The Hashemite University, Zarqa, Jordan
Faculty of Prince Al-Hussein Bin Abdallah II for Information Technology
Computer Information Systems Department

PHARMA ROAD

A project submitted.
in partial fulfillment of the requirements for the
B.Sc. Degree in Business Information Technology

By

DANIA HABIS MAHMOUD ALHALAIGAH (2038221)
REEM ISSA MOHAMAD ABUSIRDANEH (2036909)
TAMARA NOURDDINE H. AMIS (2034573)
ZAIN MAHDI KHALED AL-RAFATI (2035242)

Supervised by

Dr. Nadera Aljawabreh

Committee Member Names

Dr. Hind Ahmad Ismail Bani Milhem

Dr. Mamoon Yusef Mohammed Obiedat

Jan-2024

CERTIFICATE

It is hereby certified that the project titled **<Pharma Road>,** submitted by undersigned, in partial fulfillment of the award of the "bachelor's in **Business Information Technology**" embodies original work done by them under my supervision.

All the analysis, design and system development have been accomplished by the undersigned. Moreover, this project has not been submitted to any other college or university.

DANIA HABIS MAHMOUD ALHALAIGAH (2038221)	DANIA ALHALAIGAH
REEM ISSA MOHAMAD ABUSIRDANEH (2036909)	REEM ISSA
TAMARA NOURDDINE H. AMIS (2034573)	TAMARA AMIS
ZAIN MAHDI KHALED AL-RAFATI (2035242)	ZAIN AL-RAFATI

ABSTRACT

"Pharma Road" is a location-based mobile application designed to aid users in locating nearby pharmacies that carry a specific medication. The application enables customers to search for medicines and pharmacies within their local vicinity, view the pharmacy information.

Additionally, the search feature allows users to search for specific medications by name and locate nearby pharmacies that carry them. Pharmacies (Managers) can also update their inventory to ensure accurate information is available for customers. The development of **"Pharma Road"** is motivated by the limited availability of similar applications that provide comprehensive information on medication availability and location services to facilitate easy access to pharmacies.

By providing reliable information about medication availability, pricing, and contact information for local pharmacies, "**Pharma Road"** aims to enhance access to healthcare services and improve medication management for citizens.

ACKNOWLEDGEMENTS

First, we would like to thank Allah for the beginning and the end, and for putting us in love, striving and enduring in every period of our lives. Secondly, the family is the corner of the heart. Seeing them reassured us. Thank you for what we didn't see in us and for the encouragement we received from you. We would like to thank those who contributed to reaching this stage, the teachers, and their support for us all this time, especially our virtuous Dr. Nadera al-Jawabreh, who was the best guide for us. Friends of the first and last trip. In the end, we thank ourselves, because without praise, we would not have completed the road.

TABLE OF CONTENTS

List Of Figures

LIST OF TABLES

CHAPTER 1:INTRODUCTION

In this chapter we will introduce our project **“Pharma Road”**, so you have a better idea about the app, what motivated it and what problem it attempts to solve.

1.1 Overview

“Pharma Road” is a user-friendly mobile application designed to assist people in **Sweileh - Amman** to locate their required medication within local pharmacies. The primary objective of the application is to simplify the process of finding the right medicine and save users' valuable time and effort. Pharma Road boasts a straightforward interface, providing users with easily accessible information on the nearest pharmacies to their current location, including relevant contact information.

1.2 Project Motivation

Q1: What are the reasons behind your choice to develop this project?

“Pharma Road” developed to address the challenges faced by individuals who struggle to locate their required medication across multiple pharmacies and areas. The application's main objective is to offer an efficient and effective solution that caters to the needs of both local people and pharmacy managers. The application seeks to address primary issues faced by individuals who struggle to find medication due to time constraints, traffic, long distances, and insufficient information on medication availability. By providing users with a comprehensive resource for medication management, Pharma Road helps to simplify the process of searching for medication and streamline pharmacy operations by improving inventory management and sales.

