370 P.G.D.A.V. College (M), University Of Delhi

B. Sc. (H) Computer Science (II Year)

SEMESTER IV

Software Engineering Project

MEDILARM

SUPERVISOR:

Dr. Aparna Datt

SUBMITTED BY:

Meghansh Tyagi(3923)

Deepali Kalra(3927)

CERTIFICATE

This is to certify that the project entitled, “MEDILARM” has been done by: **MEGHANSH TYAGI** and **DEEPALI KALRA** of Bachelor of Science in Computer Science during semester IV from P.G.D.A.V.(M) College ,University of Delhi under the supervision of **Dr. APARNA DATT**.

DECLARATION

We hereby declare that this Project Report titled “MEDILARM” submitted to the Department of Computer Science, P.G.D.A.V.(M) College, University of Delhi is a record of original work done by the team under the guidance of **Dr. Aparna Datt**.

The information and data given in the report is authentic to the best of the team’s knowledge.

This Project Report is not submitted to any other university or institution for the award of any degree, diploma or fellowship or published any time before.

ACKNOWLEDGEMENT

We would like to take this opportunity to express our profound gratitude and deep regards to our teacher Dr. Aparna Datt for her exemplary guidance, monitoring and constant encouragement throughout the course of this project.

Our primary thanks to her, who poured over every inch of our project with painstaking attention and helped us throughout the working of the project.

It’s our privilege to acknowledge our deepest sense of gratitude to her for her inspiration which has helped us immensely. We are extremely grateful to her for unstilted support and encouragement in the preparation of this project.

INDEX

➔ **Pseudocode**

(I) SignUp Module

(II) LogIn Module

➔ **Control Flow Graph(CFG)**

(I) SignUp

(II) LogIn

➔ **Cyclomatic Complexity**

(I) SignUp

(II) LogIn

➔ **Testing**

(I) Black Box Testing

(II) White Box Testing

➔ **Test Cases**

(I) LogIn Module

(II) Refill and Dosage Module

➔ **Function Point Analysis**

➔ **Bibliography**

PROBLEM STATEMENT

In a household scenario especially during covid it has become immensely important not to run out of medicine that often comes in use, these include medicines for headaches, sanitary pads, vitamins, digestive medicines and so on. But medicines that are required every now and then and do not need a prescription are often forgotten about until the very last moment they are needed, which sometimes may often lead to serious issues or worsening of the patient's illness.This can be solved by setting up alarms for ordering each and every medicine but different medicines have different usage hence every medicine needs to be refilled in different timespan.

Along with this, We encounter that working people or aged people often forget to take their prescribed medicines at the right time which leads to delayed or often no recovery of the patient. Thus, Setting up alarms for the same can solve this problem.

Setting up alarms for each and every dose and medicine refills is a very difficult, tedious and time consuming task.

No medicine platform provides a solution to such problems as their main objective is to sell the medicines only. We want to provide proper medicinal support.

Solution To The Problem

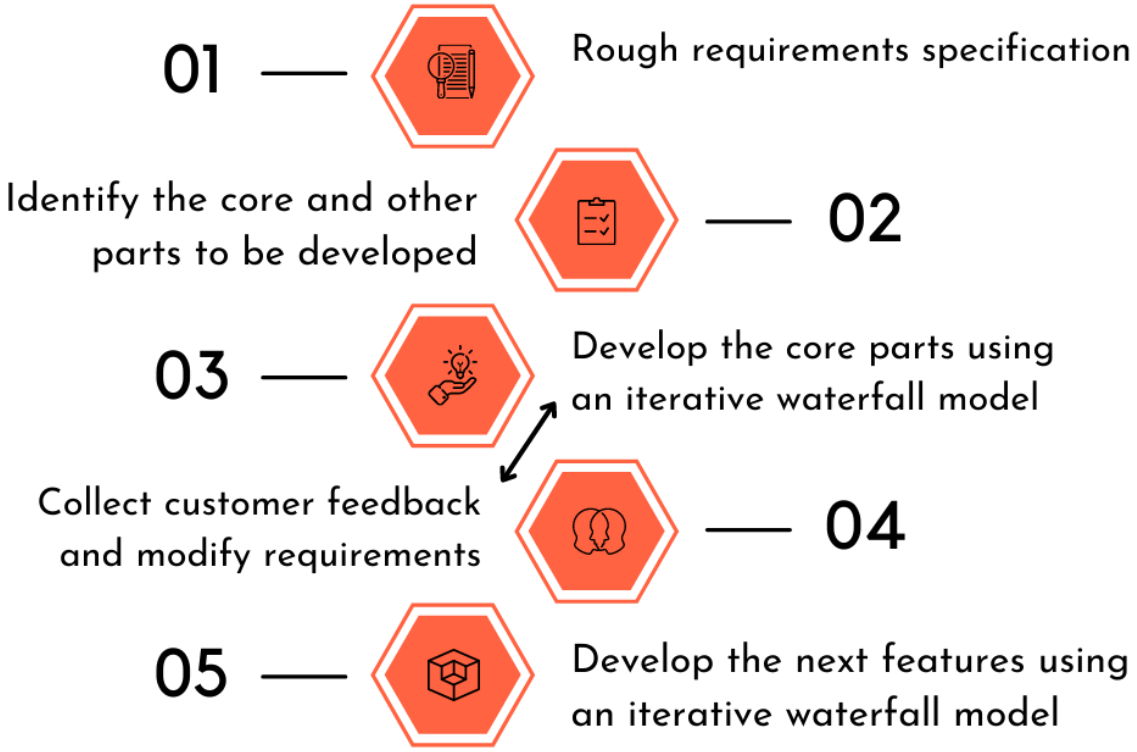
Creating an app which will take care of your dosage and refill timings and will remind you for the same accordingly and if needed automatically order medicine supplies when needed or direct you towards your desired medicine e-commerce platform so that you can order medicine manually. In case of a midway update by the doctor the dosage and refill mechanism will also be updated accordingly.

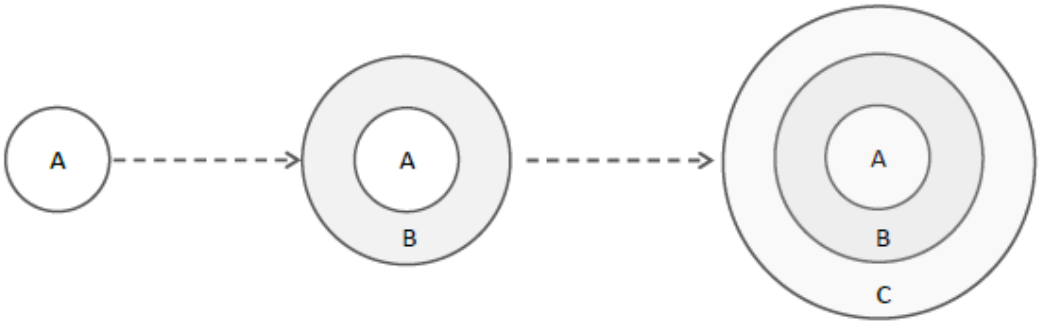
The application will take prescription as input either in digital format or by scanning a printed prescription by the user. Moreover the confirmation of the right prescription by the user will be mandatory. A proper authentication will be required to amend the prescription as any wrong information in prescription may lead to casualties.

Furthermore, Users who Whatsapp the prescription of their medicine to their local pharmacist, In this case the application can do so for the user.

PROCESS MODEL

Evolutionary Model





Evolutionary Development of a Software Product with new functionalities added after each iteration

The preferred model for our software implementation is the **Evolutionary Model** which is a combination of Iterative and Incremental process models.

Incremental Approach –

Evolutionary Model requires the project to be broken into modules which can be effectively achieved in our project by identifying the core modules which are refill tracking and updating, dosage tracking and updating along with additional functionalities of description scanning and ordering medicine by redirecting which can be incrementally constructed in subsequent modules once the core modules are fully developed and approved by the clients.

Iterative Approach –

The main objective of our Project is fulfilling the needs of the patient. Thus, user interaction is of utmost importance which is achieved during the development of each module through the iterative approach of the Evolutionary process model.

User satisfaction and requirement fulfillment can be elicited since each Module is reviewed by the users which will allow changes to take place according to the needs of the user until the module becomes fully functioning. Core modules will get tested thoroughly before additional functionalities are added which decreases the chances of encountering major errors and bugs at later stages of software development.

Software Requirements Specification

Requirement analysis use(why we need srs)

Overall objective, how its working with our project(development cost and time required)

- The user will get reminders for their refill and for their hourly dosage both having independent modules, if a user wants to use only single functionality

- A user can register by entering the prescription or refill data manually or by scanning; for this another module will be created which will scan and store the data in database

- For out project user authorization is of utmost importance since we are storing sensitive data such as prescriptions and refill data

- A user should also be given the option to update their prescription/refill data without having to reenter it.

- Once the reminder for refills is given to the user the application will ask for their confirmation and redirect them to whatsapp or desired E-commerce platform.

Entities -> User, Developers

Users can be divided into types :

1) Using our application for refill reminders.
2) Using our application for dosage reminders.
3) Using our application for both the purposes.

