medicine's specific intake time and disables the emoji for editing User repeats step 4-5 if user wants to log condition against more than one medicine / for more than one intake and system consequently repeats step 6.
Alternative scenario	a. Android OS is crashed.
Special requirements	Android phone or tablet
Technology	
Frequency	Many times a day

Table 2.17 Tracking order

UCD -17 Tracking order	
Primary Actor:	Customer
Pre-Condition:	Order(medicine) is received
Post-Condition:	Customer views tracking details.
Main success scenario:	1. User selects Schedule option 2. System displays Schedule list. 3. User then selects Schedule 4. System opens schedule 5. User presses tracking button 6. System displays the status(Order Placed /Order on its way)
Alternative scenario	a. Android OS is crashed.
Special requirements	Android phone or tablet
Technology	None
Frequency	Many times a day

Table 2.18 UCD for View Prescription

UCD-18 View Prescription	
Primary Actor:	Customer/Admin
Pre-Condition:	Customer/Admin is logged on to the system.
Post-Condition:	Desired prescription is viewed

Main success scenario:	1. User selects prescription option 2. System displays a list of prescriptions 3. User selects prescription to view 4. System opens that prescription.
Alternative scenario	a. Internet is not available. b. Android OS is crashed.
Special requirements	Android phone or tablet
Technology	None
Frequency	Many times, a day.

Table 2.19 UCD for View Schedule

UCD-19 View Schedule	
Primary Actor:	Customer/Admin
Pre-Condition:	Customer/Admin is logged on to the system.
Post-Condition:	Desired Schedule is viewed
Main success scenario:	1. User selects Schedules option 2. System displays a list of Schedules 3. User selects Schedule 4. System opens that Schedule.
Alternative scenario	a. Internet is not available. b. Android OS is crashed.
Special requirements	Android phone or tablet

Technology	None
Frequency	Many times, a day.

Table 2.20 UCD for Update Status of order

UCD-20 Update Status of order	
Primary Actor:	Admin
Pre-Condition:	Admin is logged on to the system.
Post-Condition:	Status is updated
Main success scenario:	1. User selects Schedule to confirm 2. System displays schedule with confirm Schedule option field(yes/no) 3. User selects yes 4. System marks the schedule confirmed.
Alternative scenario	a. Internet is not available. b. b. OS has crashed
Special requirements	Desktop or laptop
Technology	None
Frequency	Many times, a day.

2.6 System Sequence Diagrams

A system sequence diagram is an interaction diagram that shows how objects operate with one another and in what order.

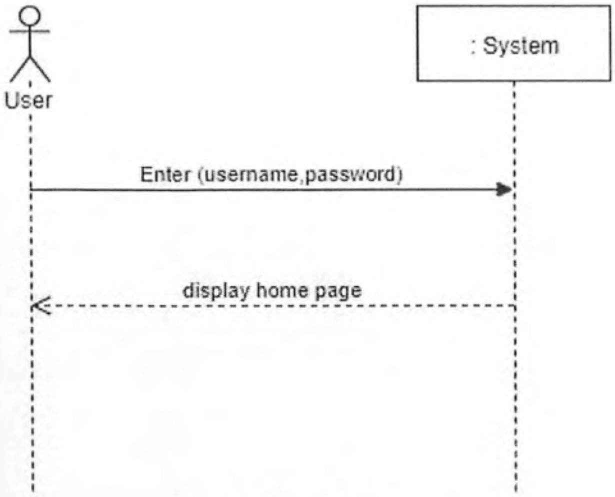


Figure2.3 Login SSD

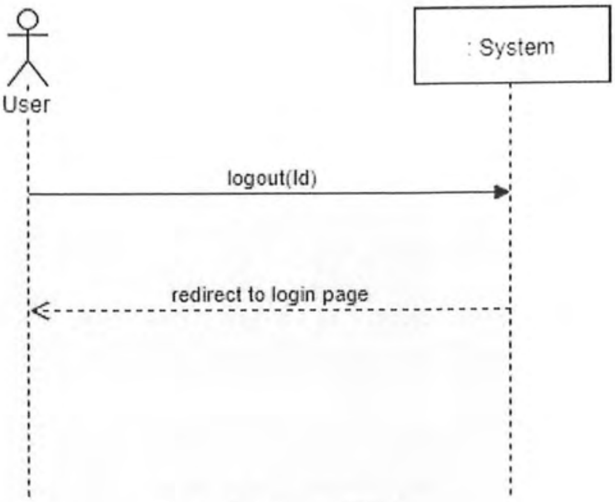


Figure 2.4 Logout SSD

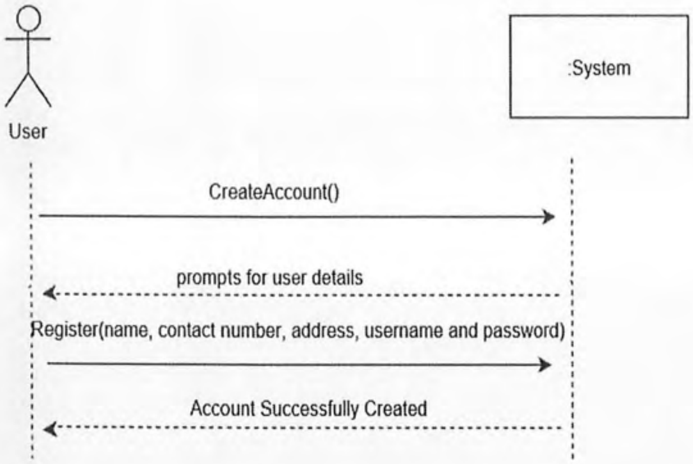

User
:System
CreateAccount()
prompts for user details
Register(name, contact number, address, username and password)
Account Successfully Created

Figure 2.5 Create Account SSD

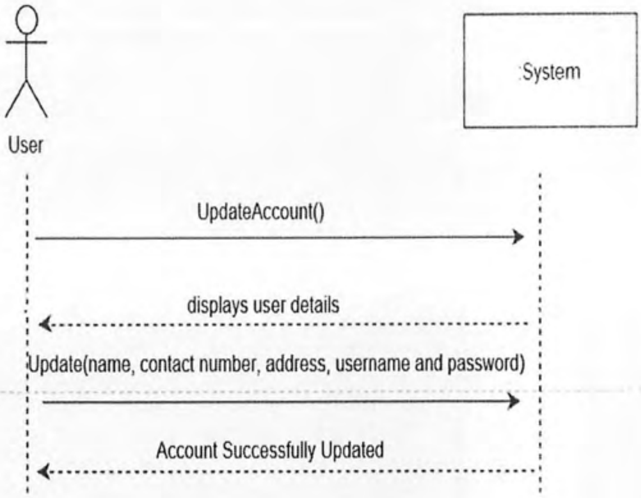

User
:System
UpdateAccount()
displays user details
Update(name, contact number, address, username and password)
Account Successfully Updated

Figure 2.6 Update Account SSD

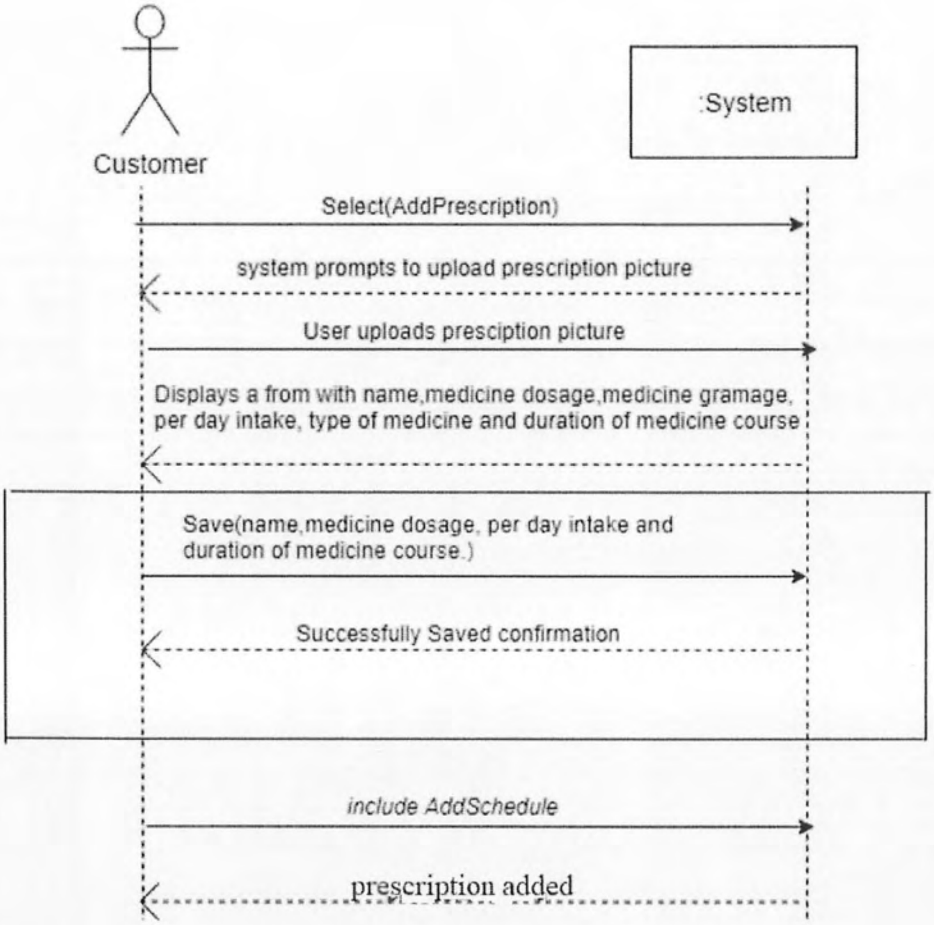

Customer
:System
Select(AddPrescription)
system prompts to upload prescription picture
User uploads presciption picture
Displays a from with name,medicine dosage,medicine gramage, per day intake, type of medicine and duration of medicine course
Save(name,medicine dosage, per day intake and duration of medicine course.)
Successfully Saved confirmation
include AddSchedule
prescription added

Figure 2.7 Add Prescription SSD

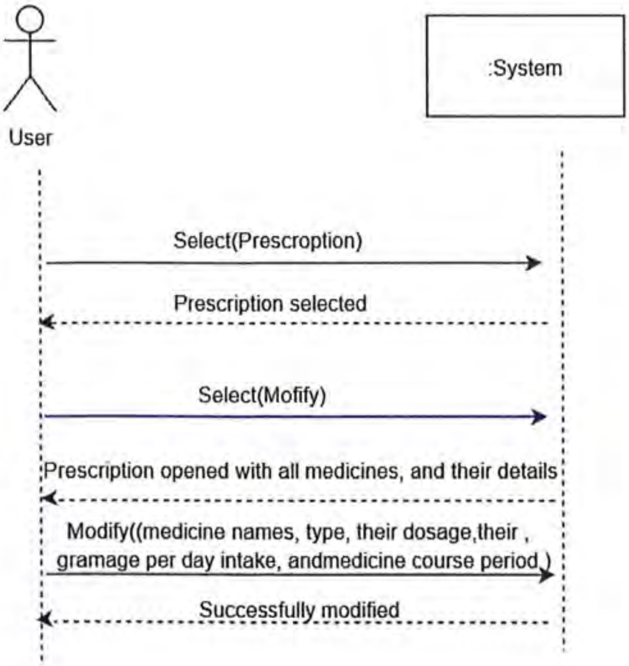

User
:System
Select(Prescroption)
Prescription selected
Select(Mofify)
Prescription opened with all medicines, and their details
Modify((medicine names, type, their dosage,their , gramage per day intake, andmedicine course period.)
Successfully modified