“Pharma Road” offers a modern solution for individuals seeking a more efficient way to manage their medication needs. The application's user-friendly interface provides an easy-to-use digital platform for locating pharmacies with the required medication in stock. By simplifying the process of searching for medication and streamlining pharmacy operations, Pharma Road enhances access to healthcare services while saving time and effort for users. Overall, the application's integration of technology aligns with current trends and offers a convenient option for managing medication needs.

Q2. Why your project is important?

Because the process of finding the required medicine or specific pharmacy is an important matter that all people need, and we believe that people should have a quick way to obtain it and it will help maintain the health of patients, as some people are lazy to search for a medicine if they do not find it quickly, which will affect their health. As well as managing the pharmacy stock in a process that does not take minutes.

Q3. What is the new idea that has been proposed by this project?

"Pharma Road" introduces a new concept to the Jordanian market by utilizing technology to keep track of medication availability and presenting up-to-date information. The application has the potential to enhance the demand for online services in the country and improve the efficiency of pharmacies. By providing a more convenient option for managing medication needs, Pharma Road offers a unique opportunity to contribute to the development of online services in Jordan.

1.3 Problem Statement

In Jordan, the most common way to obtain medicines is through traditional means, which involves physically visiting multiple pharmacies to locate the required medication. This approach often results in several problems, including the following:

- Medication Availability: Some medications required by certain patients are not widely available, leading to wasted time and effort in searching for them in multiple locations.
- Anxiety and Tension: Individuals who require a specific medicine and cannot find it in the pharmacy often experience anxiety and tension. This can adversely affect their mental and physical health when offered alternatives by pharmacists.
- Limited Access: People in remote areas may encounter difficulties in finding pharmacies that have the required medication, making it challenging to receive proper medical care.

1.4 Project Aim and Objectives

Q1. What is the goal that this project wants to achieve?

The primary goal of the application is to enhance the accessibility and efficiency of obtaining medication, resulting in better health care for those in need. The application achieves this goal by offering various features, including:

- Helping users to easily locate nearby pharmacies that have the required medication available.
- Enabling users to search for medication by name and other features.
- Enabling users to search for pharmacy by name or neighborhood and other accurate information
- Allowing pharmacy owners to edit their list of available medication.
- Providing contact information and other relevant details for pharmacies.
- Improving the efficiency of pharmacies through better management of inventory and sales.

Q2. How can this project achieve this goal?

- Comprehensive database and updated: The application must contain a comprehensive database containing drug information, including brand and the database must be updated periodically to include new drugs.
- Increasing cooperation with the owners of pharmacies and the application and giving them the databases, they need in the pharmacy to create the application.
- Evaluation and review (Feedback): The service provided through the application can be evaluated and reviewed, and developers are provided with feedback and suggestions.

1.5 Project Limitations

1. The app only serves one region in the recent time.
2. The inability to find an alternative to the medicine.
3. Some advanced features are considering as future work.

1.6 Project Expected Output

At the end of our project, we expect that there will be a great demand for the application, as it is considered a new idea of its kind and rarely used in Jordan.

1.7 Project Schedule

2. Table (1): Project Schedule

Activity	Start Date	Finish Date
Introduction	22/3/2023	26/3/2023
Literature Review	27/3/2023	31/3/2023
Requirement Analysis	1/4//2023	10/4/2023
Architecture and Design	15/4/2023	29/4/2023
Implementation Plan	1/5/2023	20/12/2023
Dart Course	1/3/2023	1/4/2023
Flutter Course	2/4/2023	7/10/2023
Start implementation	25/9/2023	Recent
Database building	18/11/2023	1/1/2024
Firebase / back-end	18/11/2023	recent
Project testing and demo	1/3/2024	

2.1 Report Organization

The rest of the report is organized as follow:

- Chapter (2)introduces a discussion on reviewing some of the available literature related to the project.
- Chapter (3)lists the requirements analysis that involves requirements elicitation, non-functional user requirements, stockholders, and the use case diagram and workflow for each use case.
- Chapter (4)presents the System architecture, design, and screenshots of the application, all components, and the UML diagrams generally (software design) all it is described in this chapter.

- Chapter (5)will discuss the implementation plan and the programming language that we used to develop this project.
- Chapter (6)testing plan and test cases for some criteria.
- Chapter (7)presents future work and concludes the report.