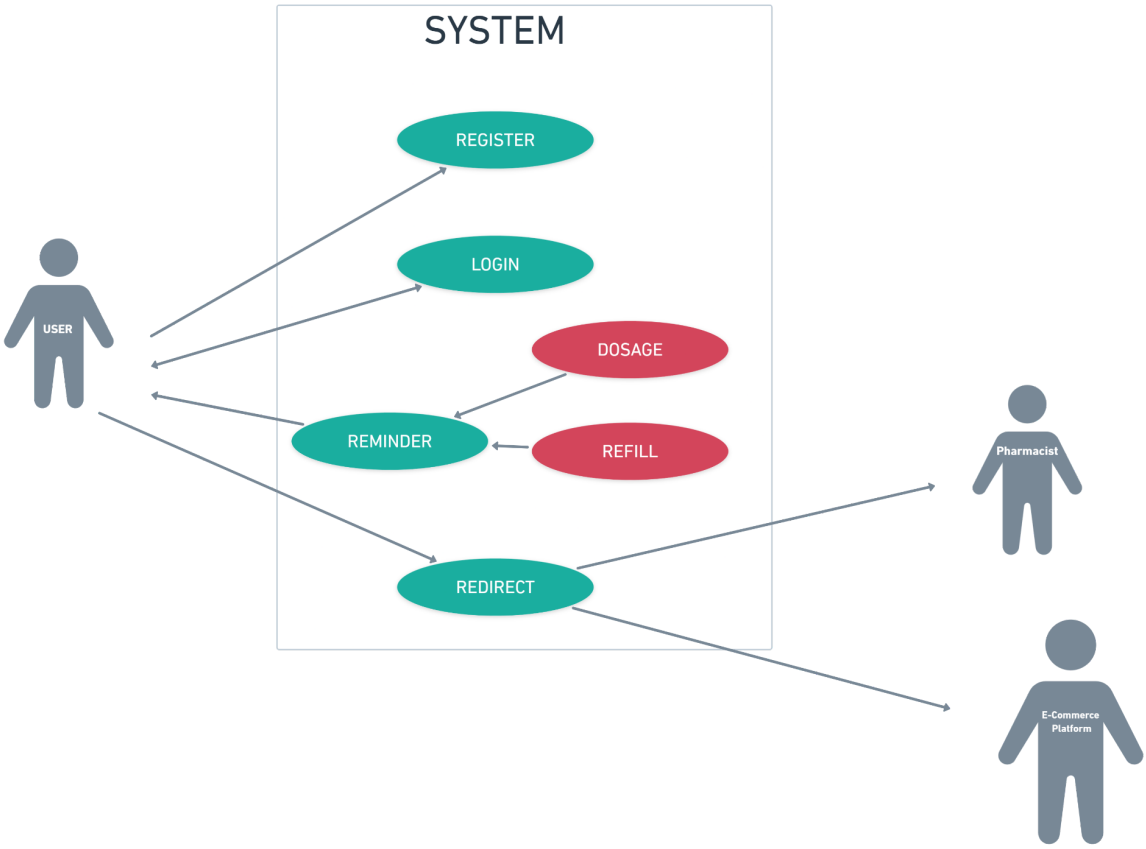
Developers

Project team is responsible for GUI and maintenance of our application

Modules/Functionalities

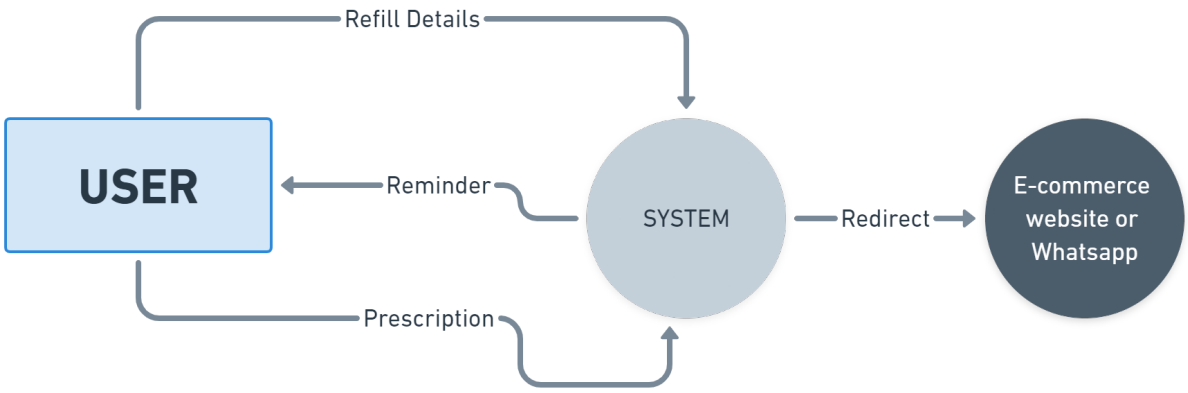
Login module, registration module,reminder module for dosage and refill,manual entry and scanning module, redirection to desired E-commerce platform/whatsapp

Use Case

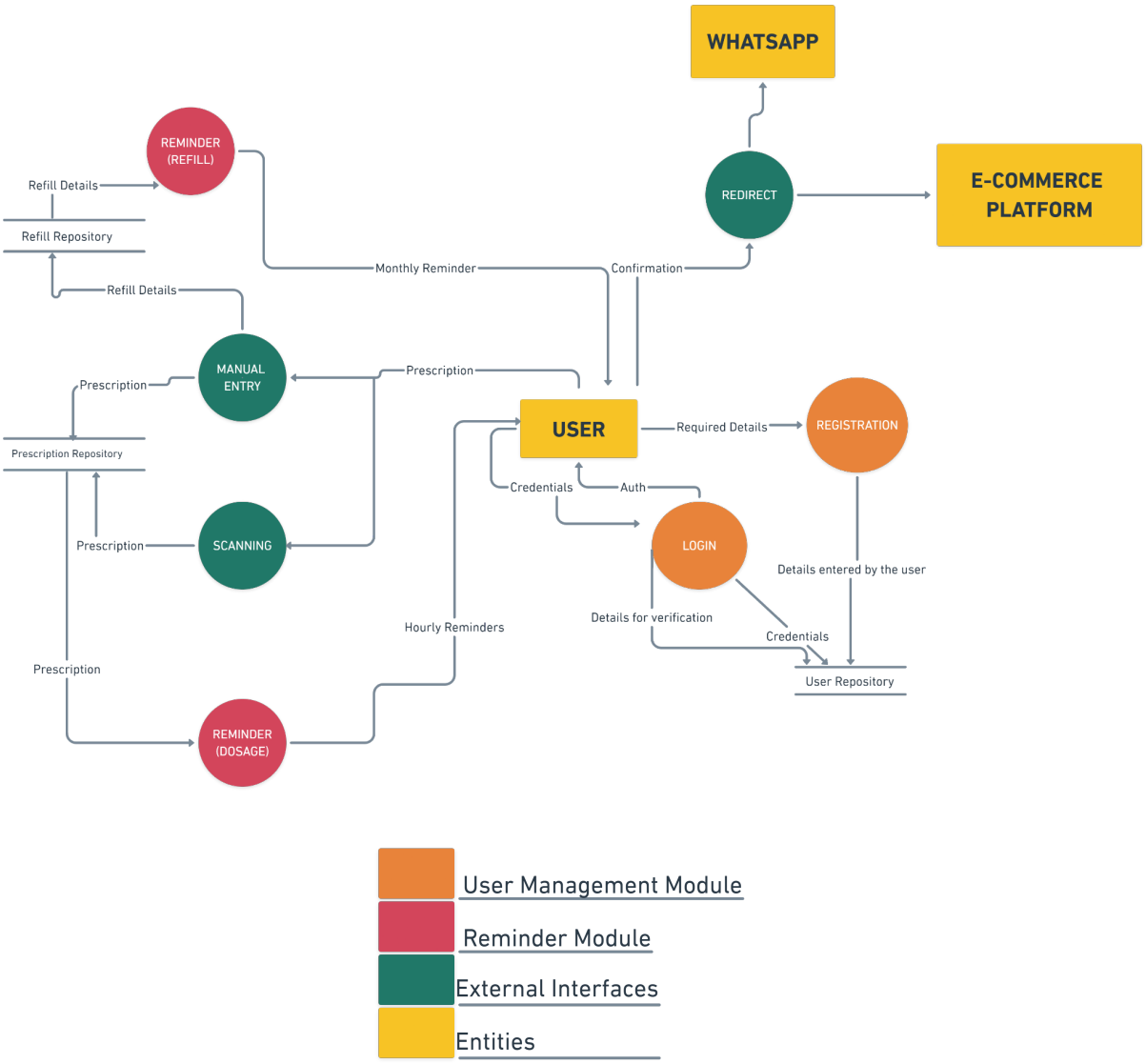

SYSTEM
REGISTER
LOGIN
DOSAGE
REMINDER
REFILL
REDIRECT
USER
Pharmacist
E-Commerce Platform