Figure 2.8 Modify Prescription SSD

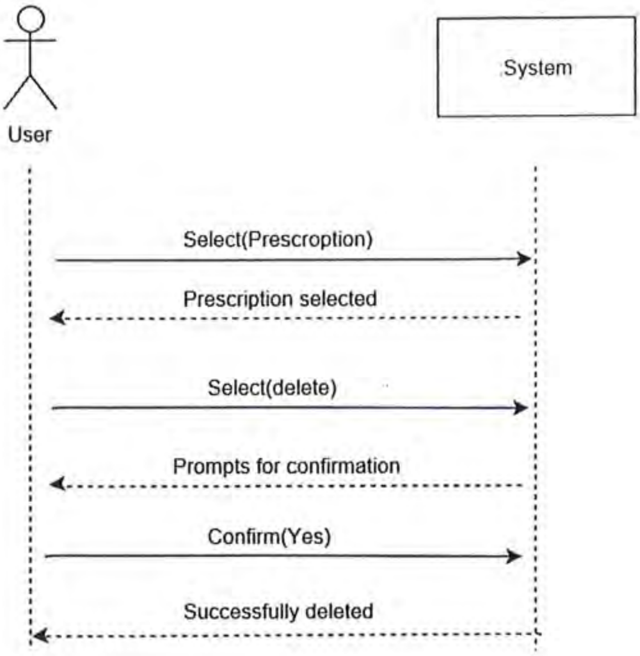


Figure 2.9 Delete Prescription SSD

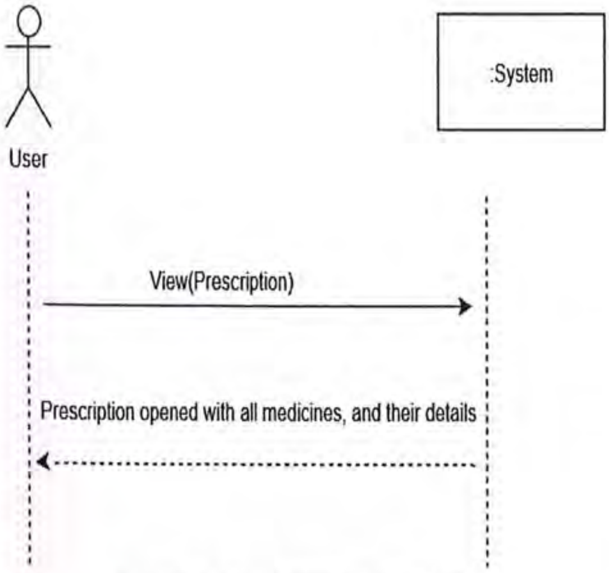


Figure 2.9 View prescription SSD

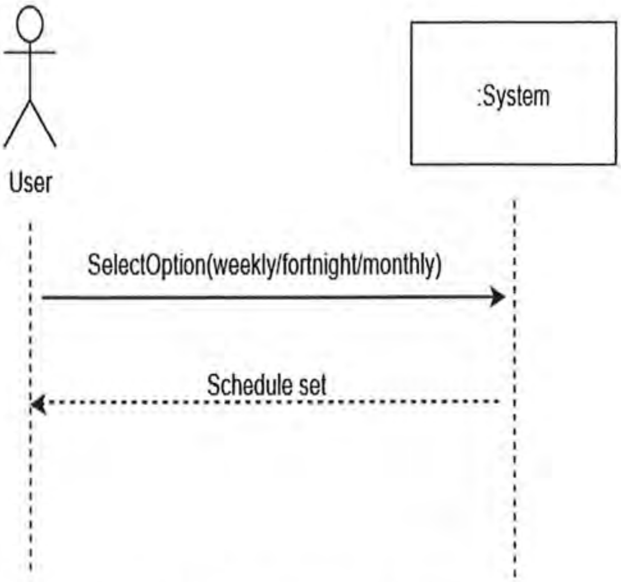


Figure 2.10 Add Schedule SSD

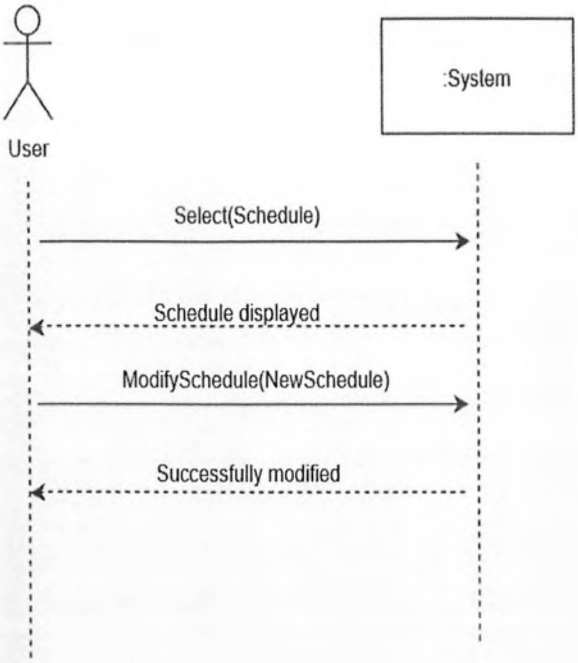


Figure 2.11 Modify Schedule SSD

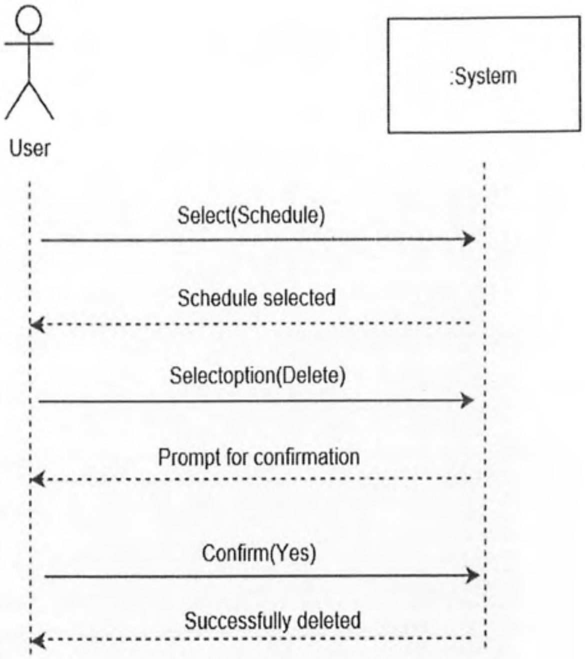


Figure 2.12 Delete Schedule

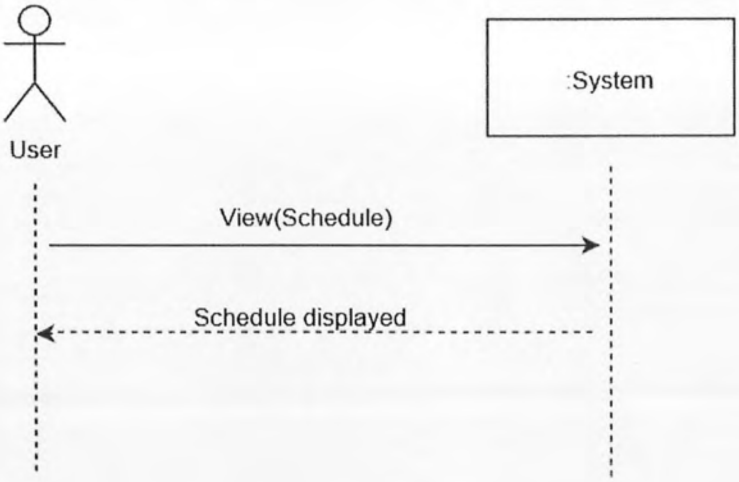


Figure 2.13 View Schedule

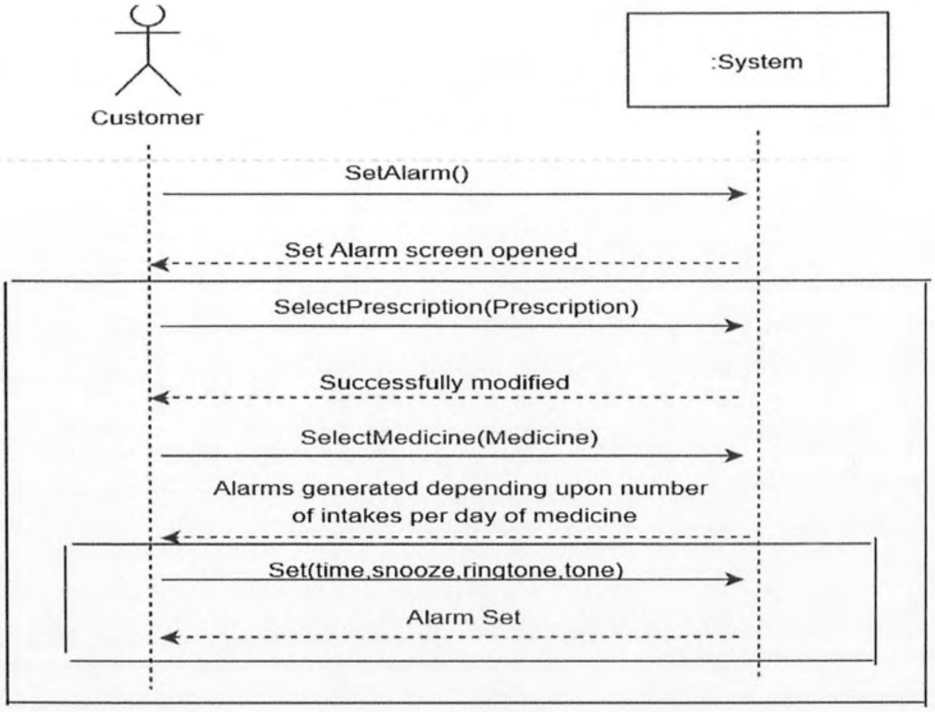

Customer
:System
SetAlarm()
Set Alarm screen opened
SelectPrescription(Prescription)
Successfully modified
SelectMedicine(Medicine)
Alarms generated depending upon number of intakes per day of medicine
Set(time,snooze,ringtone,tone)
Alarm Set

Figure 2.14 Set Alarm

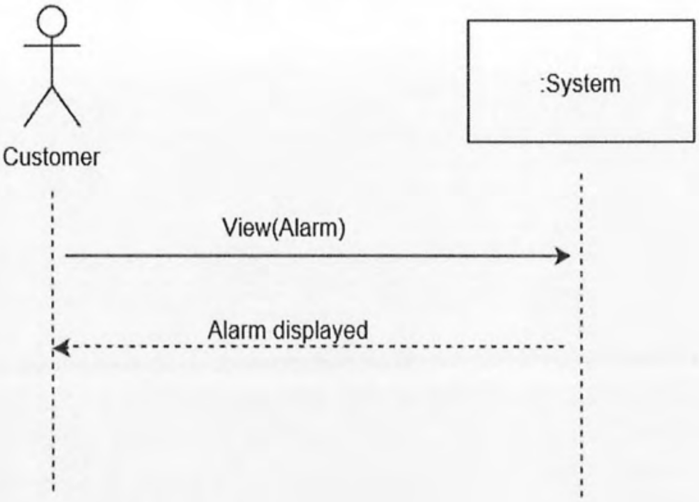


Figure 2.15 View Alarm

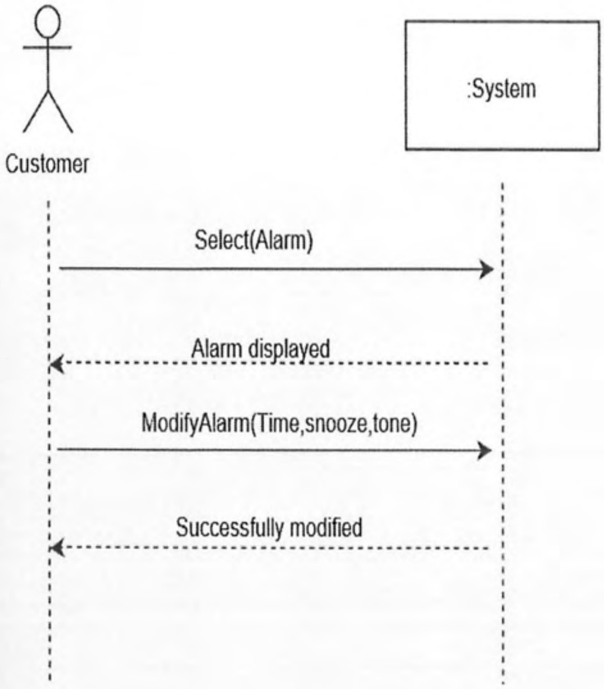


Figure 2.16 Modify Alarm SSD

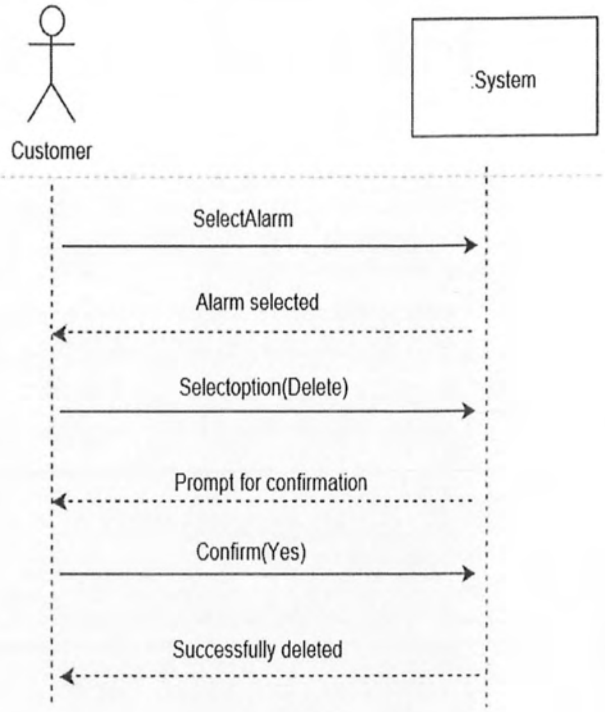


Figure 2.17 Delete Alarm SSD

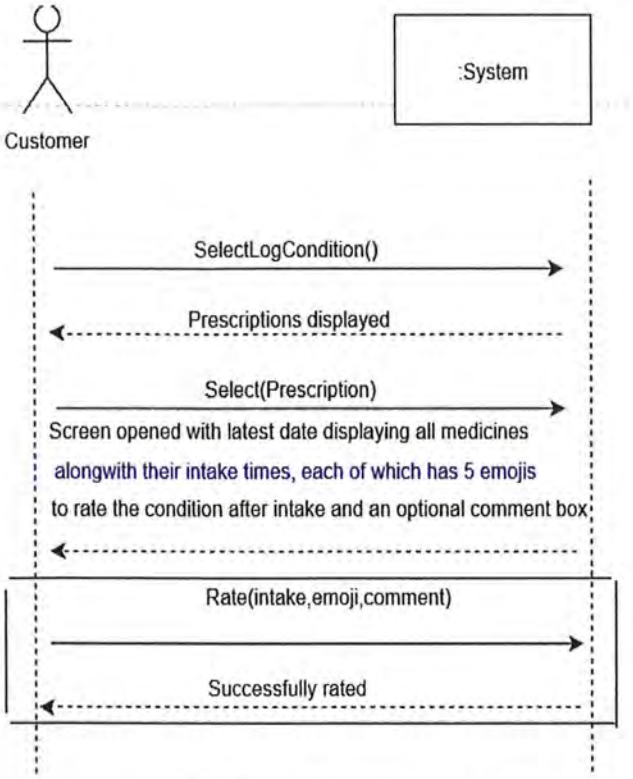

Customer
:System
SelectLogCondition()
Prescriptions displayed
Select(Prescription)
Screen opened with latest date displaying all medicines
alongwith their intake times, each of which has 5 emojis
to rate the condition after intake and an optional comment box
Rate(intake,emoji,comment)
Successfully rated

Figure 2.18 Log Condition SSD

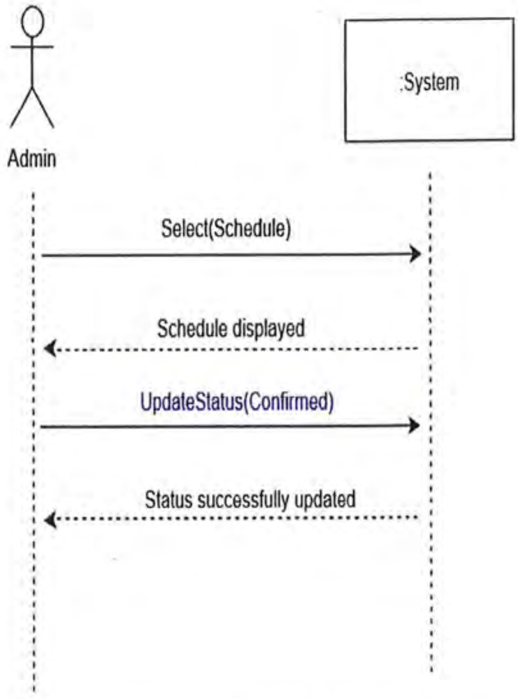


Figure 2.19 Update status of order SSD

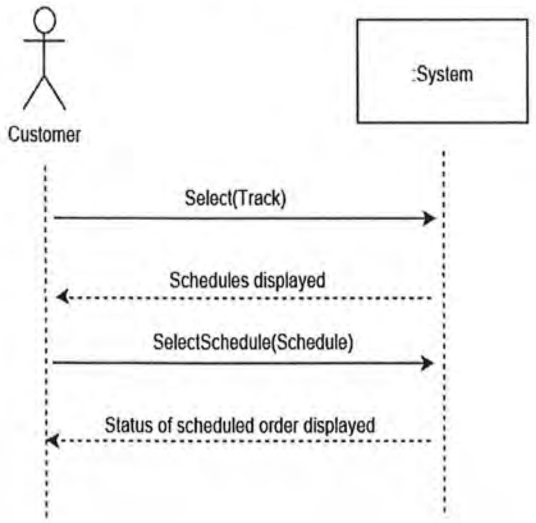


Figure 2.20 Track Order SSD

2.7 Activity Diagrams

Activity diagram is UML behavior diagram which shows flow of control or object flow with emphasis on the sequence and conditions of the flow.