CHAPTER 2: LITERATURE REVIEW

In this chapter, we will discuss previous applications and projects that have similar ideas to our project, the problems they cause, and the features that characterize our project.

2.1 Introduction

Medicine search apps are very useful for individuals who need to find medicines quickly, whether they are patients or other users.

As far as we know, there are few applications available in Jordan in this field. The following list includes what we found.

2.2 Existing Systems

Pharmacy & chemist finder

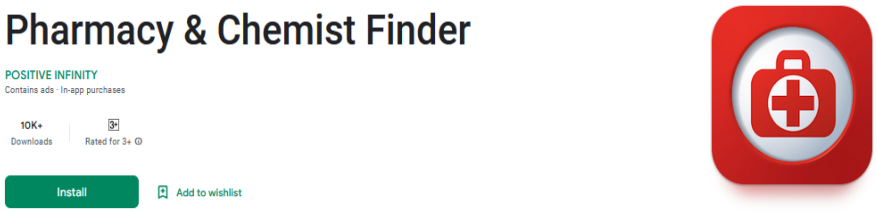
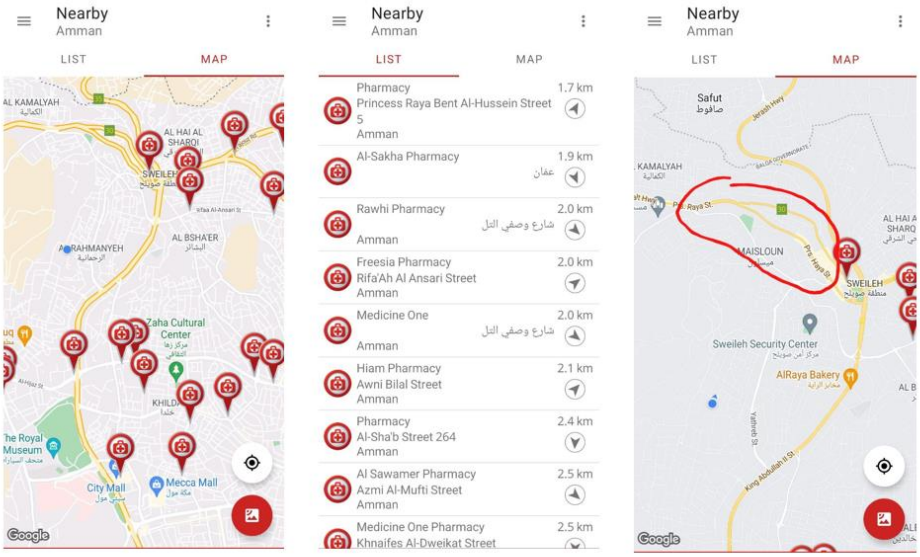


Figure 1: previous applications name.