Data Flow Diagram

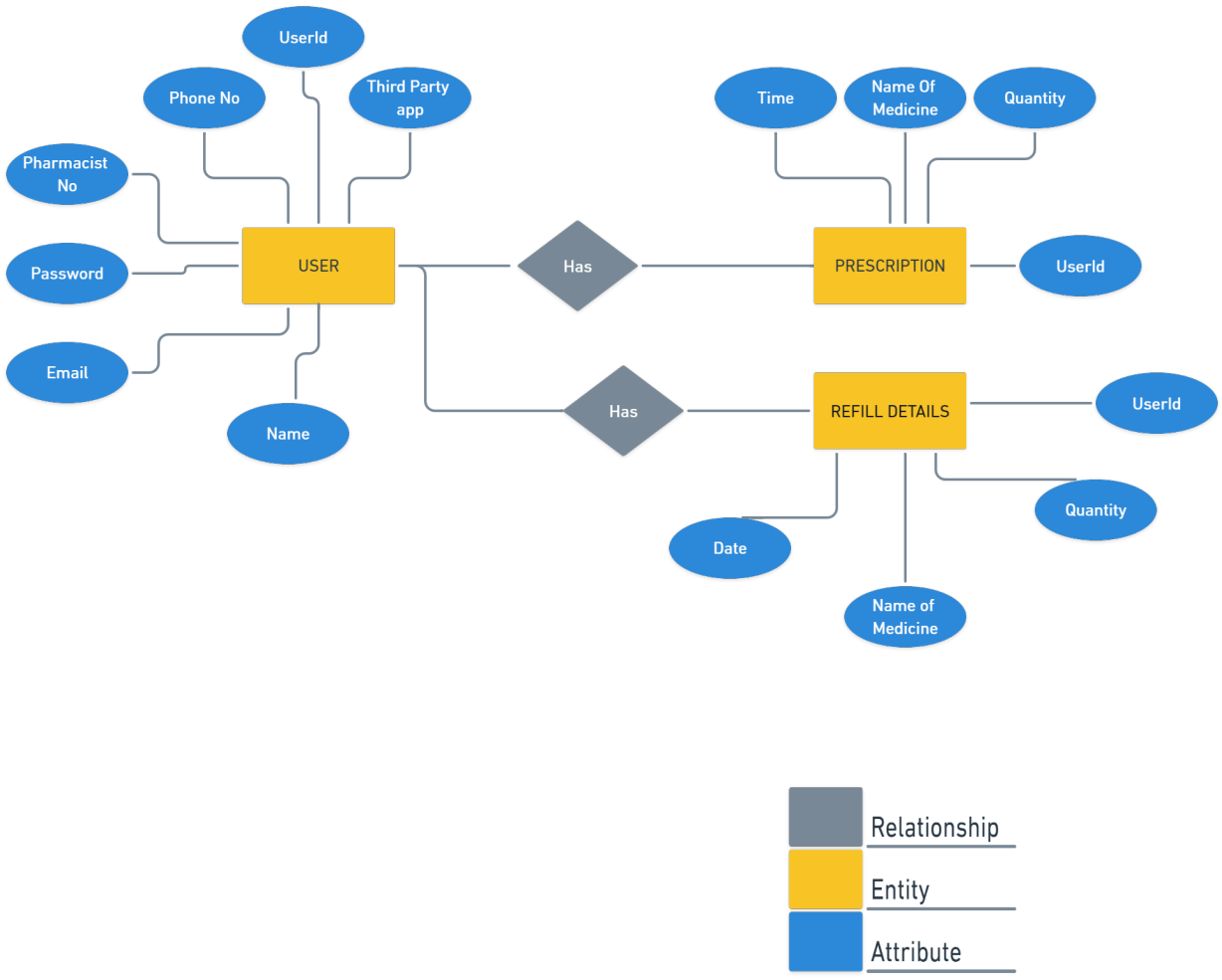
Level 0



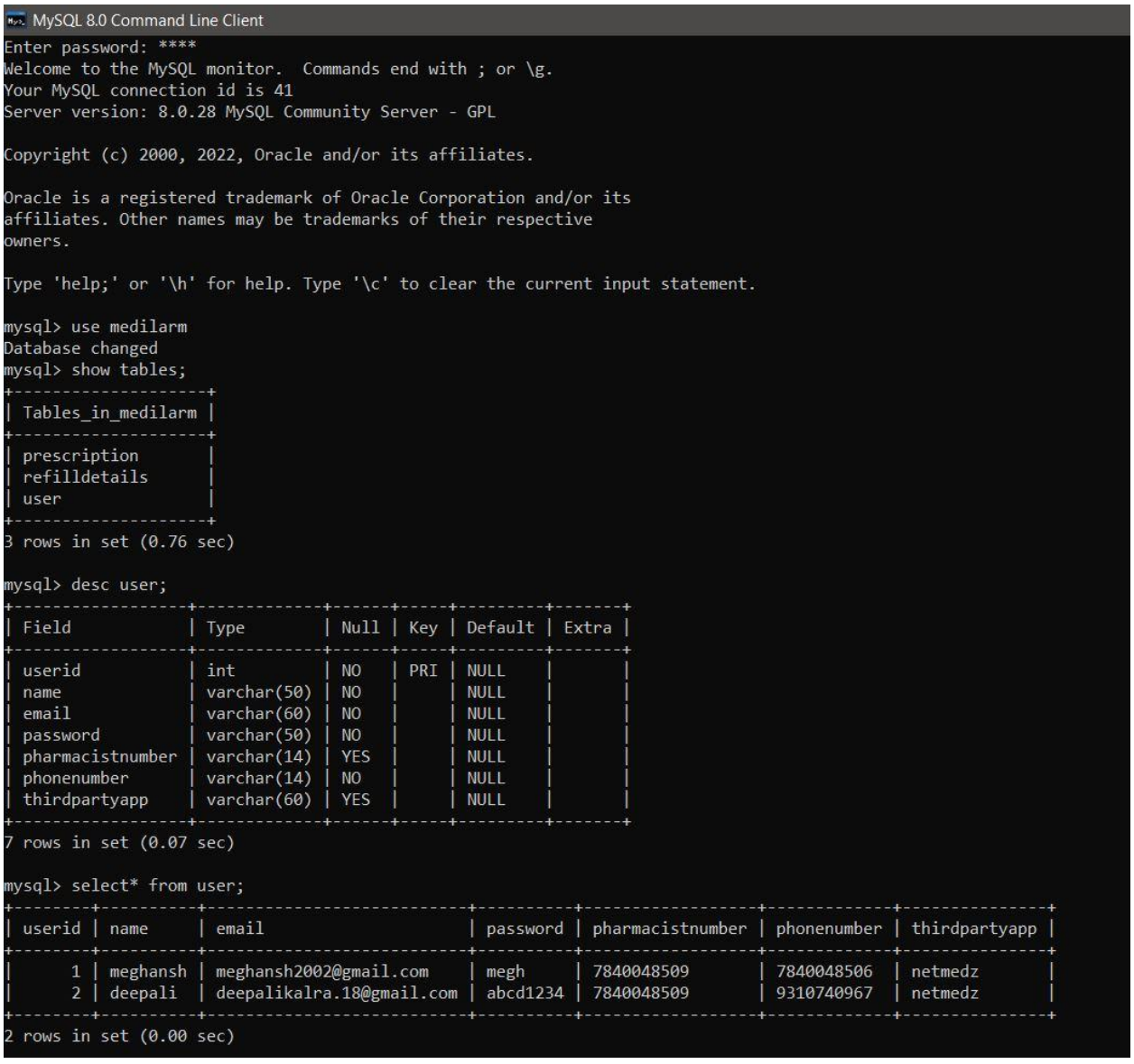
Level 1


WHATSAPP
REMINDER (REFILL)
Refill Details
Refill Repository
REDIRECT
E-COMMERCE PLATFORM
Monthly Reminder
Confirmation
Refill Details
MANUAL ENTRY
Prescription
Prescription
Prescription Repository
USER
Required Details
REGISTRATION
Credentials
Auth
Prescription
SCANNING
LOGIN
Details entered by the user
Hourly Reminders
Details for verification
Credentials
Prescription
User Repository
REMINDER (DOSAGE)
User Management Module
Reminder Module
External Interfaces
Entities

Entity Relationship Diagram



Relational Database Executed in SQL



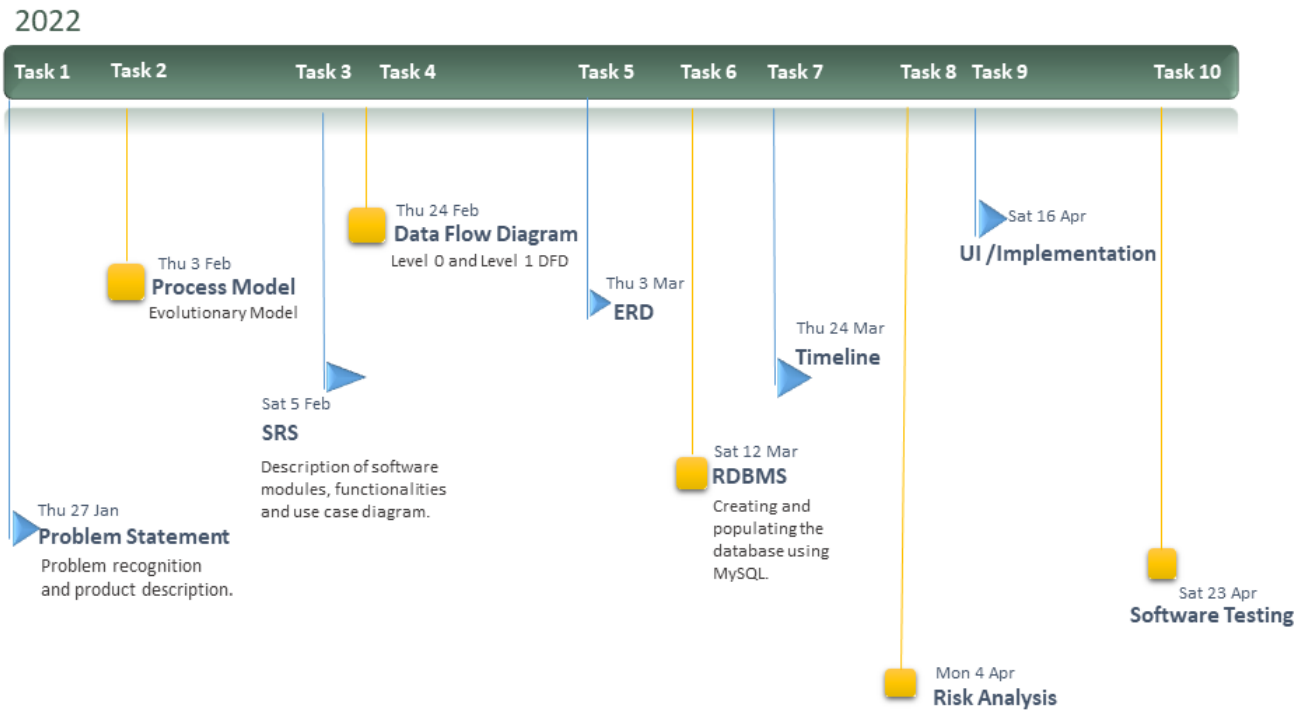
```
MySQL 8.0 Command Line Client
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 41
Server version: 8.0.28 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use medilarm
Database changed
mysql> show tables;
+--------------------+
| Tables_in_medilarm |
+--------------------+
| prescription       |
| refilldetails      |
| user               |
+--------------------+
3 rows in set (0.76 sec)

mysql> desc user;
+------------------+-------------+------+-----+---------+-------+
| Field            | Type        | Null | Key | Default | Extra |
+------------------+-------------+------+-----+---------+-------+
| userid           | int         | NO   | PRI | NULL    |       |
| name             | varchar(50) | NO   |     | NULL    |       |
| email            | varchar(60) | NO   |     | NULL    |       |
| password         | varchar(50) | NO   |     | NULL    |       |
| pharmacistnumber | varchar(14) | YES  |     | NULL    |       |
| phonenumber      | varchar(14) | NO   |     | NULL    |       |
| thirdpartyapp    | varchar(60) | YES  |     | NULL    |       |
+------------------+-------------+------+-----+---------+-------+
7 rows in set (0.07 sec)

mysql> select* from user;
+--------+----------+--------------------------+----------+------------------+-------------+---------------+
| userid | name     | email                    | password | pharmacistnumber | phonenumber | thirdpartyapp |
+--------+----------+--------------------------+----------+------------------+-------------+---------------+
|      1 | meghansh | meghansh2002@gmail.com   | megh     | 7840048509       | 7840048506  | netmedz       |
|      2 | deepali  | deepalikalra.18@gmail.com | abcd1234 | 7840048509       | 9310740967  | netmedz       |
+--------+----------+--------------------------+----------+------------------+-------------+---------------+
2 rows in set (0.00 sec)
```