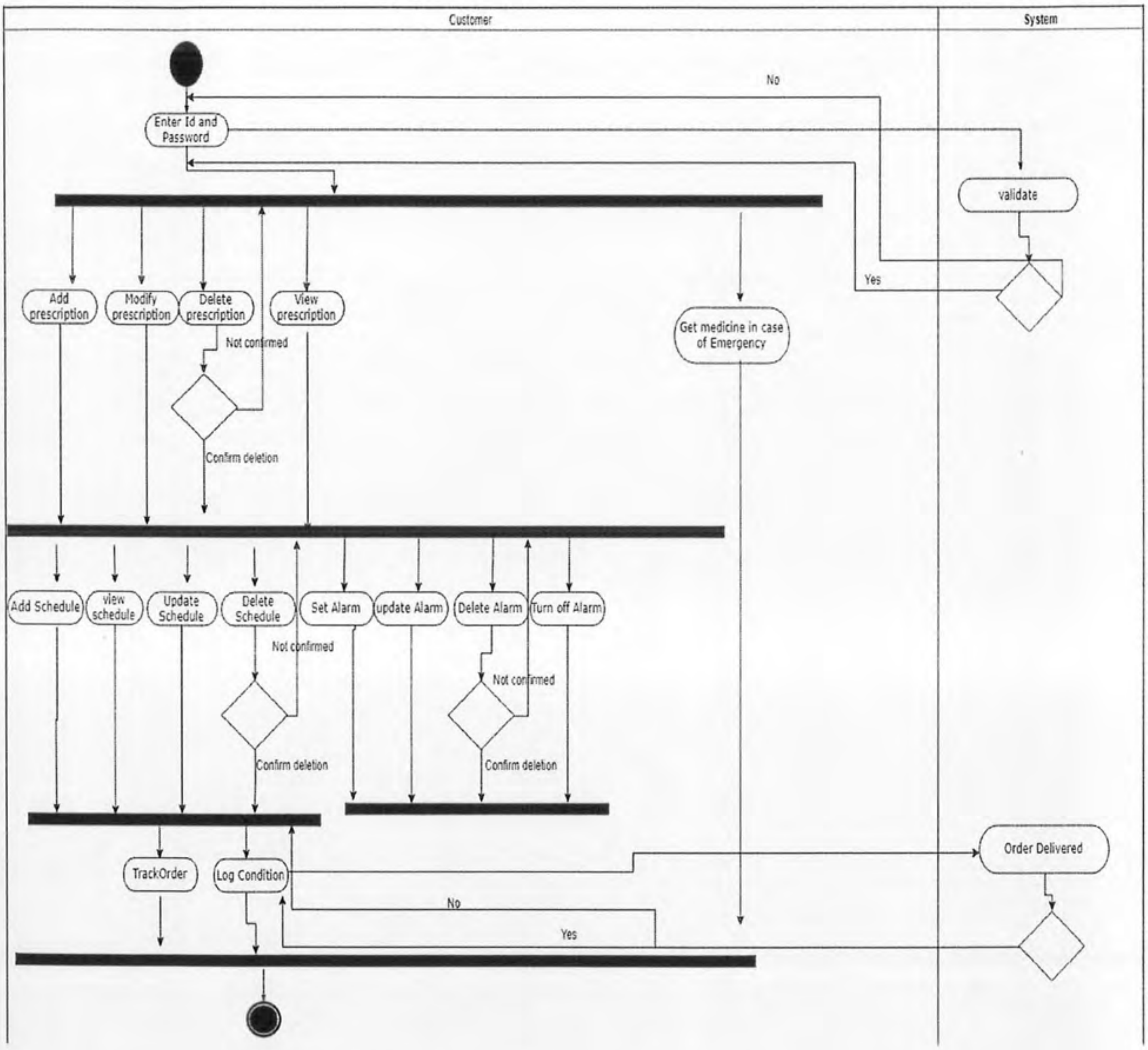


Figure 2. 24 Activity Diagram for Customer

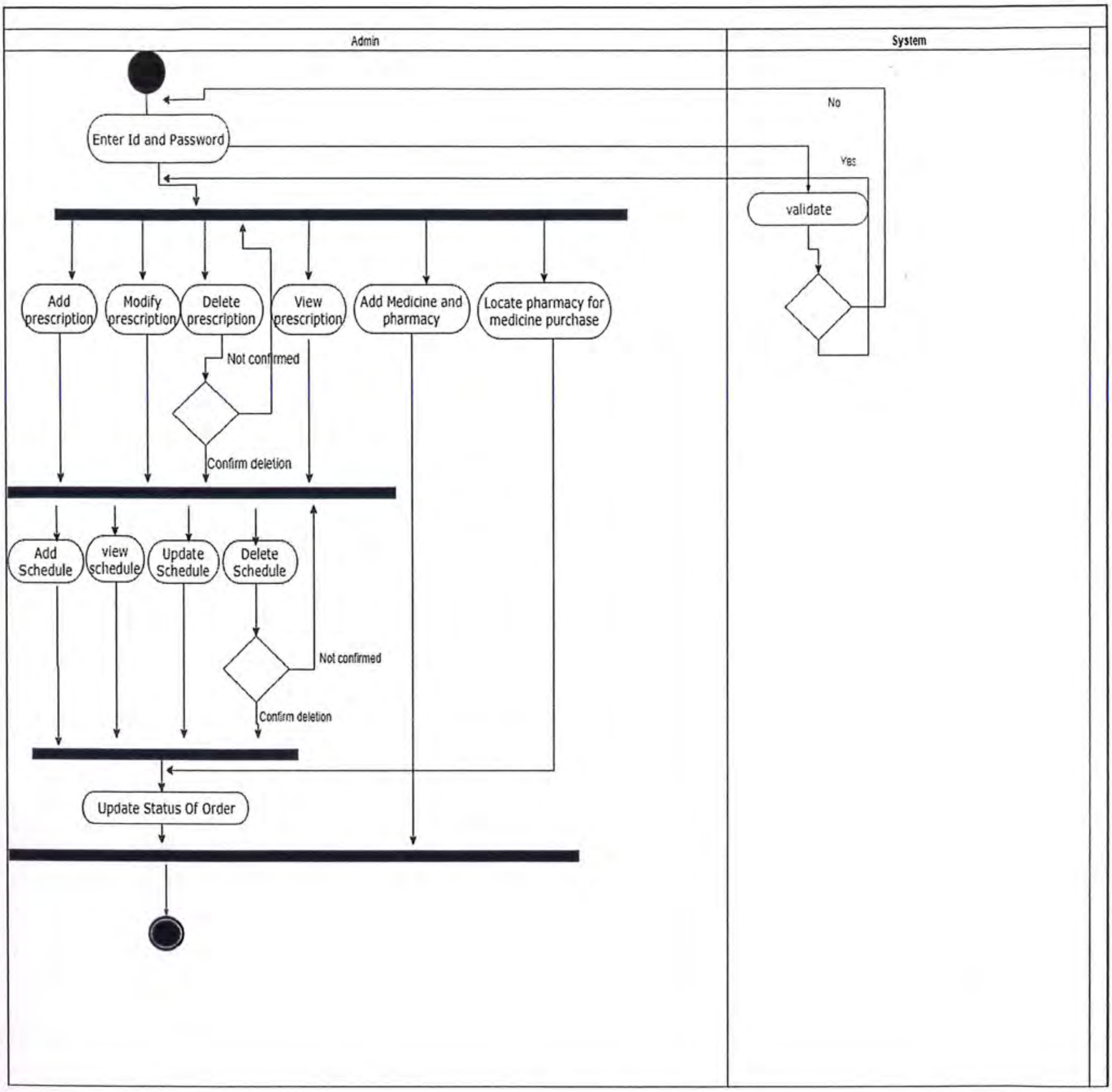


Figure 2.25 Activity Diagram for Admin

2.6 Domain Model

Domain Modeling is a way to describe and model real world entities and the relationships between them, which collectively describe the problem domain space.

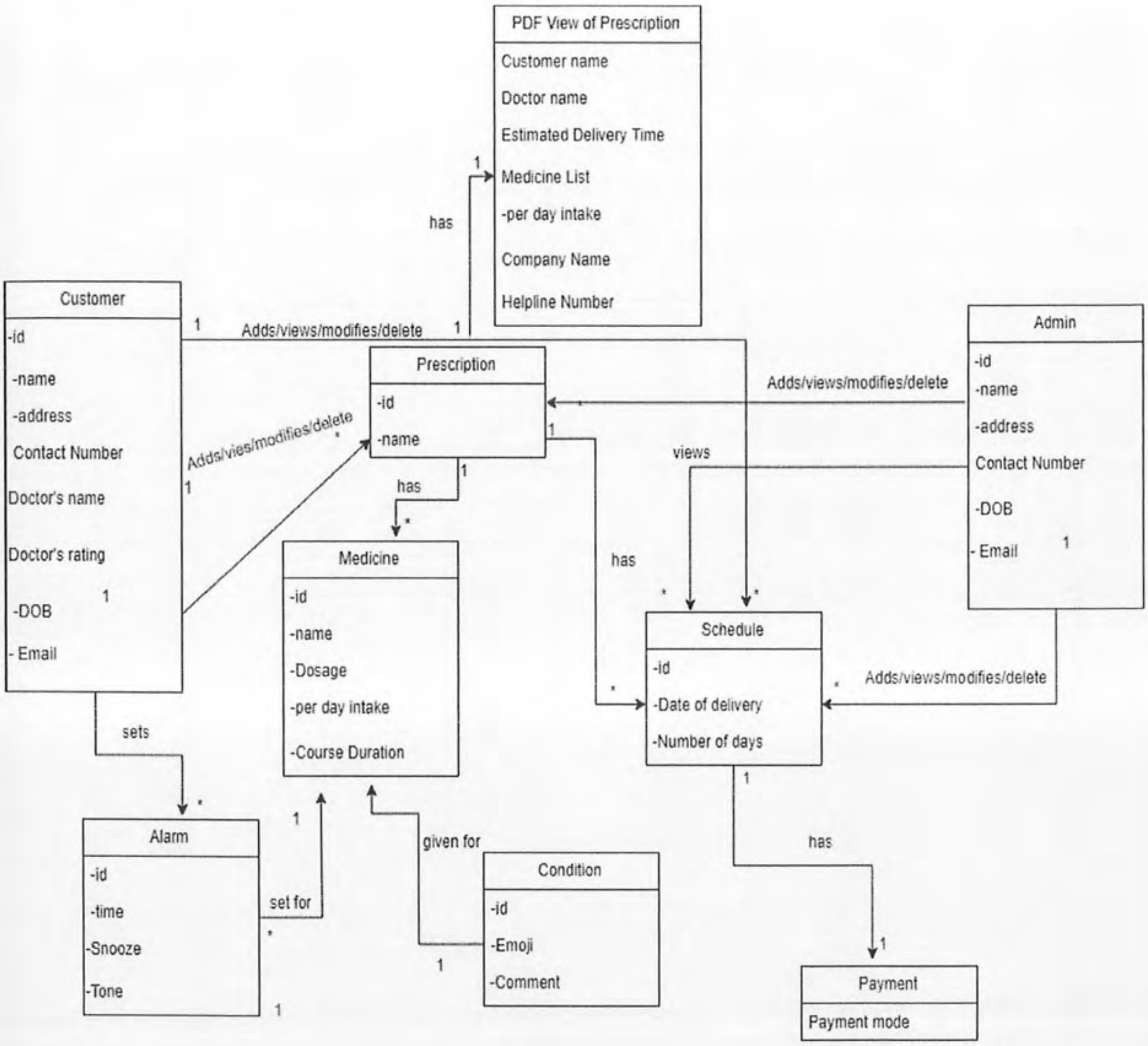


Figure 2.26 Domain Model

2.7 Software System Attributes

Software system attributes define overall factors that affect run-time behavior, application design, and user experience. To develop high quality application, software system attributes are the benchmarks that describe system's intended behavior within the environment for which it was built. Here is detail of some software system attributes.

2.7.1 Reliability

System must be able to get and store correct data

2.7.2 Availability

The system will be available all the time but it requires internet connection

2.7.3 Security

The security section describes the need to control access to the data. This includes controlling who may view and alter application data.

- Only registered users can use this application.
- A user who uses this application should have a login id and password.

2.7.4 Maintainability

The program will use modular approach (Separation of concerns) so that it would be easier to update or change code when needed.

2.7.5 Portability

The app can run on android platform only.

2.8 Database Requirements

Firebase is a platform that provides mobile and web application developers the facility and ease to create real time databases.

The Firebase Cloud Firestore is a cloud-hosted NoSQL database that lets you store and sync between your users in realtime.

NO SQL:

NoSQL is particularly useful for storing unstructured data, which is growing far more rapidly than structured data and does not fit the relational schemas of RDBMS

There are four categories of NoSQL database:

1. Key-value data stores
2. Document stores
3. Wide-column stores
4. Graph stores

Firebase Real-time Database is of Key-value datatype and is really just one big JSON object that the developers can manage in real-time.

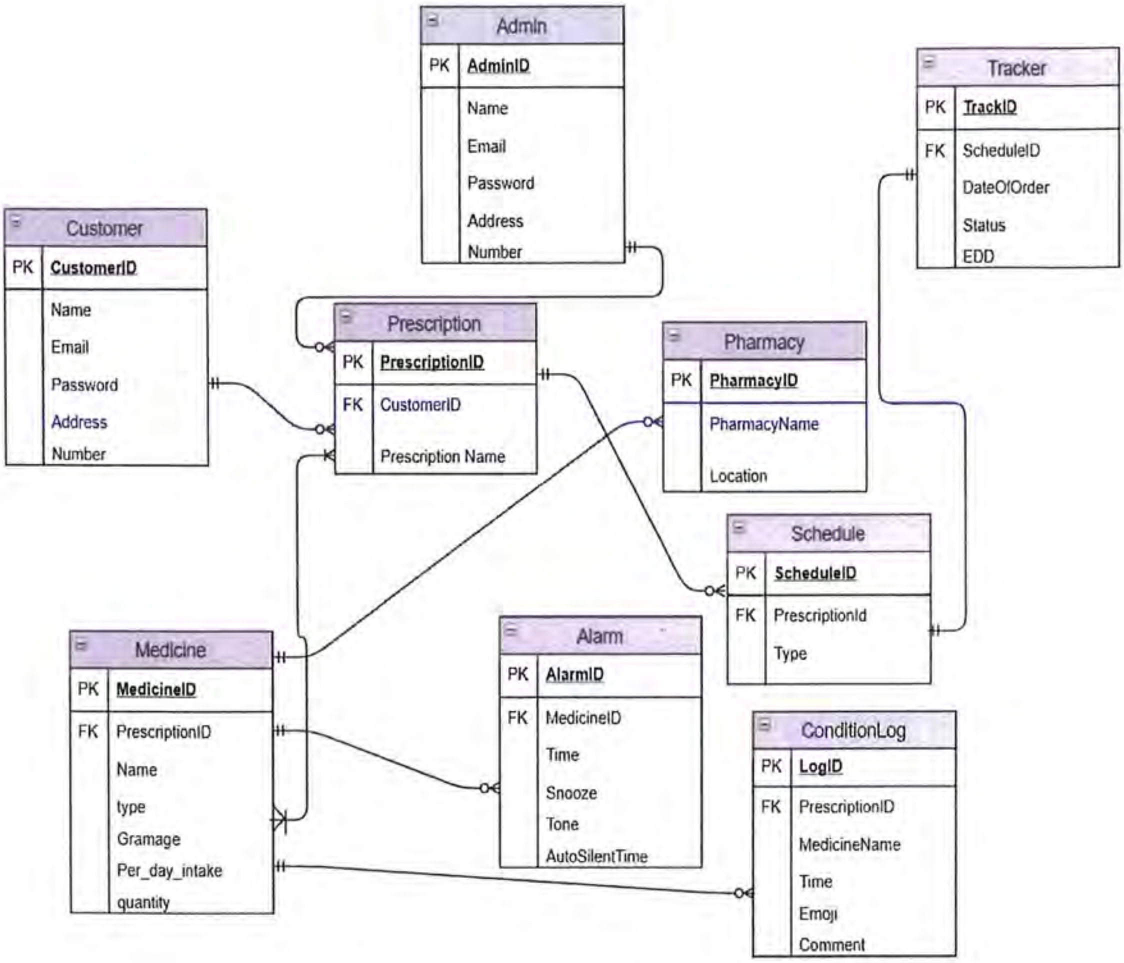


Figure 2.27 Entity Relation Diagram

Summary:

This chapter provides the specification of what the system will have, all functionalities, any external interfaces, brief description of technologies to be used and a plan depicting how the time will be managed to deliver the final product.