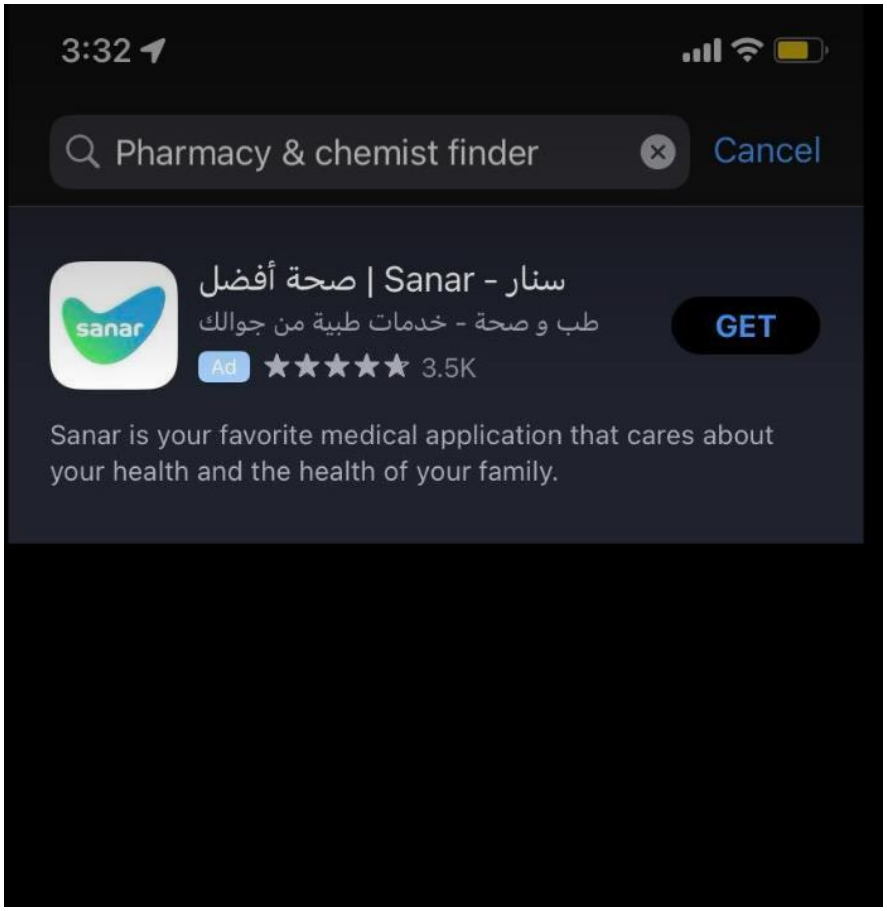


Figure 2: previous applications prototype.

Application overview:

Find pharmacies near your location and anywhere in the world. The application displays pharmacies in a list as well as on a map and allows you to easily navigate to an item with one click. This application also works abroad, use it all over the world.

2.3 Overall Problems of Existing Systems

1. The search feature is not available to the user for medicines, only the names and locations of pharmacies near to the current location appear.
2. May not be compatible with all systems. (Only available on **Android**).
3. After a range of distance, it's showing the location not names.
4. It may happen that the information available in the application is inaccurate because it is not updated regularly.

2.4 Overall Solution Approach

1. The feature of searching for the name of medicines will be added, showing the names and locations of the pharmacies available in them.
2. **Pharma road** would be available on both platforms iOS and android.
3. With the availability of the feature of adding medicines and pharmacies by the developers and the owner of the pharmacy, the information will be constantly

CHAPTER 3: REQUIREMENT ANALYSIS

This Chapter talks about requirement analysis and the definition of it and who is going to use this system and who are the stakeholders and what is the non-functional requirement of the system.

3.1 Stakeholders

In the context of any application, stakeholders are all individuals and entities that are directly or indirectly affected by the performance of the application.

It is important to identify the needs and concerns of all stakeholders when developing the application, as this can help ensure that the application meets the needs of all stakeholders and is successful.

Figure presents the primary and secondary stakeholders of **Pharma Road** application:

User: Individual who will be using the application searching for the required medicine.

Manager: The owner of the pharmacy.

Admin: The owner of the program/application who will develop and maintain the program, seeking to meet the needs of users and improve their experience in using the application.

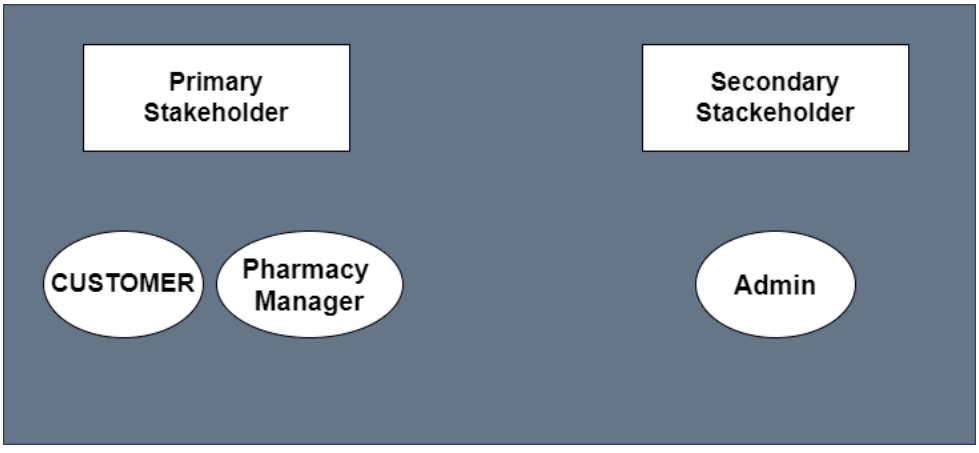


Figure 3: Primary and secondary stakeholders.