MySQL 8.0 Command Line Client

```
mysql> desc refilldetails;
+----------------+--------------+------+-----+---------+-------+
| Field          | Type         | Null | Key | Default | Extra |
+----------------+--------------+------+-----+---------+-------+
| userid         | int          | NO   |     | NULL    |       |
| quantity       | int          | YES  |     | NULL    |       |
| nameofmedicine | varchar(300) | YES  |     | NULL    |       |
| dateofrefill   | date         | YES  |     | NULL    |       |
+----------------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> select* from refilldetails;
+--------+----------+--------------------------+--------------+
| userid | quantity | nameofmedicine           | dateofrefill |
+--------+----------+--------------------------+--------------+
|      1 |       15 | Dolo-650                 | 2022-04-12   |
|      1 |       10 | Genvol Plus              | 2022-04-14   |
|      1 |       14 | Famocid 40               | 2022-04-14   |
|      2 |       15 | Dolo-650                 | 2022-04-12   |
|      2 |       30 | Becadexamin Multivitamin | 2022-04-20   |
|      2 |       10 | Evion 400                | 2022-04-20   |
+--------+----------+--------------------------+--------------+
6 rows in set (0.05 sec)

mysql> desc prescription;
+----------------+--------------+------+-----+---------+-------+
| Field          | Type         | Null | Key | Default | Extra |
+----------------+--------------+------+-----+---------+-------+
| userid         | int          | NO   | MUL | NULL    |       |
| nameofmedicine | varchar(200) | YES  |     | NULL    |       |
| quantity       | int          | YES  |     | NULL    |       |
| timeofmedicine | time         | YES  |     | NULL    |       |
+----------------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> select* from prescription;
+--------+----------------+----------+----------------+
| userid | nameofmedicine | quantity | timeofmedicine |
+--------+----------------+----------+----------------+
|      1 | Dolo-650       |        1 | 13:00:00       |
|      1 | Chericof Syrup |        2 | 14:00:00       |
|      1 | Moxclav 625    |        1 | 14:00:00       |
|      1 | Moxclav 625    |        1 | 18:00:00       |
|      1 | Dolo-650       |        1 | 20:00:00       |
|      1 | Chericof Syrup |        2 | 22:00:00       |
+--------+----------------+----------+----------------+
6 rows in set (0.00 sec)
```

Gantt Chart

When creating a software project schedule, you begin with a set of tasks (the work breakdown structure). If automated tools are used, the work breakdown is input as a task network or task outline. Effort, duration, and start date are then input for each task. In addition, tasks may be assigned to specific individuals. As a consequence of this input, a time-line chart, also called a Gantt chart, is generated. A time-line chart can be developed for the entire project. Alternatively, separate charts can be developed for each project function or for each individual working on the project. The following figure illustrates the format of a time-line chart. It depicts a part of a software project schedule that emphasizes the concept scoping task for a word processing (WP) software product. All project tasks (for concept scoping) are listed in the left-hand column. The horizontal bars indicate the duration of each task. When multiple bars occur at the same time on the calendar, task concurrency is implied. The diamonds indicate milestones.



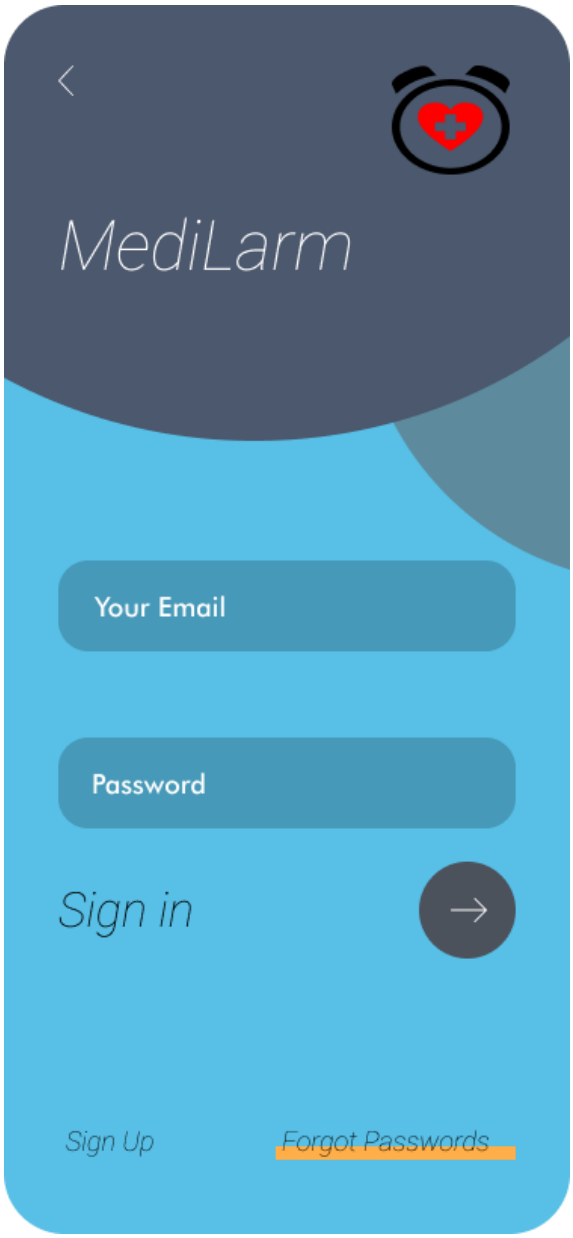
RISK ANALYSIS

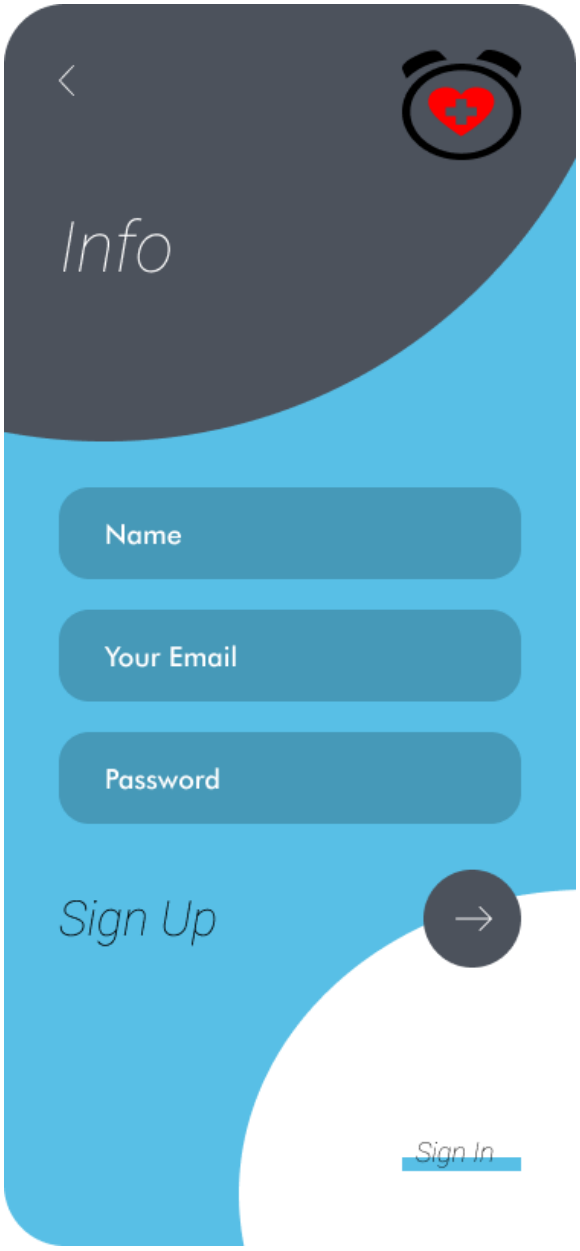
A risk table provides you with a simple technique for risk projection. During the risk analysis process every identified risk is considered and a perception/probability of the seriousness of that risk is made. Every risk is listed with their probability of occurrence alongside its impact depending upon an RMMM (Risk Mitigation, Monitoring and Management) plan is made.

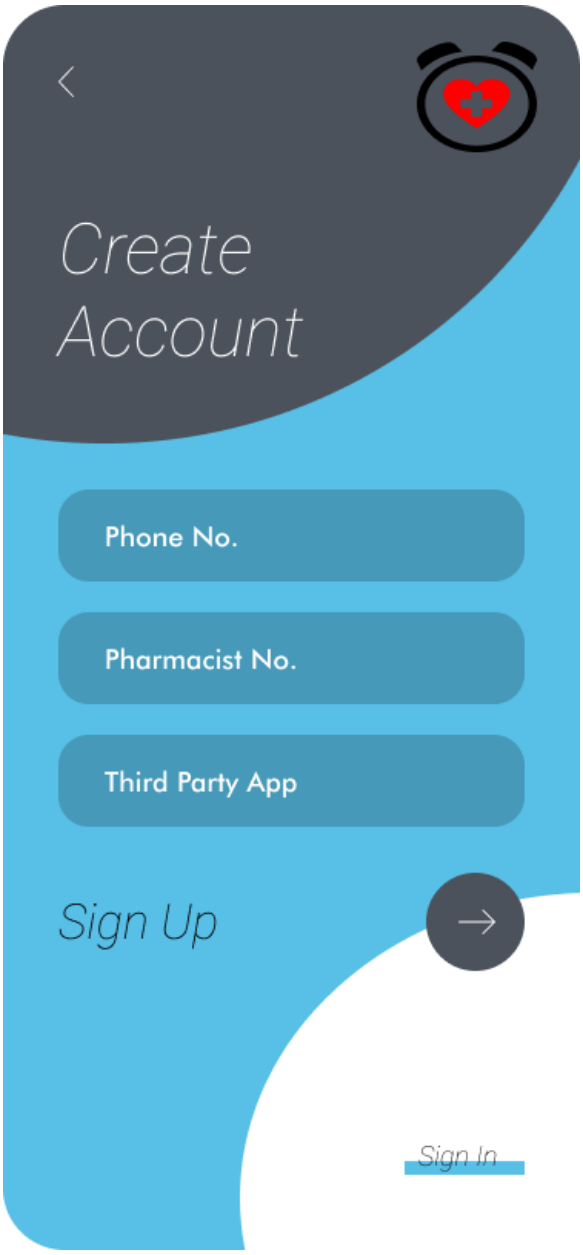
S.No.	Risk	Risk Category	Probability of Occurrence of Risk	Impact of Risk	Risk Factor/Risk Exposure(RE)	RMMM(Risk Mitigation, Monitoring and Management)
1.	Loss of user login data	Technical Risk	10%	5	0.5	Prioritizing security of database management
2.	Loss of prescription or refill data	Technical Risk	10%	5	0.5	Prioritizing security of database management
3.	Error in updating of prescription/refill	Technical Risk	20%	4	0.8	Re-verifying prescription/refill data before updating
4.	Error in scanning of prescription/refill	Technical Risk	20%	4	0.8	Re-scanning prescription/refill data before updating
5.	Failed connection with third party application	Technical Risk	20%	3	0.6	Using an external working module for redirection