Chapter 3

Software Design Description

3.1 Introduction

This chapter aims at the design of software to be developed such as architecture, sequence diagrams, class diagrams and design of user interfaces.

3.1.1 Overview

The Software Design Document is a document which provides documentation to aid in software development by providing the details for how the software should be built. Within the Software Design Document are narrative and graphical documentation of the software design for the project including architecture diagram, sequence diagrams, class diagrams and other object behavior models and supporting requirement information. It includes the description of HOW the software will meet the requirements.

3.1.2 Purpose

The purpose of the Software Design Document is to provide a description of the design of MedDeliver. It allows for software development to proceed with an understanding of what is to be built and how it is expected to be built.

3.1.3 Requirements Traceability Matrix Requirement

Traceability Matrix or RTM captures all requirements proposed by the client or development team and their traceability in a single document delivered at the conclusion of the life-cycle. In other words, it is a document that maps and traces user requirement with test cases. The main purpose of Requirement Traceability Matrix is to see that all test cases are covered so that no functionality should miss while testing

3.2 System Architectural Design

Architectural design is concerned with understanding how a system should be organized and designing the overall structure of that system.

It identifies the main structural components in a system and the relationships between them. The product of architectural design is an architectural model which highlights the components of system and the interaction between them. [1]

3.2.1 Selected System Architecture

The selected architecture for the underlying system is a 3-tier architecture.A three-tier architecture is a client-server architecture in which the functional process logic, data access, computer data storage and user interface are developed and maintained as independent modules on separate platforms. Three-tier architecture is a software design pattern and a well-established software architecture. Three-Tier Architecture provides the following benefits.

- Allows for separation of concerns i.e. every layer is independent and is responsible for only itself.
- A separate back-end tier, for example, allows you to deploy to a variety of databases instead of being locked into one particular technology. It also allows you to scale up by adding multiple web servers.
- It provides an ease of maintenance of the code base, managing presentation code and business logic separately, so that a change to business logic. [2]

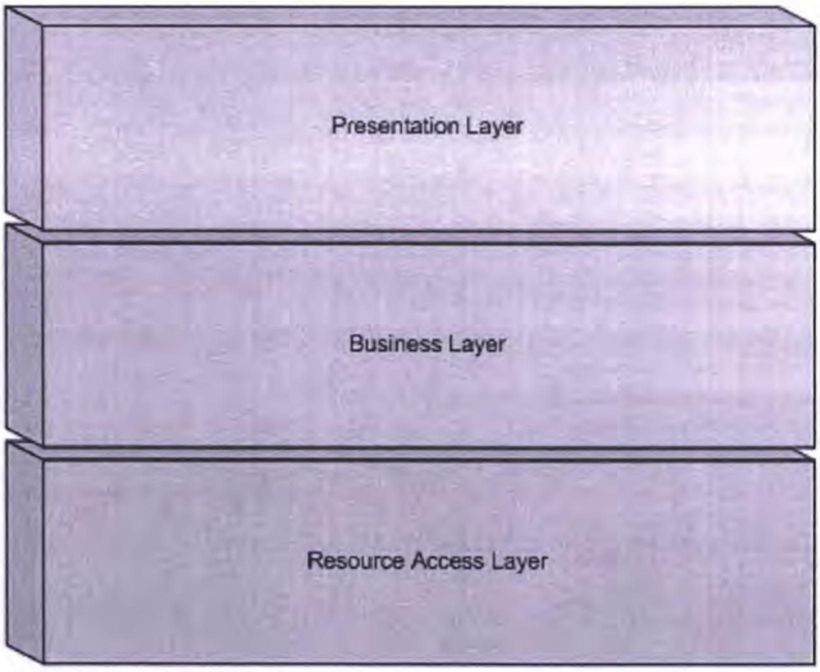


Figure 3.1 3-tier Architecture

3.2.2 Discussion of Alternative Designs

An alternative design was to use two-tier architecture, based on Client Server. It allows for direct communication between client and server. Tight coupling of a 2-tiered architecture makes the system run faster. But tight coupling makes it difficult to make changes in one tier without affecting the other. In addition to this, server cannot respond to multiple requests at same time, as a result it causes a data integrity issue. Communication is easy but performance is affected by increasing the users.

Therefore, I discarded this option and went on with 3-tier architecture.

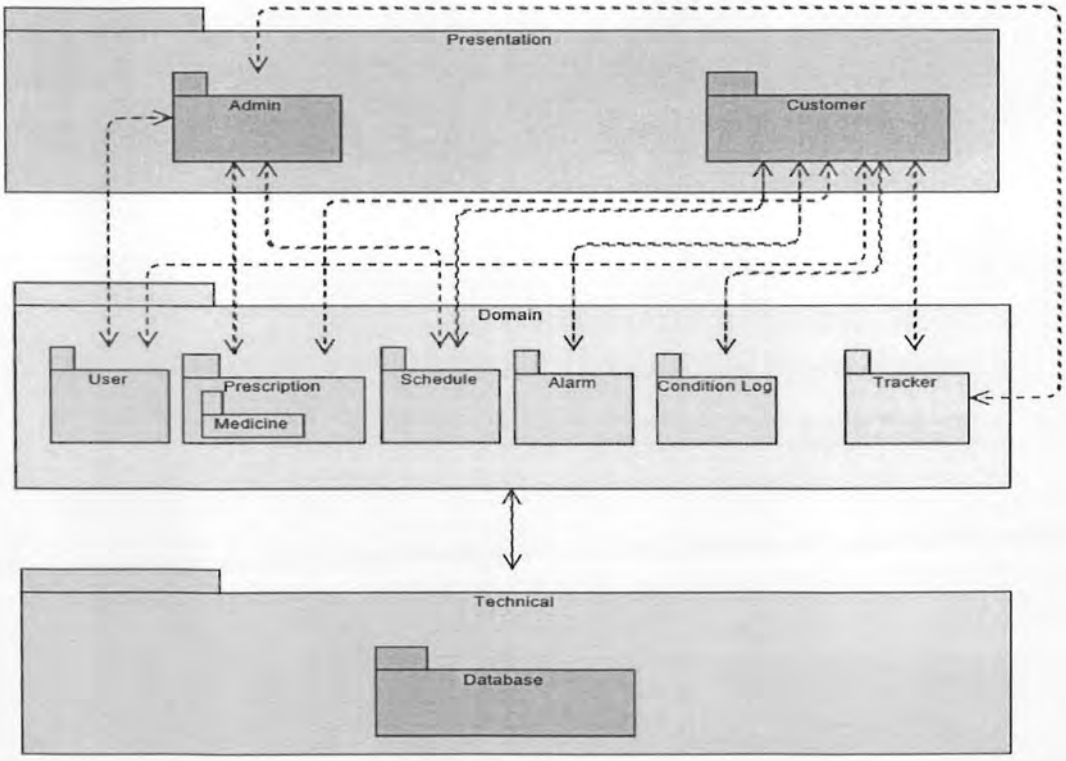


Figure 3.3 Package Diagram

3.2.3 System Interface Description

System interface describes the flow of resources. It is the logical characteristics of each interface between the software product and the hardware components of the system. Figure 3.4 shows the software interface of the software. It clearly shows how different entities of system are interacting with each other.

3.3 Detailed Description of Components

In the world of UML 2, components are less physical and more conceptual stand-alone design elements such as a business process that provides or requires interfaces to interact with other constructs in the system. The physical elements described in UML 1, like files and documents, are now referred to as artifacts. A UML 2 component may contain multiple physical artifacts if they naturally belong together.

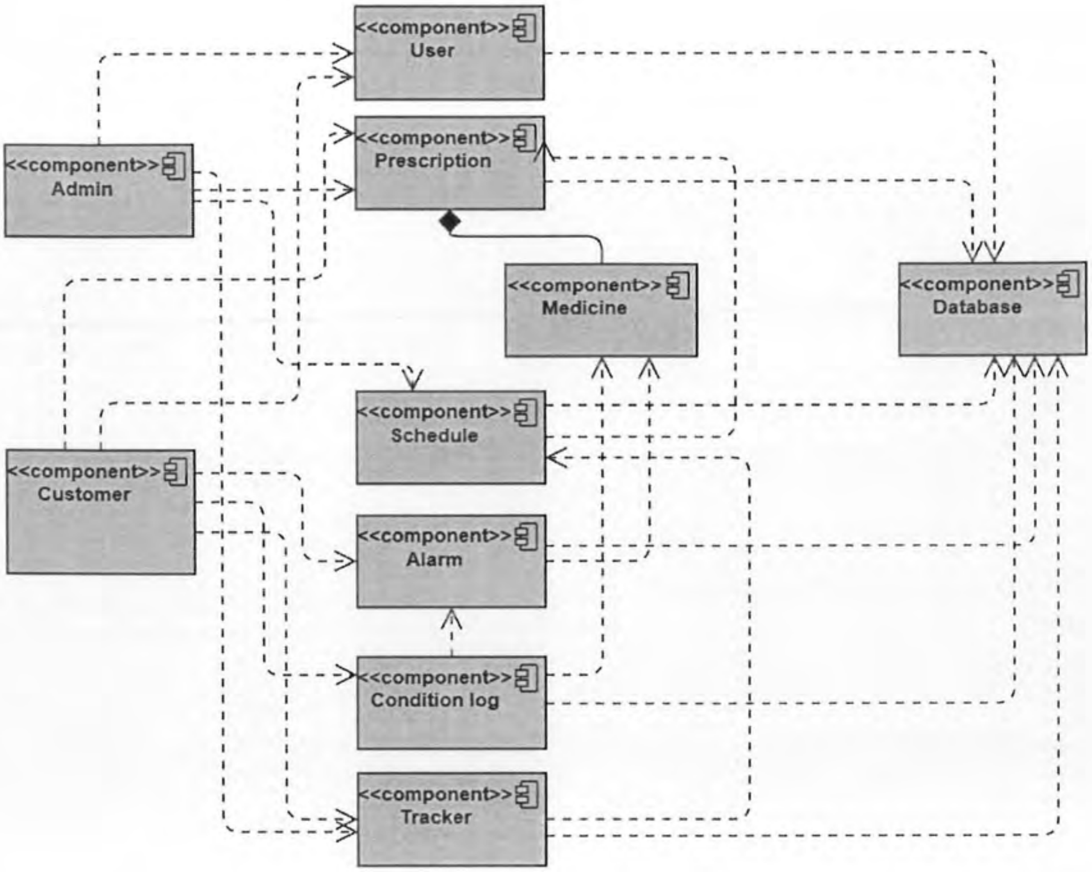


Figure 3.4 Component Diagram

Admin:

This component interacts with Users, Prescription, Schedule, Tracker. Admin can receive prescriptions and schedules added by a customer in order to place the orders in motion and then updating the status of those orders for customers to view.

Customer:

This component interacts with Users, Prescription, Schedule, Alarm, Condition log, Tracker. Admin can receive prescriptions and schedules added by a customer in order to place the orders in motion and then updating the status of those orders for customers to view.

Prescription:

This component contains medicine component, customer uses this component to add the medicines he wishes to order.

Medicine:

This component is a part of Prescription component used to store medicine details.

Schedule:

This component interacts with Prescription to set a schedule for its delivery.

Alarm:

This component interacts with Medicine components of Prescription allowing customers to set alarms for medicine intake time.

Condition log:

This component interacts with Medicine components of Prescription and Alarm component allowing customers to log condition after every medicine intake.

Tracker:

This component interacts with Schedule and is used to update status of order (by Admin) and view status of order (by Customer).

3.4 Sequence Diagram

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence.