3.2 Use Case Diagram

Use case diagrams are considered for high-level requirement analysis of a system. When the requirements of a system are analyzed, the functionalities are captured in use cases.

The system functionalities are written in an organized manner. The second thing which is relevant to use cases is the actors. Actors can be defined as something that interacts with the system. Actors can be human users, some internal web sites. When we are planning to draw a use case diagram, we should have the following items identified:

→ Functionalities to be represented as a use case.

→ Actors

→ Relationships among the use cases and actors

The purpose of the use of a case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose.

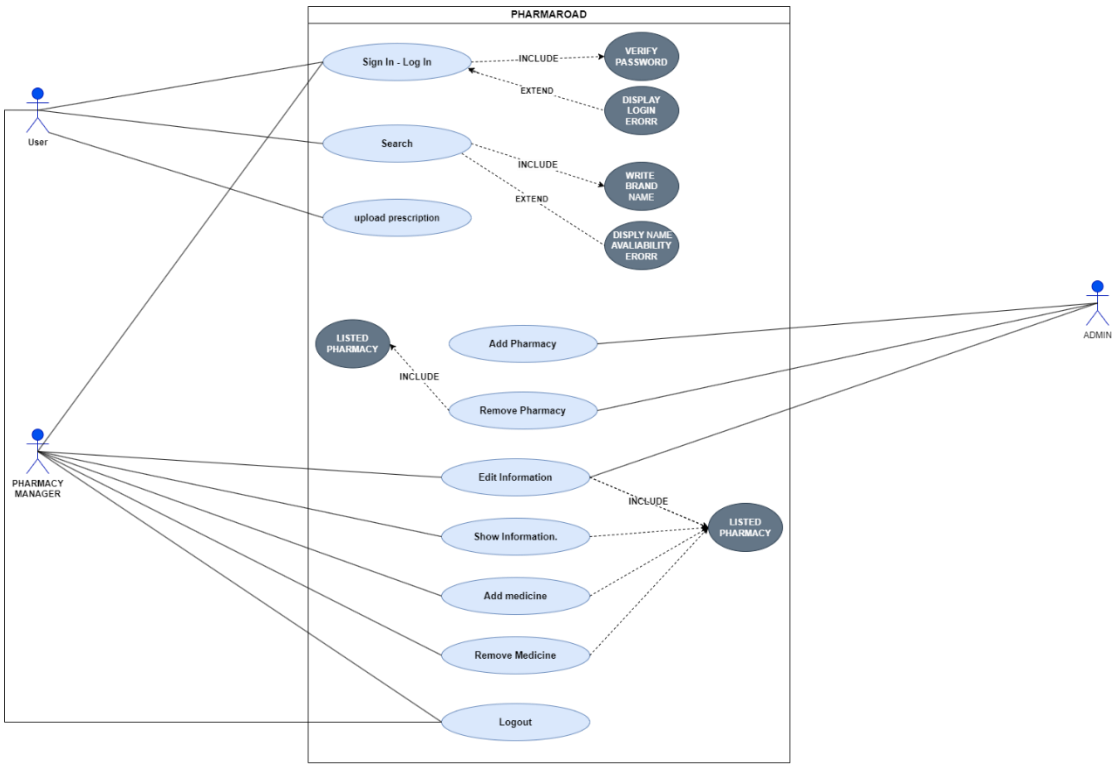


Figure 4: Use Case Diagram.