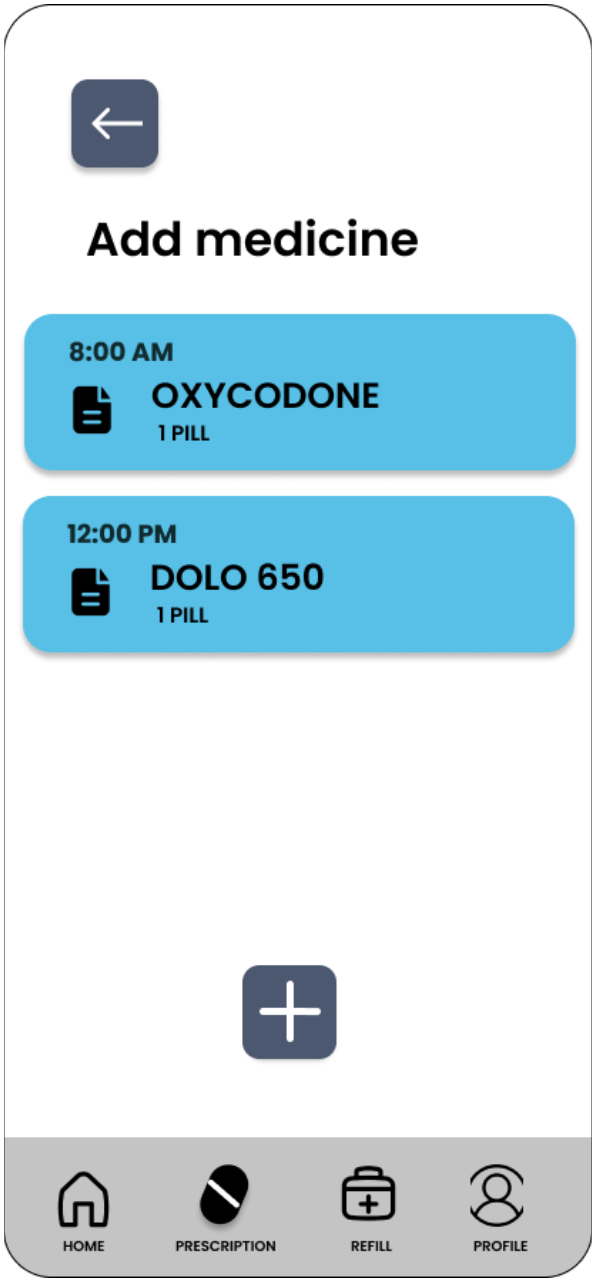
6.	Personnel shortage due to a member leaving the team	Technical Risk	30%	2	0.6	Having a backup team
7.	User dissatisfaction with the product	Business risk	20%	4	0.8	Getting user review of the product and making modifications according throughout the development