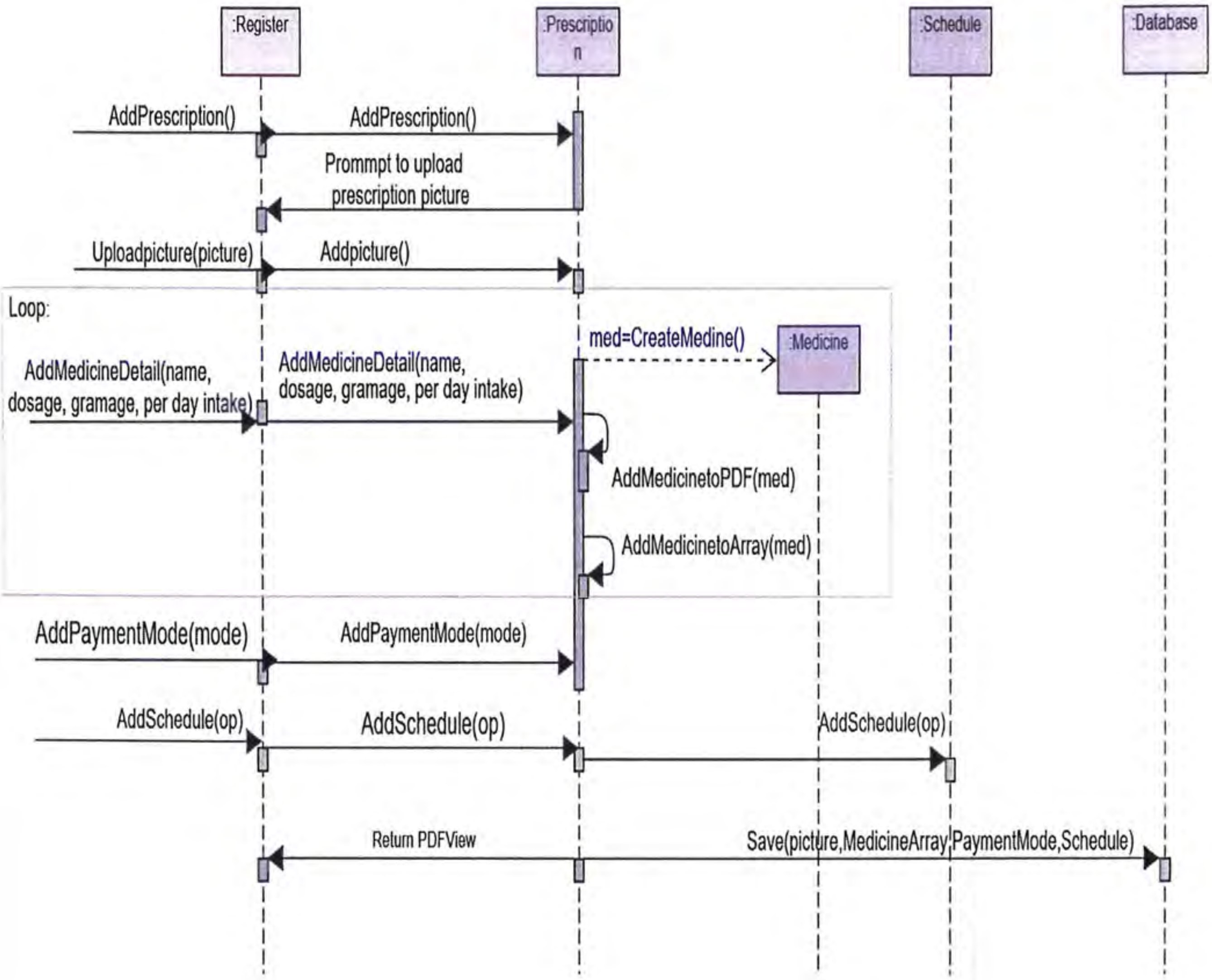


Figure 3.5 Add Prescription

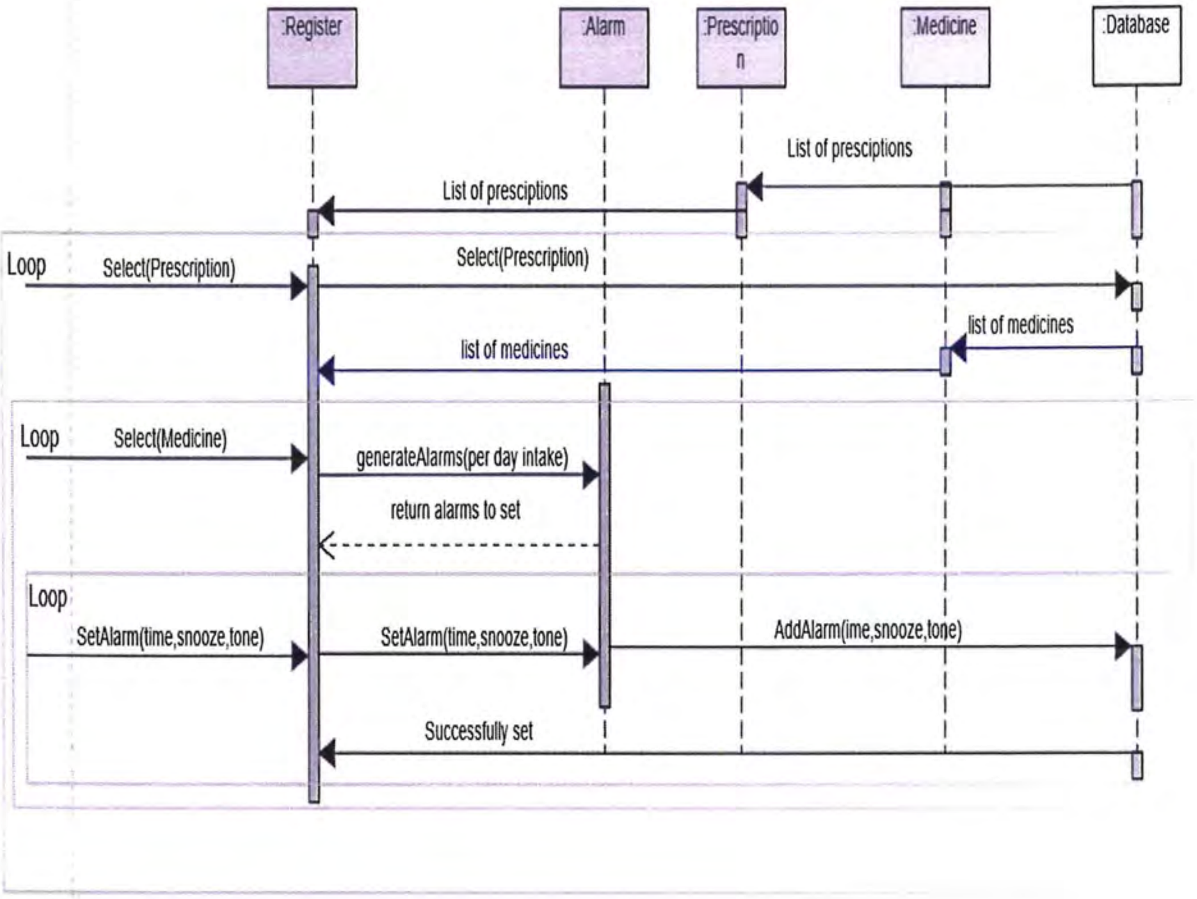


Figure 3.6 Set Alarm Sequence Diagram

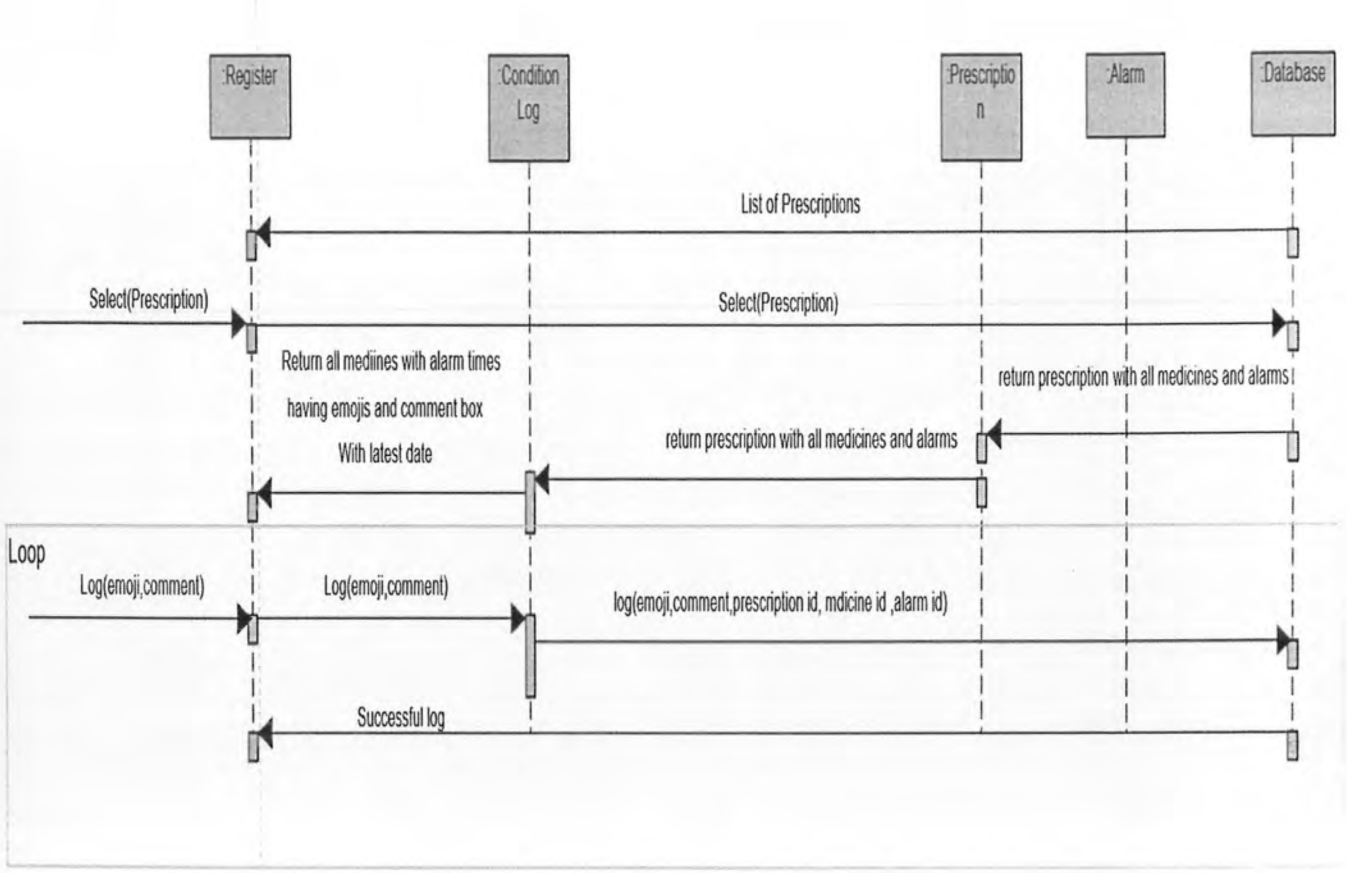

:Register
:Condition Log
:Prescriptio n
:Alarm
:Database
List of Prescriptions
Select(Prescription)
Select(Prescription)
Return all mediines with alarm times having emojis and comment box With latest date
return prescription with all medicines and alarms
return prescription with all medicines and alarms
Loop
Log(emoji,comment)
Log(emoji,comment)
log(emoji,comment,prescription id, mdicine id ,alarm id)
Successful log

Figure 3.7 Log Condition Sequence Diagram

3.5 Class Diagram

A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity

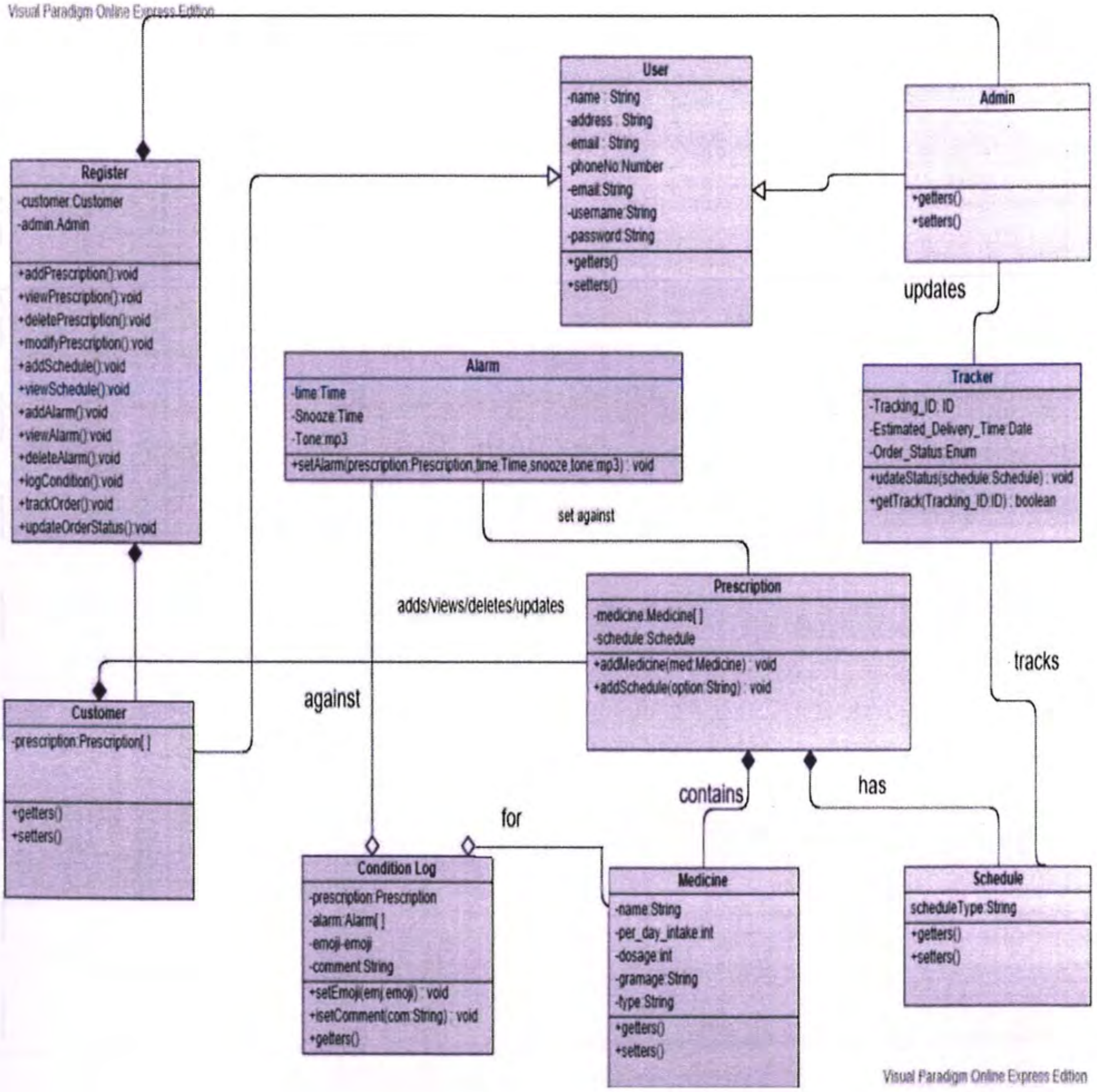


Figure 3.8 Class Diagram

3.6 User Interface Design

3.6.1 Description of the User Interface

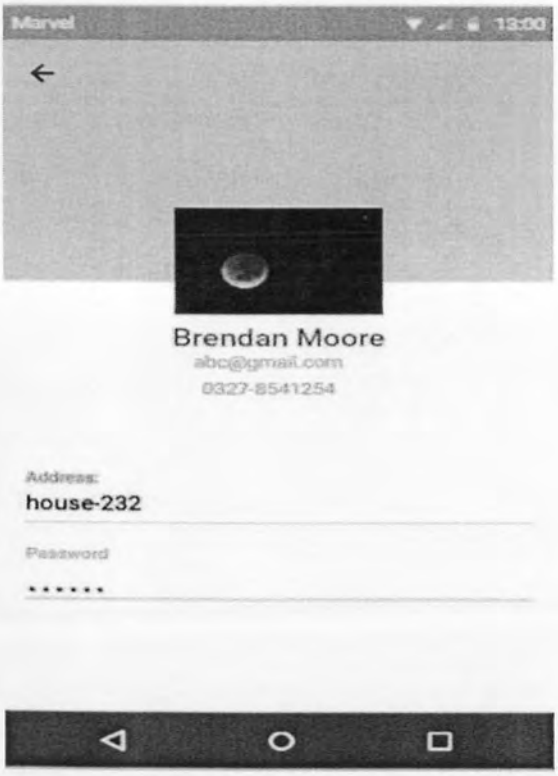
MedDeliver will be easy to use and understandable by user.

The interfaces will be designed in such a way that they will be self-explanatory leaving user with no confusion

3.6.2 Screen Images

Following are few screen images of game.