3.2.1 Use Case Section

Table (2): Normal Flow for each use case

Use case	Description	Action	Pre-condition	Post-condition
Sign in	Registration for a new user using needed information.	Choos the appropriate user type. As a customer enter the regular information as required and create new account	The information must be valid	
log in. **<<Include>>** **Verify password.** **<<Extend>>** **Display login Error**	Allow users to log-in to their accounts so they can use the application.	Users as customer enter their email and password. Users as pharmacy owners enter their ID given by the application owner in registration time and password.	Users must have a valid account	Application displays the relevant screen
Search **<<Include>>** **Write brand name.** **<<Extend>>** **display name availability error.**	The search helps users to find pharmacies that contain the required medicine.	To search for medicine, it is required to write the correct trade name of the medicine. If the writing is wrong, an error will be sent to you	you need to log in first	Users Choose one of appropriate results to see more information
Add pharmacy	Allows admins to add the new pharmacies that open in the area.	This feature facilitates the process of searching for the desired drug.		
Remove Pharmacy	If the pharmacy is closed permanently or temporarily, it will be deleted from the application so that it	Makes a filtered search, so the closed pharmacies do not appear in the search results		

	does not appear in the search process			
Edit information. << Include>> Approve edited info	Allows to the pharmacy manager to edit some of its information, like phone number or its name and medicines costs, these changes should be approved from the admins.	The modification will appear to users.	The pharmacy must have an ID with its manager/ owner	Display relevant information and keep it consistent.
Add medicine	Allows to add medicines by managers.	Only available to pharmacies managers	The pharmacy manager must have the pharmacy ID.	
Remove medicine	Allows to delete medicines from the pharmacy listed medicines if it is not available.	This feature is only available to pharmacy managers	The pharmacy manager must have the pharmacy ID	
Show information	Allows the admin to show all users and pharmacies information's. Also, managers can display pharmacy and medicines information.	Confirmation by the ID to allow only the managers/owners of pharmacy to reach private information not the other employees.		

3.3 Non-Functional User Requirements

1. **Usability:** Easy to use includes completeness, correctness, compatibility, and more critically user friendliness.

2. **Performance and scalability**: Able to serve large number of users without affecting system performance to response.

3. **Reliability:** Serve customers' orders with any system failures.

4. **Availability:** Should be available 24/7 and in lockdowns.

5. **Appearance:** To have an easy use for the application for the users.

6. **Maintainability:** To be updated according to the user's needs.

CHAPTER 4:ARCHITECTURE AND DESIGN

This chapter should include the description of the structure of the data used by the software UML diagrams. This chapter also should describe the main system parts and clarify the interaction between them.

4.1 Software (System) Architecture

4.1.1 Logical view

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the web site structure, and detailed modeling, translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interaction between classes.

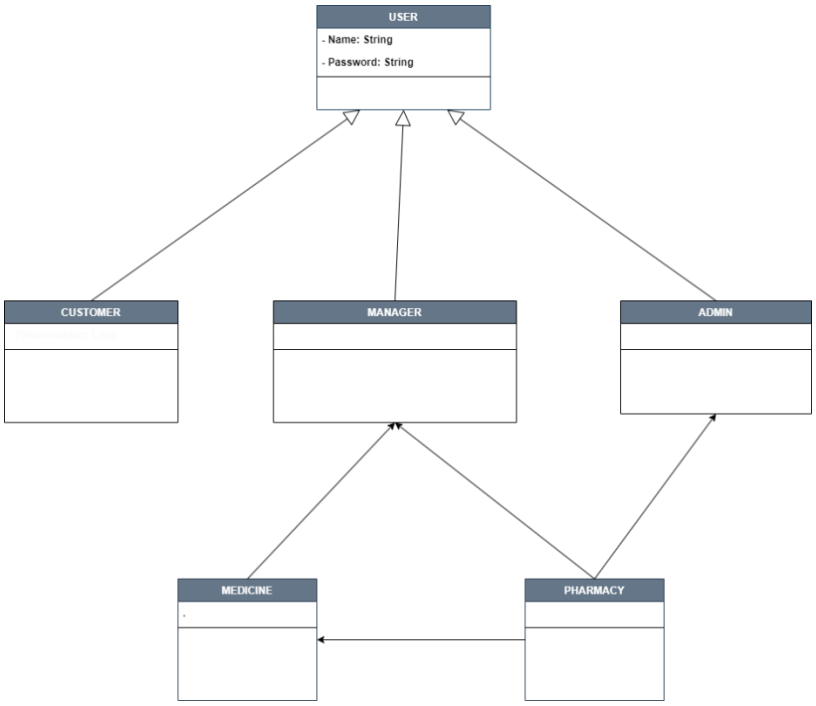


Figure 5: Logical view.