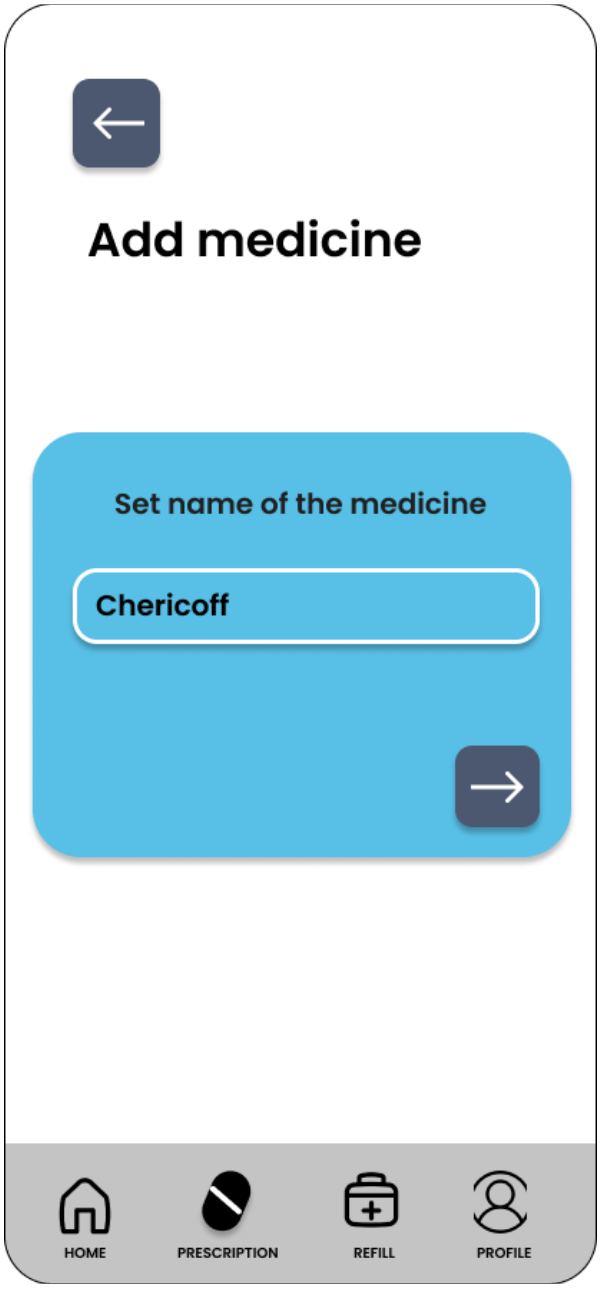
USER INTERFACE

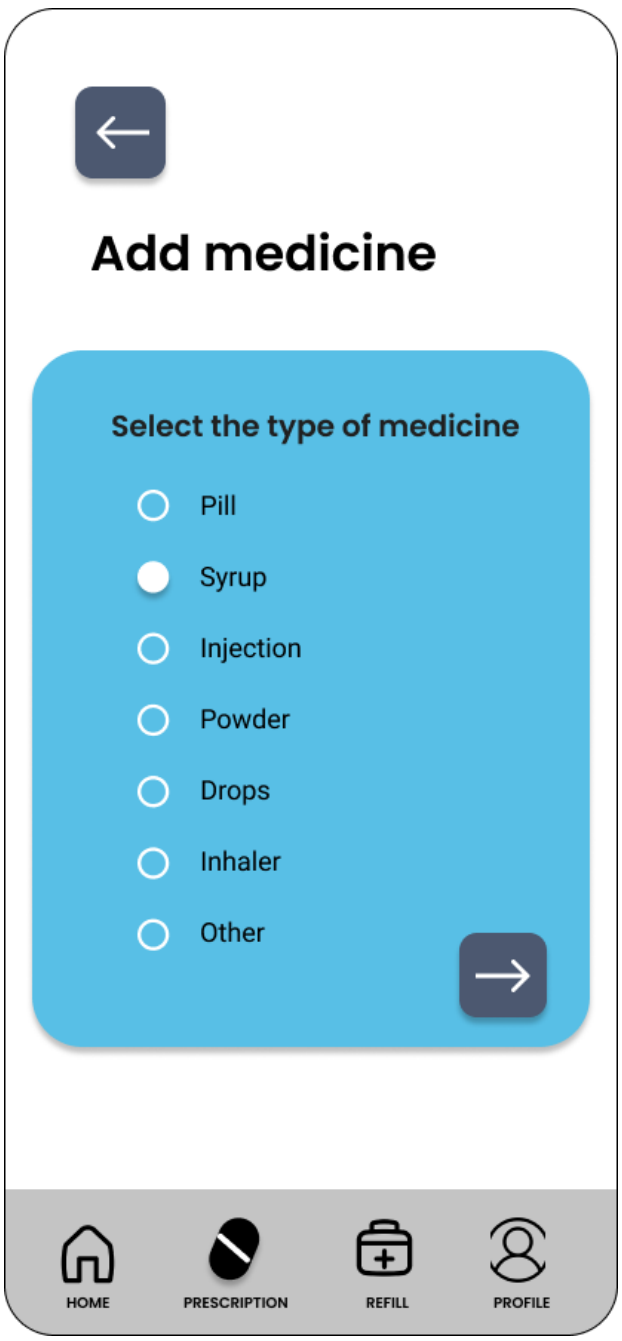


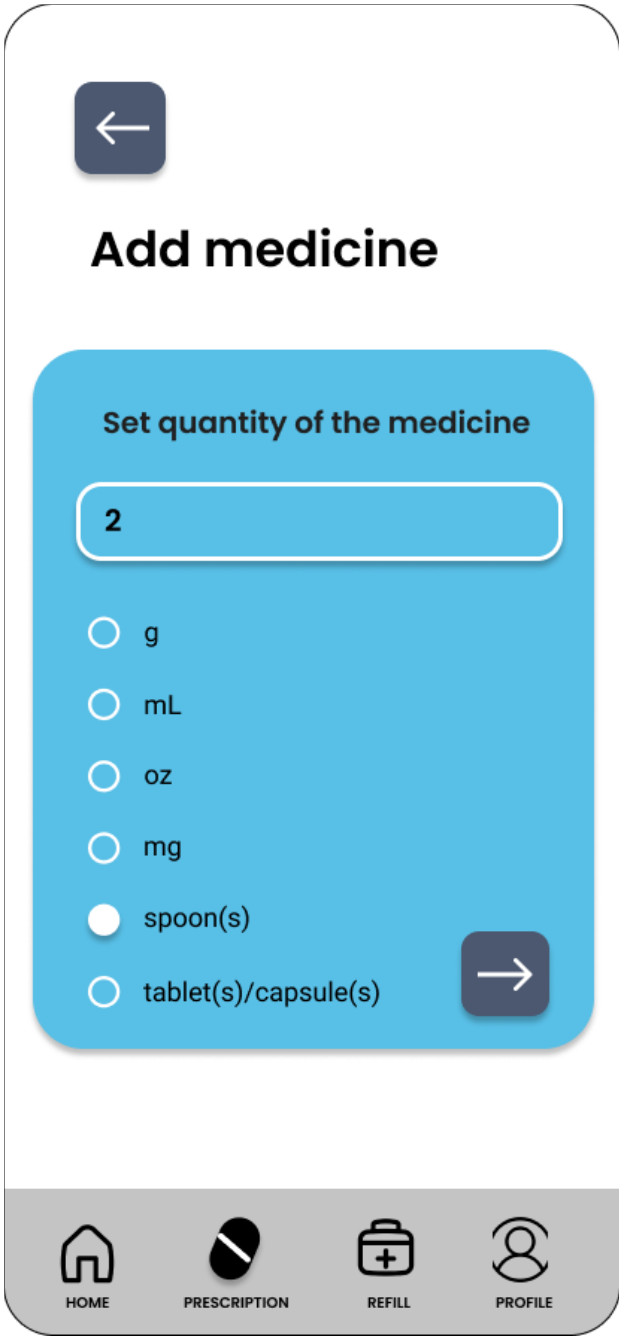

Info
Name
Your Email
Password
Sign Up
Sign In

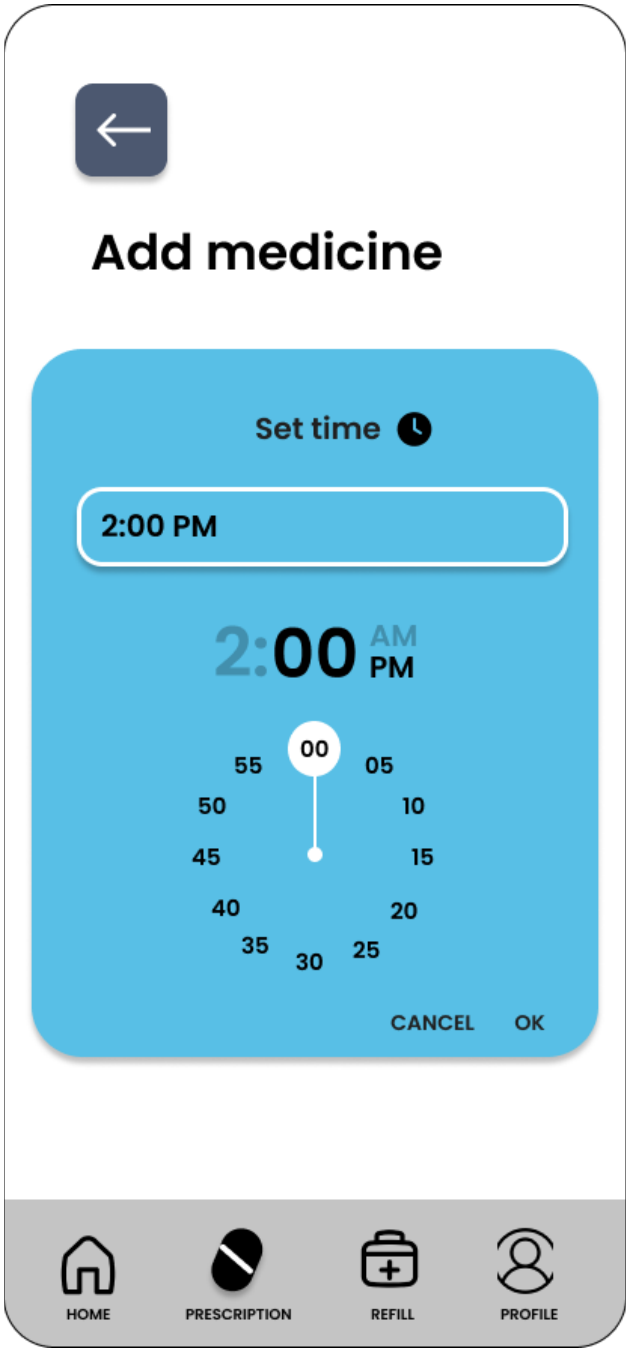

Create Account
Phone No.
Pharmacist No.
Third Party App
Sign Up
Sign In


Add medicine
8:00 AM
OXYCODONE
1 PILL
12:00 PM
DOLO 650
1 PILL
HOME
PRESCRIPTION
REFILL
PROFILE


Add medicine
Set name of the medicine
Chericoff
HOME
PRESCRIPTION
REFILL
PROFILE


Add medicine
Select the type of medicine
Pill
Syrup
Injection
Powder
Drops
Inhaler
Other
HOME
PRESCRIPTION
REFILL
PROFILE

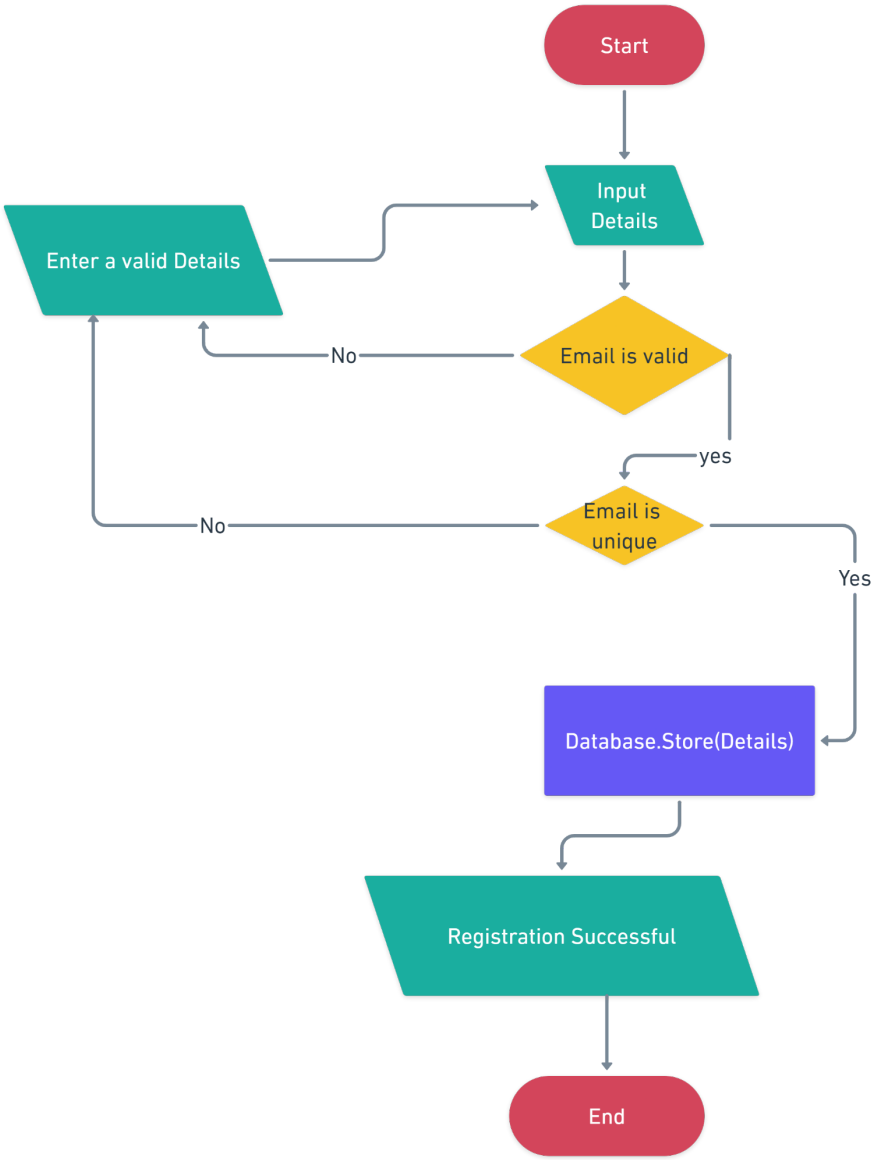

Add medicine
Set quantity of the medicine
2
g
mL
oz
mg
spoon(s)
tablet(s)/capsule(s)
HOME
PRESCRIPTION
REFILL
PROFILE


Add medicine
Set time
2:00 PM
2:00 AM PM
00
05
10
15
20
25
30
35
40
45
50
55
CANCEL
OK
HOME
PRESCRIPTION
REFILL
PROFILE