Customer Android Screens:



Brendan Moore

abc@gmail.com

0327-8541254

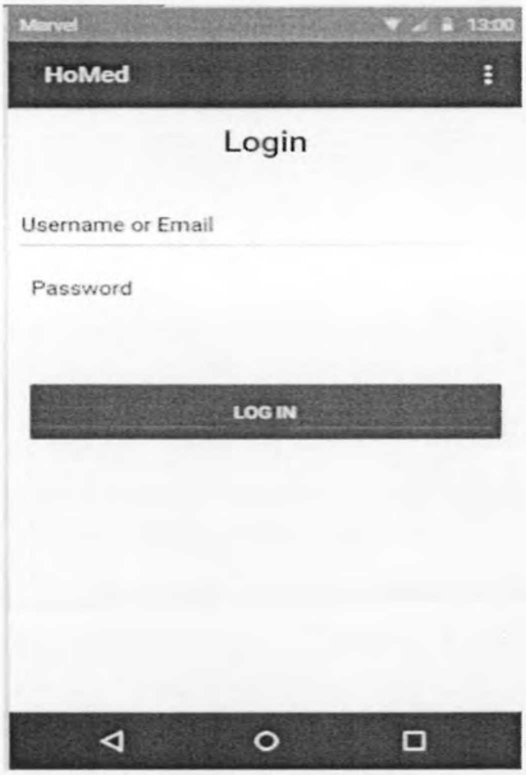
Address:

house-232

Password

••••••

Figure 3.9 profile Screen



Login

Username or Email

Password

LOG IN

Figure 3.10 Login Screen

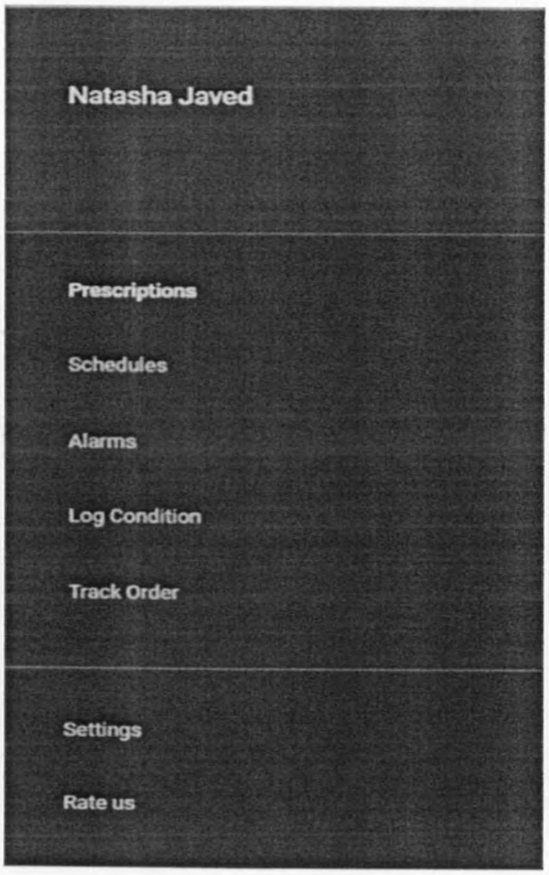


Figure 3.11 Menu Screen

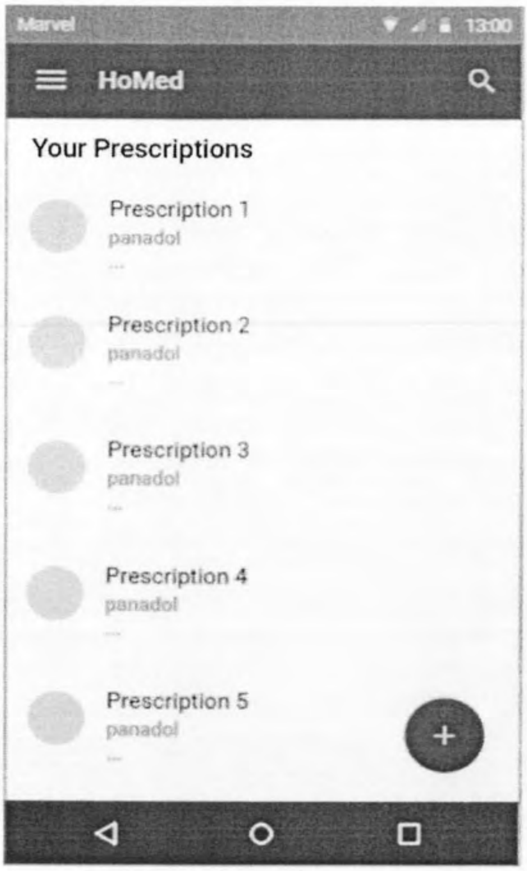
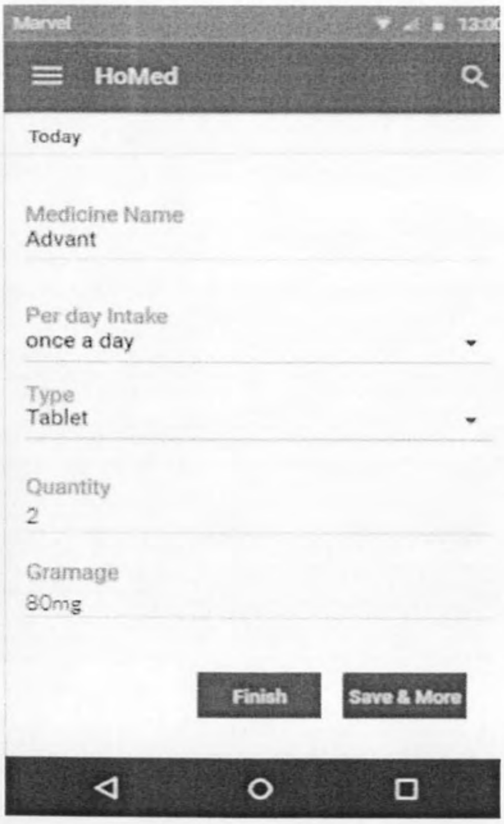


Figure 3.12 Prescription Screen 1



Today

Medicine Name
Advant

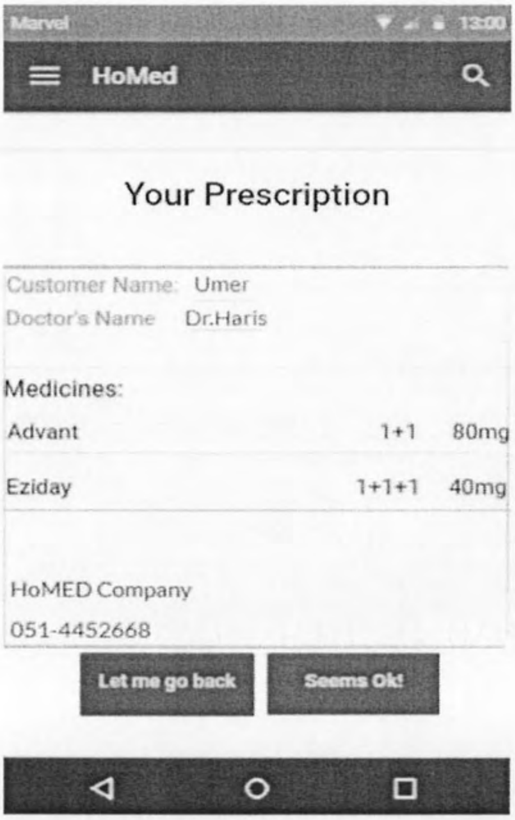
Per day Intake
once a day

Type
Tablet

Quantity
2

Gramage
80mg

Figure 3.13 Prescription Screen 2



Your Prescription

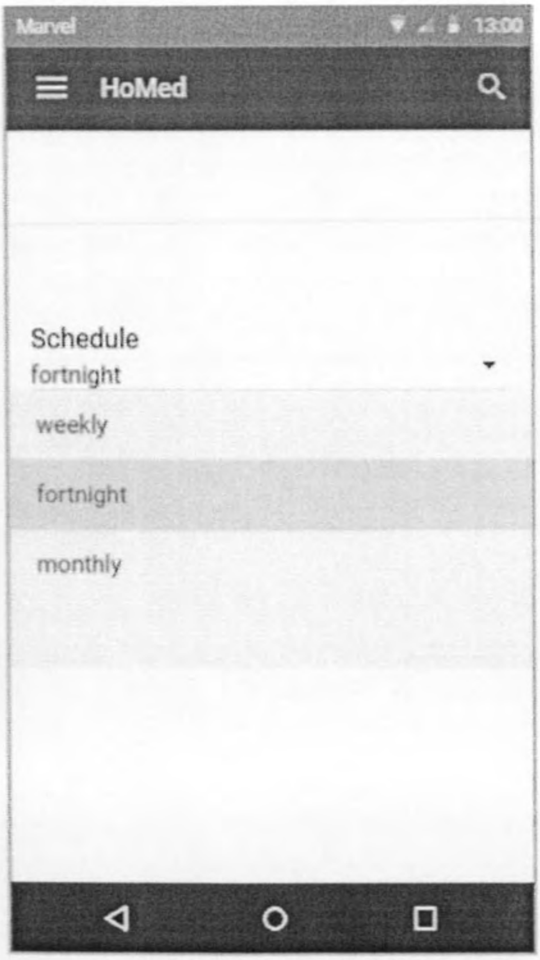
Customer Name: Umer
Doctor's Name Dr.Haris

Medicines:

Advant	1+1	80mg
Eziday	1+1+1	40mg

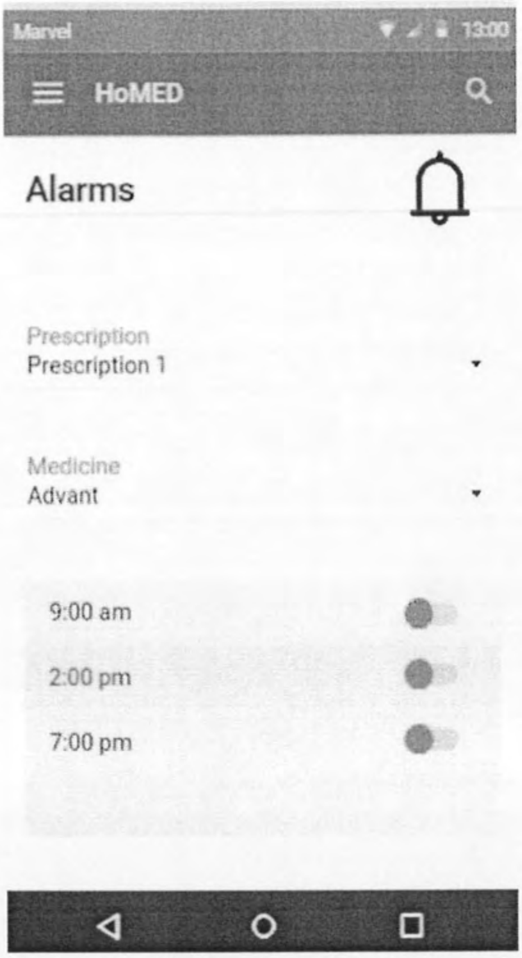
HoMED Company
051-4452668

Figure 3.14 Prescription Screen 3



Schedule
fortnight

Figure 3.15 Schedule Screen

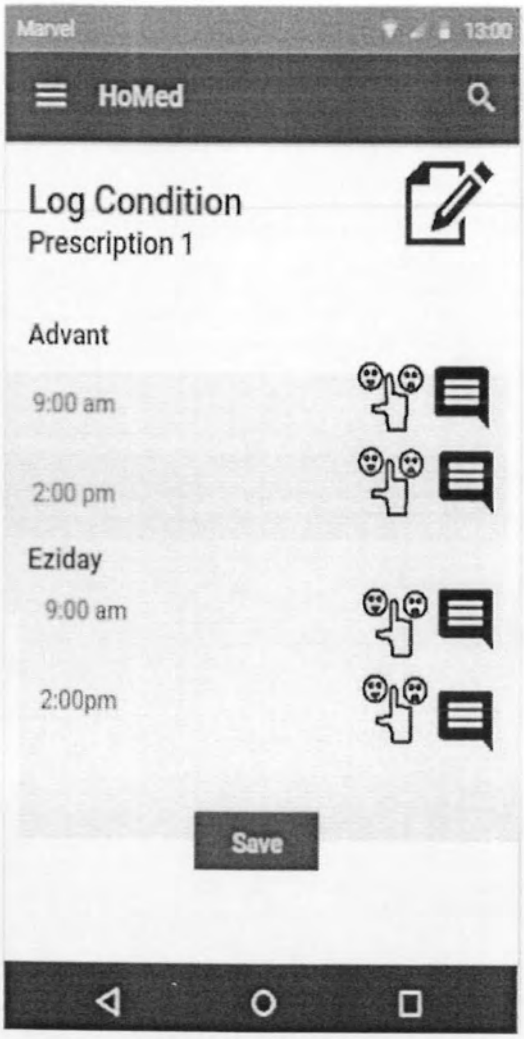


Prescription
Prescription 1

Medicine
Advant

9:00 am
2:00 pm
7:00 pm

Figure 3.16 Alarm Screen

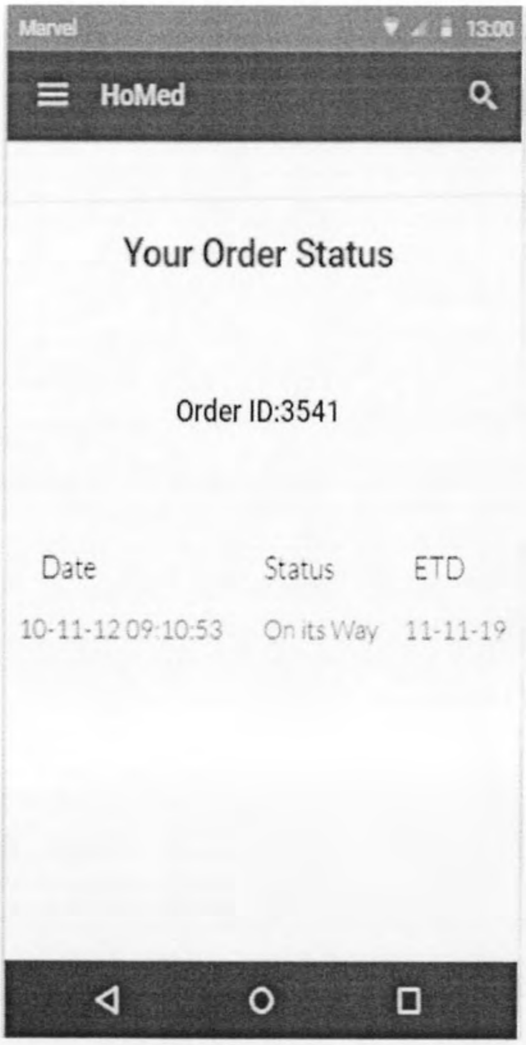


Log Condition
Prescription 1

Advant
9:00 am
2:00 pm

Eziday
9:00 am
2:00pm

Figure 3.17 Log Condition



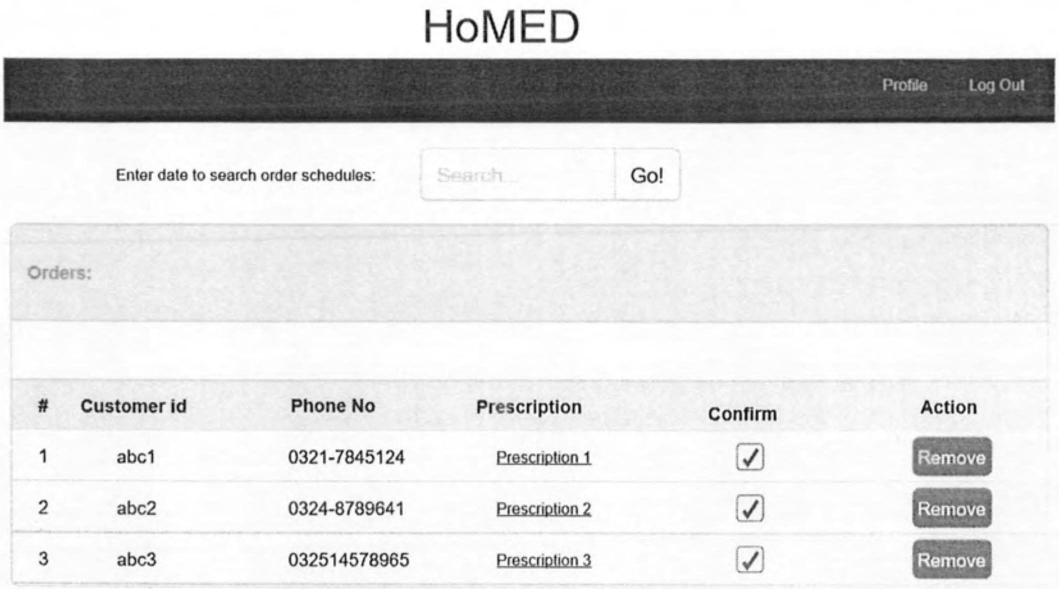
Your Order Status

Order ID:3541

Date	Status	ETD
10-11-12 09:10:53	On its Way	11-11-19

Figure 3.18 Track Order Screen

Admin Android Tablet Screen



Enter date to search order schedules:

Orders:

#	Customer id	Phone No	Prescription	Confirm	Action
1	abc1	0321-7845124	Prescription 1	☑	Remove
2	abc2	0324-8789641	Prescription 2	☑	Remove
3	abc3	032514578965	Prescription 3	☑	Remove

Figure 3.19 Admin Screen

Chapter 4

Software Test Documentation

4.1 Introduction

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. In simple words, testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements. According to ANSI/IEEE 1059 standard, Testing can be defined as - A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item.

4.1.1 System Overview

MedDeliver aims to make people's lives easier by Saving them time and money as they won't have to go to pharmacy Ensuring that medicines reach people on time and regularly Allowing people to set alarms to take medicine on time Allowing people to maintain a log of their condition after every medicine intake that the patient can show to doctor

4.1.2 Test Approach

A test approach is the test strategy implementation of a project, defines how testing would be carried out.

Testing technique used for the MedDeliver is black box testing for the time being. The main purpose of this testing is to evaluate whether system meets the business requirements or not.

4.2 Test Plans

4.2.1 Features to be tested

1. Login
2. Logout
3. Register account
4. Update account
5. Delete account

6. Add prescription
7. Update prescription
8. Delete prescription
9. View prescription
10. Schedule delivery
11. Update schedule
12. Delete schedule
13. View Schedule
14. Set alarm for medicine intake
15. Update alarm for medicine intake
16. Log condition after medicine intake
17. Track order
18. Update Status of order

4.2.2 Features not to be tested

Features not to be tested are from the developer's point of view. For example

- power used by processor
- memory consumed by the App
- Software risk factor
- Maintainability

4.2.3 Testing Tools and Environment

Following tools and environment are used for testing:

- PC/laptop
- Android phone/tablet
- Windows operating system

4.3 Test Cases

A test case describes an input, action, or an event and an expected response, to determine if a feature of a software application is working correctly.

Table 4.1 Test case for Login

ID	TC-1
DESCRIPTION	Check that the registered user can successfully login account.
SETUP	1. Create an account for user with username nat and password nj123!.
INSTRUCTIONS	1. Enter username nat. 2. Enter password nj123!. 3. Click the submit button. 4. Enter username nat. 5. Enter password 2. 6. Click the submit button.
EXPECTED RESULT	• User has successfully login. • Error message displayed "Incorrect password".
ACTUAL RESULT	User has successfully login.
VERDICT	Pass

Table 4.2 Test case for Logout

ID	TC-2
DESCRIPTION	Check that the logged-in user has successfully logged out.
SETUP	Create an account for user with username natasha and password nj123!
INSTRUCTIONS	1. Click the logout button
EXPECTED RESULT	System displays log in page.
ACT/UAL RESULT	As expected
VERDICT	Pass

Table 4.3 Test case for Register Account

ID	TC-3
DESCRIPTION	Check is a user can successfully register account.
SETUP	1. Display the registration screen.
INSTRUCTIONS	1. Enter name Sara. 2. Enter username sjaved. 3. Enter password "nj123!". 4. Enter email"sara@gmail.com". 5. Enter contact number "03246558562". 6. Click the submit button.
EXPECTED RESULT	"Successful registration" message is displayed.
ACTUAL RESULT	As expected
VERDICT	Pass

Table 4.4 Test case for Update Account

ID	TC-4

DESCRIPTION	Check that the user can update his/her account information.
SETUP	1. Provide name in the name field Sara. 2. Provide username in the username field sjaved. 3. Provide password in the password field snj123!. 4. Provide email in the email field sara23@gmail.com. 5. Provide number in the number field 0324655846
INSTRUCTIONS	1. Open Profile 2. Enter new username sjavedRajput. 3. Click the submit button.
EXPECTED RESULT	System has updated the information.
ACTUAL RESULT	As expected
VERDICT	Pass

Table 4.5 Test case for Delete Account

ID	TC-5
DESCRIPTION	Check that the user can delete account.
SETUP	1. Login account with username and password. 2. Display the profile page.
INSTRUCTIONS	1. Click the delete button.
EXPECTED RESULT	Account deleted successfully.
ACTUAL RESULT	Account deleted successfully.
VERDICT	Pass

Table 4.7 Test case for Add Prescription

ID	TC-7

DESCRIPTION	Check if prescription can be added or not
SETUP	Account created with username sara password snj12!.
INSTRUCTIONS	1. Select prescription from menu. 2. Select add prescription. 3. Upload prescription picture. 4. Enter Advant in medicine name, tablet in type, 1 tablet in dosage, 80mg in gramage, twice a day in per day intake, and week in medicine course . 5. Click the Save&More button. 6. Enter Co-Eziday in medicine name, tablet in type, 1 tablet in dosage, 80mg in gramage, twice a day in per day intake, and week in medicine course . 7. Click the save button. 8. Select COD from payment mode. 9. Include test case 11 10. Select add prescription. 11. Select add prescription. 12. Upload prescription picture. 13. Enter nothing in medicine name, tablet in type, 1 tablet in dosage, 80mg in gramage, twice a day in per day intake, and month in medicine course . 14. Click the save button. 15. Enter Eziday in medicine name, tablet in type, 1 tablet in dosage, 80mg in gramage, twice a day in per day intake, and month in medicine course . 16. Click the save button.

EXPECTED RESULT	- First prescription is successfully added with medicines Advant and Co-Eziday. - Second prescription displays error message for medicine name missing - Third prescription doesn't proceed further as prescription picture is not uploaded.
ACTUAL RESULT	- As Expected - As Expected - As Expected .
VERDICT	Pass

Table 4.8 Test case for Modify Prescription

ID	TC-8
DESCRIPTION	Check if prescription can be modified or not
SETUP	1. Login account with admin username and password. 2. Prescription 1 is added but its order is not confirmed. 3. Prescription 2 is added and its order is also confirmed.

INSTRUCTIONS	1. Select prescription from menu. 2. Select prescription 1. 3. Enter Eziday instead of advent in medicine name 4. Press Save 5. Select prescription 2. 6. Enter Panadol instead of dispirin in medicine name 7. Press Save
EXPECTED RESULT	• First prescription is successfully modified • Second prescription displays error message "Sorry you can't edit as order is already confirmed"
ACTUAL RESULT	• As Expected • As Expected
VERDICT	Pass

Table 4.9 Test case for Delete Prescription

ID	TC-9
DESCRIPTION	Check if prescription can be deleted or not
SETUP	1. Login account with admin username and password. 2. Prescription 1 is added but its order is not confirmed. 3. Prescription 2 is added and its order is also confirmed.

INSTRUCTIONS	1. Select prescription from menu. 2. Select prescription 1. 3. Press delete 4. Select prescription 2. 5. Press delete
EXPECTED RESULT	• First prescription is successfully deleted • Second prescription displays error message "Sorry you can't edit as order is already confirmed"
ACTUAL RESULT	• As Expected • As Expected
VERDICT	Pass

Table 4.10 Test case for Set Schedule

ID	TC-10
DESCRIPTION	Check if Schedule can be set or not
SETUP	1. Account created with username natasha password nj123! 2. Prescription added
INSTRUCTIONS	8. Select fortnight from the options 9. Enter save 10. Don't select any option. 11. Enter Save
EXPECTED RESULT	• Fortnight option set for first schedule. • No schedule is set for prescription

ACTUAL RESULT	• As Expected • As Expected .
VERDICT	Pass

Table 4.11 Test case for Modify Schedule

ID	TC-11
DESCRIPTION	Check if schedule can be modified or not
SETUP	1. Account created with username nat password nj123! 2. Prescription 1 is added and schedule weekly has been set but its order is not confirmed. 3. Prescription 2 is added and schedule monthly has been set its order is also confirmed.
INSTRUCTIONS	1. Select Schedules from menu. 2. Select prescription 1. 3. Select its schedule and change it to fortnight. 4. Press save. 5. Select prescription 2. 6. Select its schedule and change it to weekly. 7. Press save.
EXPECTED RESULT	• First schedule is successfully modified. • For Second schedule error message "Sorry you can't edit as order is already confirmed" is displayed

ACTUAL RESULT	• As Expected • As Expected
VERDICT	Pass

Table 4.12 Test case for Delete Schedule

ID	TC-12
DESCRIPTION	Check if schedule can be deleted or not
SETUP	1. Account created with username nat password nj123! 2. Prescription 1 is added and schedule weekly has been set but its order is not confirmed. 3. Prescription 2 is added and schedule monthly has been set its order is also confirmed.
INSTRUCTIONS	1. Select Schedules from menu. 2. Select prescription 1 from schedules list. 3. Press delete 4. Select prescription 2 from schedules list. 5. Press delete 6. Select prescription 2.
EXPECTED RESULT	• First schedule is successfully deleted. • For Second schedule error message "Sorry you can't edit as order is already confirmed" is displayed
ACTUAL RESULT	• As Expected • As Expected

VERDICT	Pass

Table 4.13 Test case for Set Alarm

ID	TC-13
DESCRIPTION	Check if alarm can be set or not
SETUP	1. Account created with username sara password snj13! 2. Prescription 1 is added 3. Prescription 2 is added
INSTRUCTIONS	1. Select Alarm from menu. 2. Select prescription 1 from prescription list. 3. Select advant from schedules list. 4. Set time for two auto generated alarms. Set 10:00 am in first alarm and 7:00pm in next alarm. 5. Press Save 6. Select prescription 2 from prescription list. 7. Select Eziday from schedules list. 8. Set time for two auto generated alarms. Set 10:00 am in first alarm and 7:00pm in next alarm. 9. Don't Press Save
EXPECTED RESULT	• Alarm successfully set for prescription 1 • Alarm not successfully set for prescription 2 as save is not pressed.
ACTUAL RESULT	• As Expected • As Expected
VERDICT	Pass

Table 4.14 Test case for Modify Alarm:

ID	TC-14
DESCRIPTION	Check if alarm can be modified or not
SETUP	1. Account created with username Sara password nj123! 2. Prescription 1 is added. 3. Prescription 2 is added.
INSTRUCTIONS	1. Select Alarm from menu. 2. Select prescription 1 from prescription list. 3. Select advant from schedules list. 4. Modify time for two auto generated alarms. Set 10:00 am in first alarm to 9:00am and 7:00pm to 8:00pm in next alarm. 5. Press Save. 6. Select prescription 2 from prescription list. 7. Select Eziday from schedules list. 8. Set time for two auto generated alarms. Set 10:00 am in first alarm and 7:00pm in next alarm. 9. Don't Press Save
EXPECTED RESULT	• Alarm successfully set for prescription 1 • Alarm not successfully set for prescription 2 as save is not pressed.

ACTUAL RESULT	- As Expected - As Expected
VERDICT	Pass

Table 4.15 Test case for Log Condition

ID	TC-15
DESCRIPTION	Check if condition against medicine can be logged or not
SETUP	1. Account created with username Sara password nj123! 2. Prescription added
INSTRUCTIONS	1. Log in with username sara password nj123!. 2. Select log condition from the menu. 3. Select prescription 1 from the prescription list. 4. Now rate against Advant's 9:00 am intake time by 😐 And write No Effect in comment box 5. Now rate against Co-Eziday 9:00 am intake time by 🙁 And write Dizziness in comment box
EXPECTED RESULT	- Successfully logged. - Successfully logged.
ACTUAL RESULT	- As Expected

	• As Expected .
VERDICT	Pass

Table 4.16 Test case for Track Order:

ID	TC-16
DESCRIPTION	Check if order can be tracked or not
SETUP	1. Account created with username Sara password nj123! 2. Prescription added
INSTRUCTIONS	1. Log in with username sara password nj123!. 2. Select Track Order from menu 3. Select Prescription 1 4. View the order status
EXPECTED RESULT	• Order placed date, status of order, estimated tie of delivery displayed.
ACTUAL RESULT	• As Expected .
VERDICT	Pass

Table 4.17 Test case for Update Order Status:

ID	TC-17
DESCRIPTION	Check if order status can be updated or not
SETUP	1. Account created with username admin password nj123!

	2. Prescription added
INSTRUCTIONS	1. Log in with username sara password nj123!.Select Schedule from menu 2. Select in the row of order confirm field and select "Yes".
EXPECTED RESULT	• Order is confirmed.
ACTUAL RESULT	• As Expected
VERDICT	Pass

Chapter 5

Software Implementation Document

5.1 Introduction

This document describes the implementation details of "MedDeliver", the language selected, the IDE used and third party libraries and services integrated.

Language Selection

The project is implemented in the following languages:

- **Kotlin**
  Kotlin is a general purpose, open source, statically typed "pragmatic" programming language for the JVM and Android that combines object-oriented and functional programming features. [3]

Tool Selection

- **Android Studio**
  Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. [4]

Services and Libraries

- **Firebase**
  Firebase is Google's mobile platform that helps people quickly develop high-quality apps and grow their business. [5]
    - User Authentication
    - Real-time Database
    - Cloud Storage
    - Cloud Functions
- **Picasso**
  A powerful image downloading and caching **library** for Android [6]

5.2 Detail Description of Language Selected

Kotlin:

Google on May 9, 2019 announced Kotlin to be the official and preferred language for android development saying

"Android development will become increasingly Kotlin-first. If you're starting a new project, you should write it in Kotlin; code written in Kotlin often mean much less code for you–less code to type, test, and maintain." [7]

There are a handful of features that Kotlin has that Java lacks. They are: [8]

Code Conciseness

Comparing a *Java class* with an equivalent *Kotlin class* demonstrates the conciseness of Kotlin code. For performing the same operation that the *Java class* does, a *Kotlin class* necessitates for less code.

Null Safety

Unlike Java, all types are *non-nullable* in Kotlin by default. However, there's a way around. In order to *assign* a *null value* to a *variable* in Kotlin, it is required to explicitly using "?"

Higher-Order Functions and Lambdas

A *higher-order function* is one that takes *functions* as *parameters* or *returns a function.* Also, Kotlin functions are *first-class*. This means that they can be stored in *data*

Type inference for variables:

In Kotlin you don't have to explicitly tell whether a variable is integer, double or String. The type of variable is inferred by the type of value with which it is.

Kotlin-Java interop:

Kotlin is designed with Java interoperability in mind. Existing Java code can be called from Kotlin in a natural way, and Kotlin code can be used from Java rather smoothly as well.

Data Classes

We frequently create classes whose main purpose is to hold data. In Kotlin, this is called a *data class*. Unlike java where you have to make explicitly make getters and setters to access the members you can simply access members by their name equivalent to getters and setters.

Example:

```
data class Person(val name: String) {   }

val person = Person("John")

person.name = 10
```

5.3 Detail Description of Tools Selected:

Android Studio-IDE:

Android studio is based on IntelliJ IDEA, which does all the functionality that Eclipse with ADT plug-in do, with lot more additional features. The initial version of android studio offers Gradle-based build support and quick fixes, lint tools to catch performance, usability, version compatibility and other problems and ProGuard and app-signing capabilities

5.3 Detail Description of Services and Libraries Used:

Firebase:

Firebase provides a real-time database and backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase's cloud

- Authentication:

  I have used firebase for authenticating users. The sign-in method Email/Password is used in my app
- Real-time database:

  Firebase manages all data real-time in the database. It also allows syncing the real-time data across all the devices.

  Admin can see orders made by customers in real time without the need of a refresh and can update their order's status which will allow the customers to track their order in real-time.
- Firebase Storage:

  Firebase storage allows to store user generated content such as photos.

  I have used firebase storage for uploading and downloading of prescription picture.

Picasso:

A powerful image downloading and caching **library** for Android

I have used this library in my app for displaying downloaded images such as prescription picture.

I have also used Picasso for resizing of my images as needed and placeholders in case of errors.