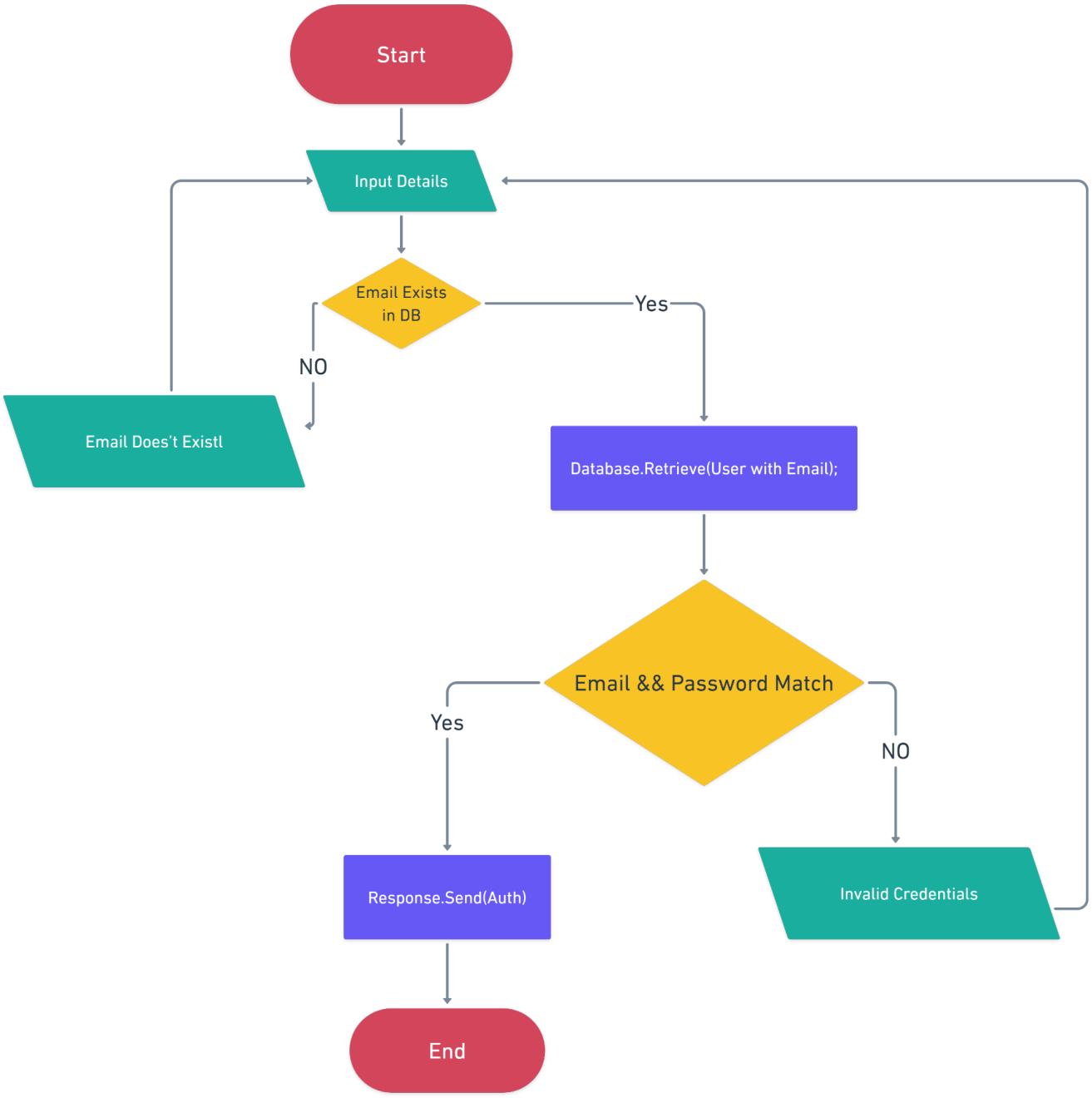
Flow Chart

A flowchart is the graphical or pictorial representation of an algorithm with the help of different symbols, shapes, and arrows to demonstrate a process or a program. With algorithms, we can easily understand a program. The main purpose of using a flowchart is to analyze different methods.

SignUp



LogIn


Start
Input Details
Email Exists in DB
Yes
NO
Email Does't Existl
Database.Retrieve(User with Email);
Email && Password Match
Yes
NO
Response.Send(Auth)
Invalid Credentials
End

Pseudocode

Pseudocode is a plain language description of the steps in an algorithm or another system. Pseudocode often uses structural conventions of a normal programming language, but is intended for human reading rather than machine reading. It typically omits details that are essential for machine understanding of the algorithm, such as variable declarations and language-specific code.

SignUp Module

1. Name = input from user;

   Email = input from user;

   Password = input from user;

   Pone_No = input from user;

   Pharmacist_No = input from user;

   Third_Party_app = input from user;

2. if(Email is valid && Email is unique){
3. database.user.store(Auto_Generated_UniqueID,Name,Email,Password,Phone_No,Pharmacist_No,Third_Party_app);
4. Output(“Registration Successful”);

}else{

5. Output(“Please enter correct details”);

}

6. return;

LogIn Module

1. Email = input from user;

   Password = input from user;

2. if(Email.exists_database()){
3. Details_database = retrieve_from_database(where User.Email == Email);
4. if(Email == Details_database.Email && Password == Details_database.Password){
5. response.send(auth);

}else{

6. Output("Invalid Credentials");

}

}else{

7. Output("Email doesn't exist");

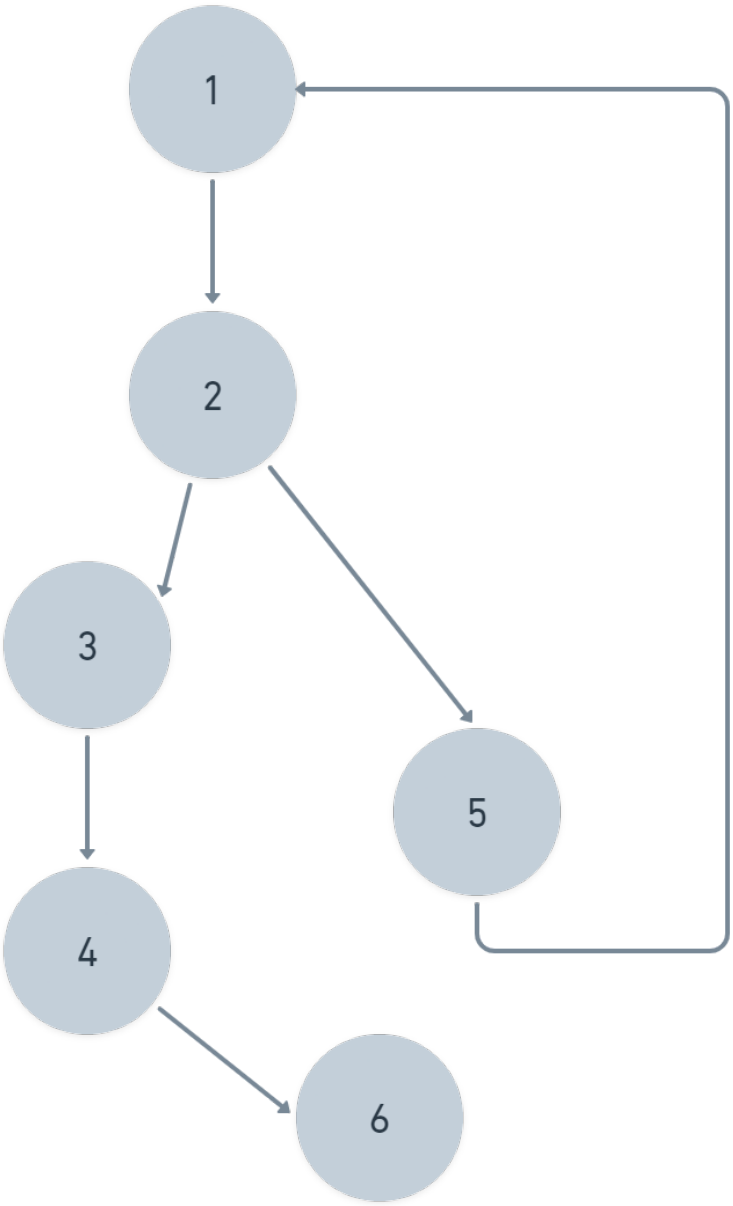
}

8. return;

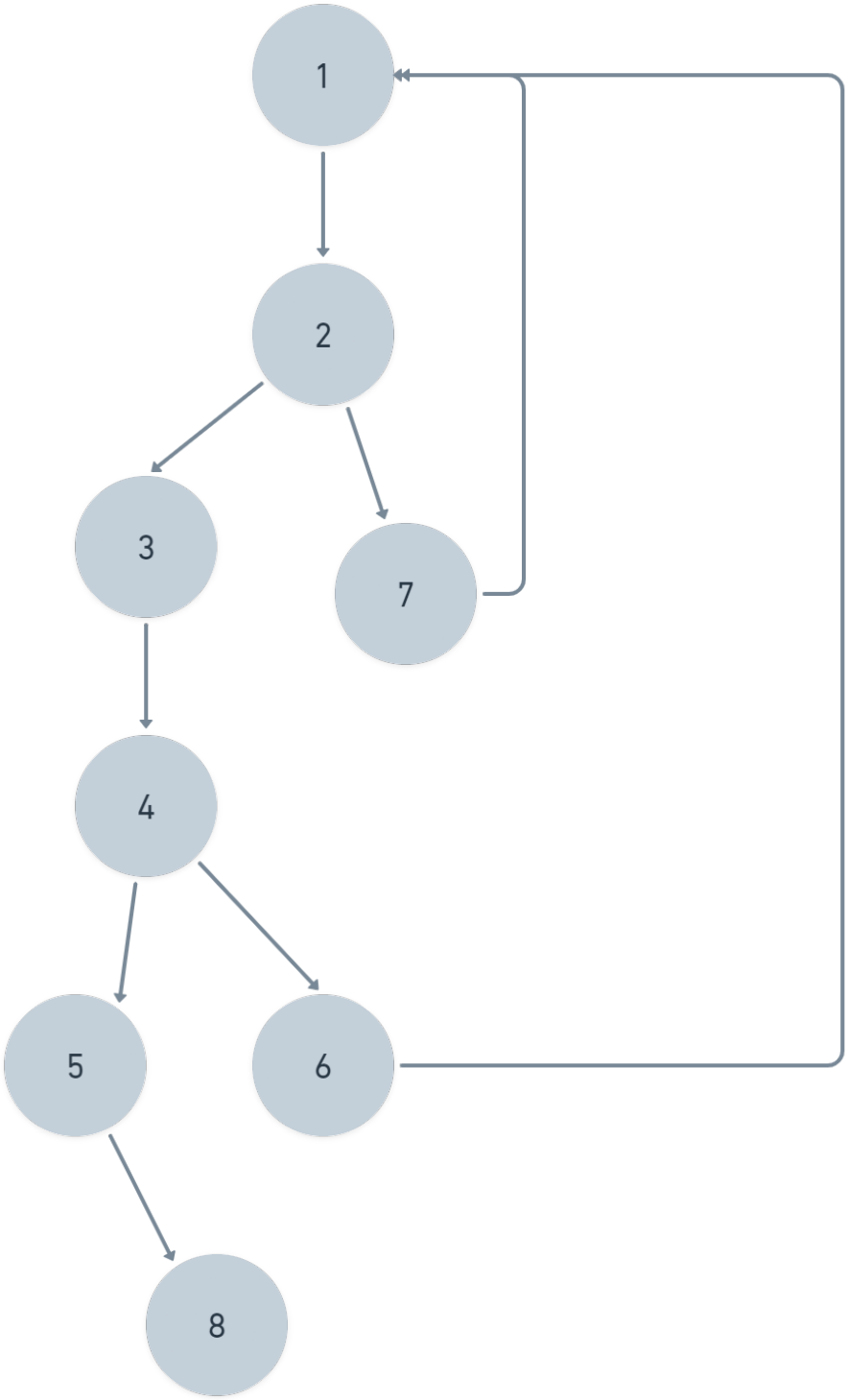
Control Flow Graph

A Control Flow Graph (CFG) is the graphical representation of control flow or computation during the execution of programs or applications. Control flow graphs are mostly used in static analysis as well as compiler applications, as they can accurately represent the flow inside of a program unit.