5.4 Application Screenshots:

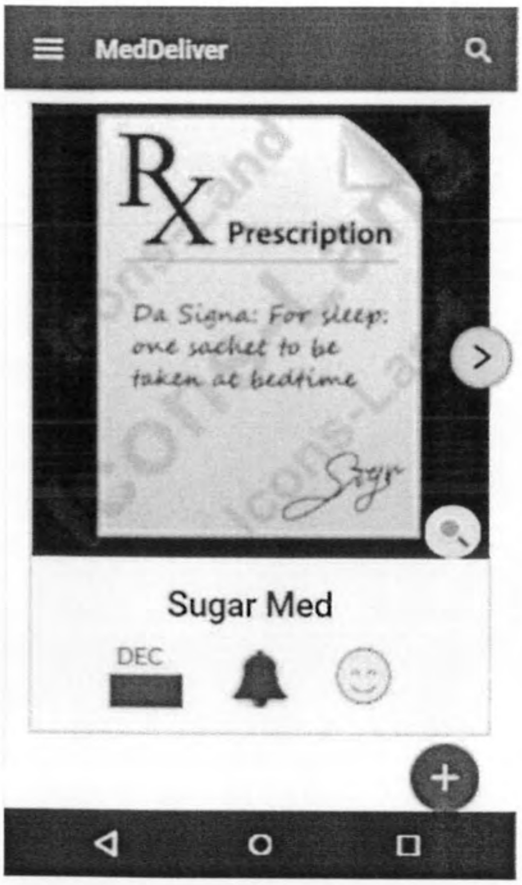
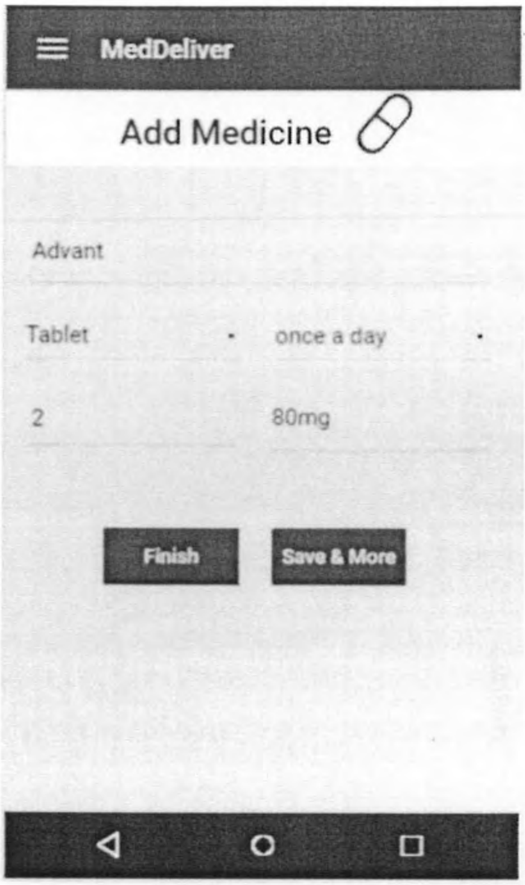


Figure 5.1 Prescription List

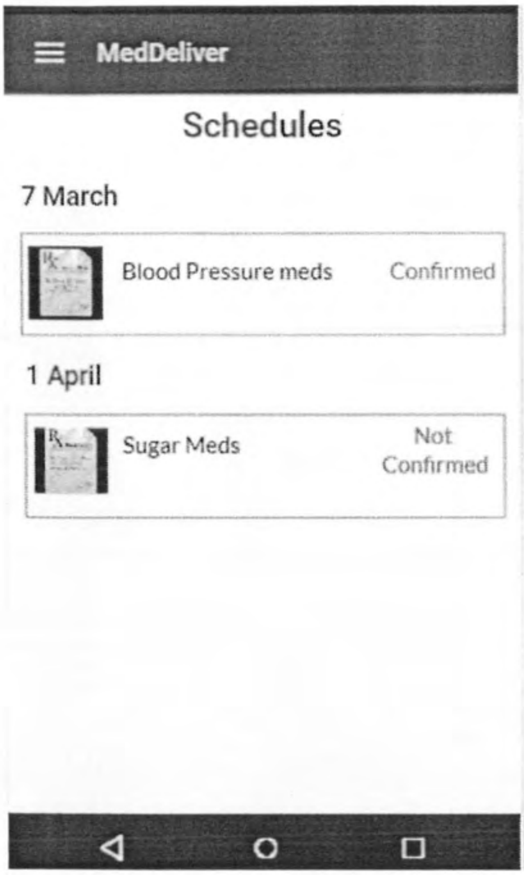


Advant

Tablet once a day

2 80mg

Figure 5.2 Add medicine



MedDeliver

Schedules

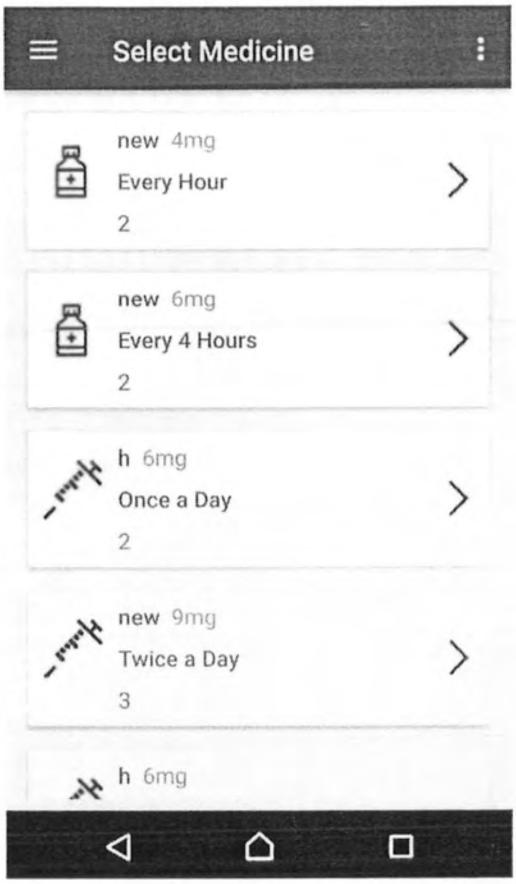
7 March

Blood Pressure meds Confirmed

1 April

Sugar Meds Not Confirmed

Figure 5.3 Schedule List



Select Medicine

new 4mg
Every Hour
2

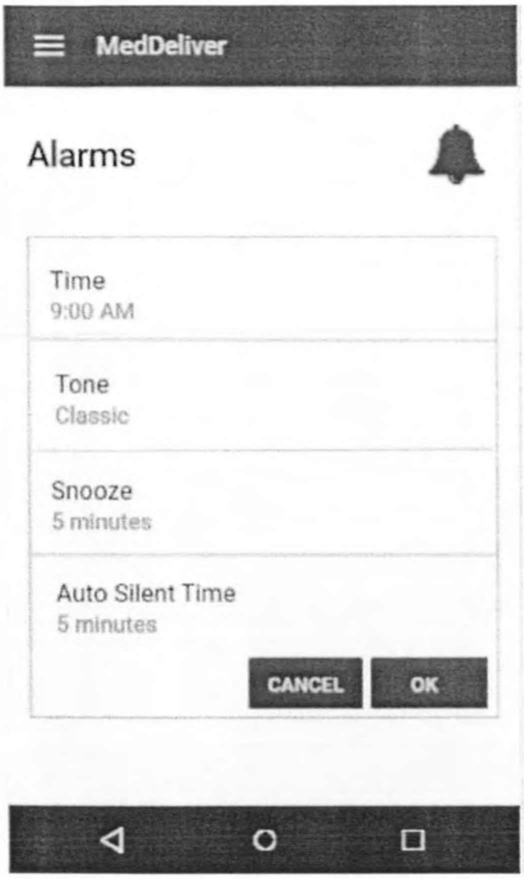
new 6mg
Every 4 Hours
2

h 6mg
Once a Day
2

new 9mg
Twice a Day
3

h 6mg

Figure 5.4 Medicine List



Alarms

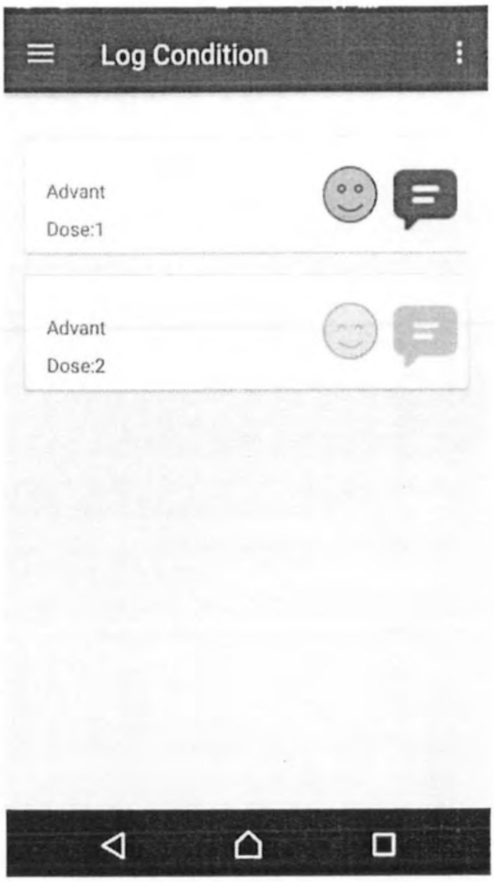
Time
9:00 AM

Tone
Classic

Snooze
5 minutes

Auto Silent Time
5 minutes

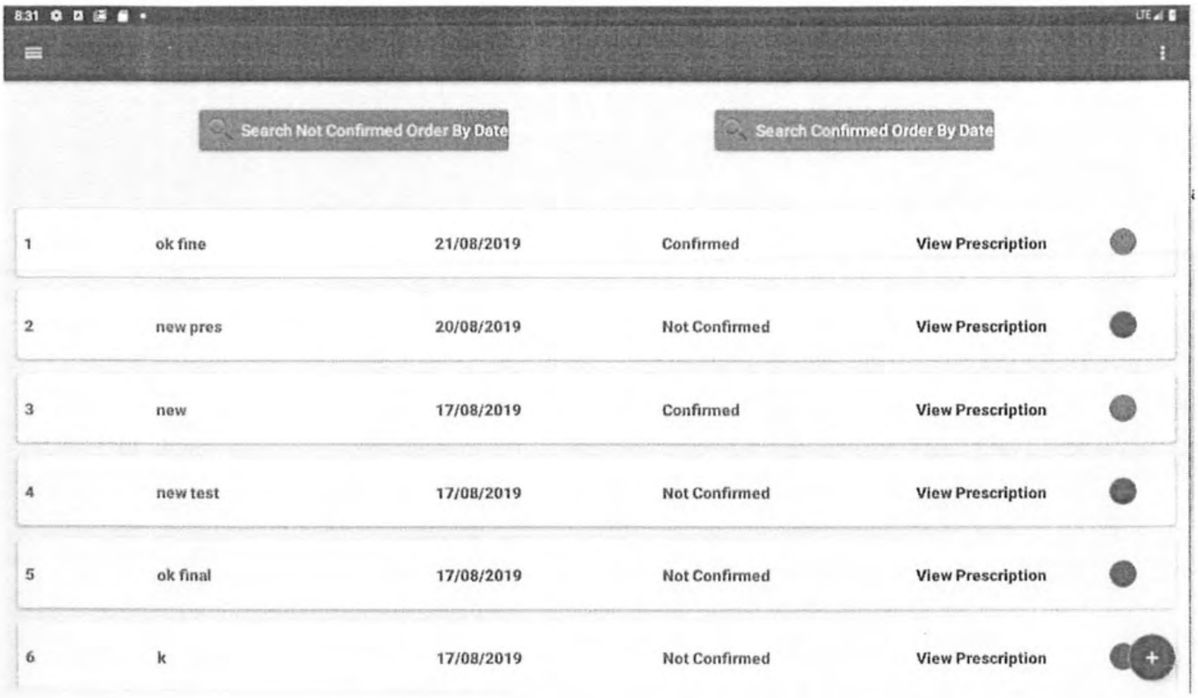
Figure 5.5 Set Alarm



Advant
Dose:1

Advant
Dose:2

Figure 5.6 Log Condition


8:31
Search Not Confirmed Order By Date
Search Confirmed Order By Date
+

1	ok fine	21/08/2019	Confirmed	View Prescription	●
2	new pres	20/08/2019	Not Confirmed	View Prescription	●
3	new	17/08/2019	Confirmed	View Prescription	●
4	new test	17/08/2019	Not Confirmed	View Prescription	●
5	ok final	17/08/2019	Not Confirmed	View Prescription	●
6	k	17/08/2019	Not Confirmed	View Prescription	●

Figure 5.7 Admin Schedule List

Chapter 6
Conclusion and Future Enhancements

6.1 Conclusions

There are thousands of patients in twin cities fighting with life-long diseases such as Blood Pressure, Diabetes etc. They need to take their medicines timely in order to stay healthy. Moreover, medicines need to be genuine considering there is a lack of check and balance from drug regulatory authority fake medicines are in abundance and it's quite a task finding genuine pharmacies

MedDeliver is a system that primarily, automates the process of medicine delivery as per patient's prescription and ensures genuine medicines reach the patient directly from authentic pharmacy in timely manner, this will save people time, and commute to pharmacies. Customer can add prescription and then set schedule on it for delivery. Admin get the updation in real-time and can can start working on the delivery process. Additionally, Filters allow admin to sort orders by the current date and order status so the process of delivery becomes efficient and smooth.

New orders will be automatically generated for those prescriptions on which a schedule is set, eliminating the need for placing order of same prescription again and again. This scheduling will primarily be helpful for people with diseases like Blood Pressure, Diabetes for whom it's crucial to take medcines regularly and on time

Additionally Alarms can be set so people are reminded of taking their medicines on time.

Lastly, people can benefit from MedDeliver by logging their condition. Whenever a patient starts new medication he/she faces some effects and he/she has to report the effects to the doctors but there is a chance you might miss some details or don't remember exactly when you felt a certain symptom which could be crucial in some cases. MedDeliver will automate this process for patients by allowing them to log their condition after every medicine dosage so a complete track is maintained throughout their medicine course and in order to provide a concise picture of patient's health Emojis are used for depicting Patient's condition along with an option for adding details of his condition .Then the patient can not only show this to their doctor but can also self-track his own condition.

6.2 Future Enhancements

In future application can be enhanced by:

- A view can be made for delivery person as well who will be notified of the confirmed orders in real-time and he can then deliver them to the person who placed the order. In addition to this, he can also add the delivery details such as delivered Date, bill and Pharmacy.
- Include tracking of orders through GPS so once the order is on the way, customer can view the live location of delivery person.
- Allow customers to add delivery location through GPS, so that they can get medicines delivered at specific location rather than only on the address mentioned in their profile.
- Allowing payments through credit/debit cards or JazzCash/EasyPaisa

References:

[1] "https://www.synopsys.com/software-integrity/resources/knowledge-database/software-architecture.html," [Online].

[2] "https://www.izenda.com/5-benefits-3-tier-architecture/," [Online].

[3] "https://kotlinlang.org/docs/reference/comparison-to-java.html," [Online].

[4] "https://developer.android.com/studio," [Online].

[5] "https://firebase.google.com/," [Online].

[6] "https://square.github.io/picasso/," [Online].

[7] "https://techcrunch.com/2019/05/07/kotlin-is-now-googles-preferred-language-for-android-app-development/," [Online].

[8] "https://kotlinlang.org/docs/reference/comparison-to-java.html," [Online].

[9] "https://techcrunch.com/2019/05/07/kotlin-is-now-googles-preferred-language-for-android-app-development/," [Online].