SignUp



LogIn



Cyclomatic Complexity

Cyclomatic complexity is a software metric used to indicate the complexity of a program. It is a quantitative measure of the number of linearly independent paths through a program's source code.It is computed using the control-flow graph of the program: the nodes of the graph correspond to indivisible groups of commands of a program, and a directed edge connects two nodes if the second command might be executed immediately after the first command. Cyclomatic complexity may also be applied to individual functions, modules, methods or classes within a program.

Cyclomatic Complexity = Number of Edges - Number of Nodes +2

SignUp

Method 1

Number of Edges = 6

Number of Nodes = 6

Cyclomatic Complexity(SignUp) = 6-6+2 = 2

Method 2

Number of closed regions = 1

Cyclomatic Complexity(SignUp) = 1+1(outside region) = 2

Method 3

Number of predicate nodes(π) = 1

Cyclomatic Complexity(SignUp) = π+1 = 1+1 = 2

LogIn

Method 1

Number of Edges = 9

Number of Nodes = 8

Cyclomatic Complexity(LogIn) = 9-8+2 = 3

Method 2

Number of closed regions = 2

Cyclomatic Complexity(SignUp) = 2+1(outside region) = 3

Method 3

Number of predicate nodes(π) = 2

Cyclomatic Complexity(SignUp) = π+1 = 2+1 = 3

TESTING

- Testing is the process of executing a program with the intent of finding an error. A good test case has a high probability of finding - undiscovered error.
- Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, coding.
- The purpose of product testing is to verify and validate the various work products viz. units, integrated unit, final product to ensure that they meet their requirements.

Black Box Testing:

- Black Box testing is also called functional testing.
- Black Box Testing is a test case design method that focuses on the functional requirements of the software that enables the software engineer to derive a set of input conditions that fully exercise all functional requirements for a program.
- Test the artifacts from the external point of view.
- Specifications are used to test data that is what type of input should be given to the unit or module should be specified.
- We can check the functionality on the basis of the output generated and the input, not looking at the internal coding.
- It attempts to find errors in the following categories
    1. Incorrect or missing functions
    2. Interface errors
    3. Errors in data structure or External database access
    4. Behavior or performance error
    5. Initialisation and termination errors

White Box Testing:

- It is also called glass box testing.
- White Box testing is a test case design method that uses the control structure of the procedural design to derive test cases.

- Using White Box Testing method, the software engineer can derive test cases that
    1. Guarantee that all independent paths within a module have been exercised at least once.
    2. Exercise all logical decisions on their true and false sides.
    3. Execute all loops at their boundaries and within their operational bounds.
    4. Exercise internal data structures ensure their validity.
- Test the artifacts from the internal point of view.
- It cannot detect absence of features.
- For security purposes the Email of the user is required in case he/she forgets his/her password and wants to retrieve that.

Test Cases

LogIn Module:

S.No.	Test Case	Expected Result
1	Valid Email and Valid Password	LogIn Successful
2	Valid Email and Invalid Password	LogIn Unsuccessful
3	Invalid Email and Valid Password	LogIn Unsuccessful
4	Non Existing Email and Invalid Password	LogIn Unsuccessful

Refill and Dosage Module

S.No.	Test Case	Expected Result
1	User enters wrong timings	User will receive reminders at wrong time
2	User enters wrong medicine	User will be reminded to take wrong medicine
3	Date of refill entered by user is not the actual required refill date	User will be reminded to refill the medicine either early or late
4	Medicine to be refilled entered by the user is wrong	Wrong medicine will be displayed while the refill reminder pops up
5	Pharmacist number entered by the user is incorrect	User will be redirected to whatsapp but the chat window will not open

Function Point Analysis

Function point analysis is a method of quantifying the size and complexity of a software system in terms of the functions that the system delivers to the user. It is independent of the computer language, development methodology, technology or capability of the project team used to develop the application.

What are functional points?

A Function Point (FP) is a unit of measurement to express the amount of business functionality, an information system (as a product) provides to a user. FPs measure software size. They are widely accepted as an industry standard for functional sizing

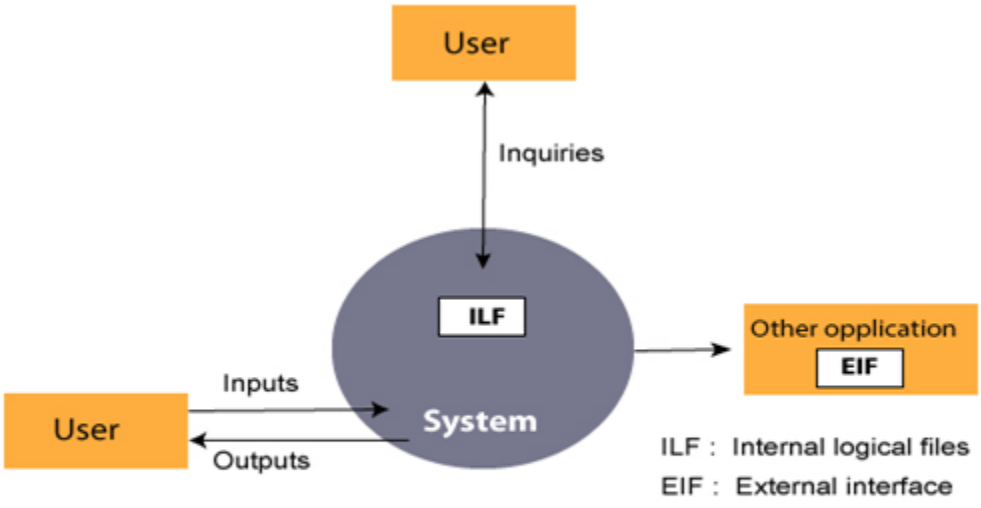
Number of user inputs/ External Inputs (EI): those items provided by the user that describe distinct application-oriented data (such as file names and menu selections)

Number of user outputs/External outputs (EO): those items provided to the user that generate distinct application-oriented data

Number of user inquiries/External inquiries (EQ): interactive inputs requiring a response from the system

Number of External files (EIF) – machine-readable interfaces to or from other systems

Number of Internal files (ILF) – logical master files in the system



FPAs Functional Units System

EXTERNAL INPUTS:

1) User Credentials: Username, password
2) Prescription data: Time, name, type, and quantity of medicine
3) Refill data: Date of Refill, Quantity of refill, Medicine Name
4) Redirection Details: Pharmacist No./Third party application, User confirmation

EXTERNAL OUTPUTS:

1) User authorization
2) Refill reminder
3) Prescription reminder

LOGICAL INTERNAL FILES:

1) User Details
2) Prescription Details
3) Refill Details
4) Redirection Details

EXTERNAL INTERFACE FILES:

1) Transferal of refill details

Calculation

External Inputs=4

External Outputs=3

Internal Logical Files=4

External Interface Files=1

External Inquiries=0

Function units	Low	Average	High
EI	3	4	6
EO	4	5	7
EQ	3	4	6
ILF	7	10	15
EIF	5	7	10

Table : Unadjusted Functional Point

Information Domain Values	**Count**	**Weighing Factor - Simple**	**Weighing Count**
External Inputs	4	3	12
External Outputs	3	4	12
External Inquiries	0	3	0
Internal Logical Files	4	7	28
External Logical Files	0	5	0
External Interface Files	1	5	5
Count Total			57

Table : Function Point Table

S.NO.	Questions	VAF
1.	Does the system require reliable backup and recovery?	**4**
2.	Are specialized data communications required to transfer information to or from the application?	**5**
3.	Are there distributed processing functions?	**4**
4.	Is performance critical?	**3**
5.	Will the system run in an existing, heavily utilized operational environment?	**3**
6.	Does the system require online data entry?	**4**
7.	Does the online data entry require the input transaction to be built over multiple screens or operations?	**4**
8.	Are the ILFs updated online?	**5**
9.	Are the inputs, outputs, files, or inquiries complex?	**2**
10.	Is the internal processing complex?	**1**
11.	Is the code designed to be reusable?	**0**

12.	Are conversion and installation included in the design?	**2**
13.	Is the system designed for multiple installations in different organizations?	**3**
14.	Is the application designed to facilitate change and ease of use by the user?	**5**

Table : Project Estimators

Degree	Significance
0	Not Present
1	Insignificant Influence
2	Moderate Influence
3	Average Influence
4	Significant Influence
5	Strong Influence

Total Degree of Influence (TDI) = ∑(fi) = **45**

The fi (i = 1 to 14) are "*Complexity Adjustment Values*" based on responses.

Computing Function Points :

FP = Count_total *(0.65 + 0.01 *∑(fi)) [where,count total is the sum of all FP entries]

FP = 57*(0.65 + 0.01 * 45) = 57*(1.10) = **62.7**

BIBLIOGRAPHY

- Software Engineering – A Practitioner's Approach o By Roger S. Pressman
- Software Engineering- K.K. Aggarwaland Yogesh Singh
- En.wikipedia.org
- geegkforgeeks.org