4.2 Software design

4.2.1 UML sequence/communication diagram

The UML sequence/communication diagram provides scenario and shows concurrent objects and the messages type.

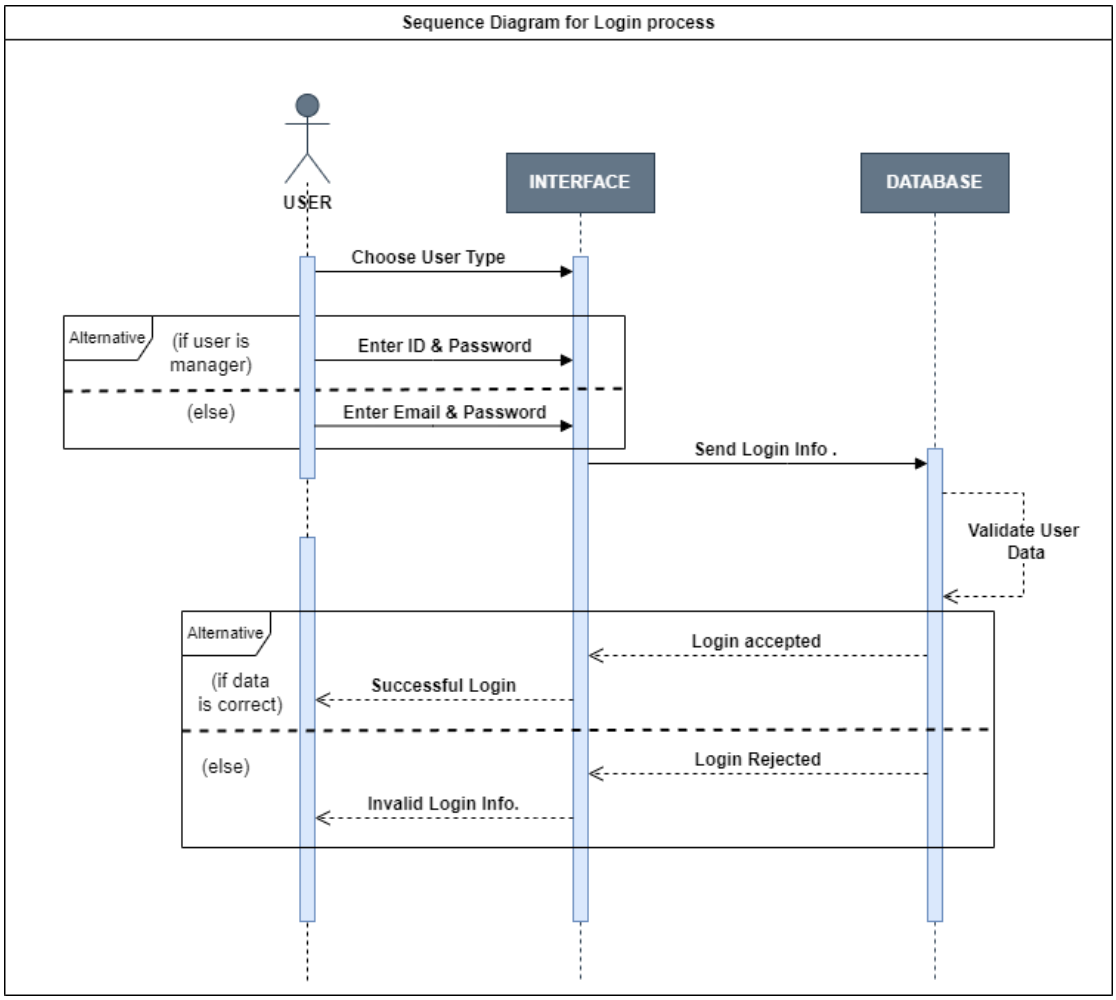


Figure 6: Sequence Diagram for log in.

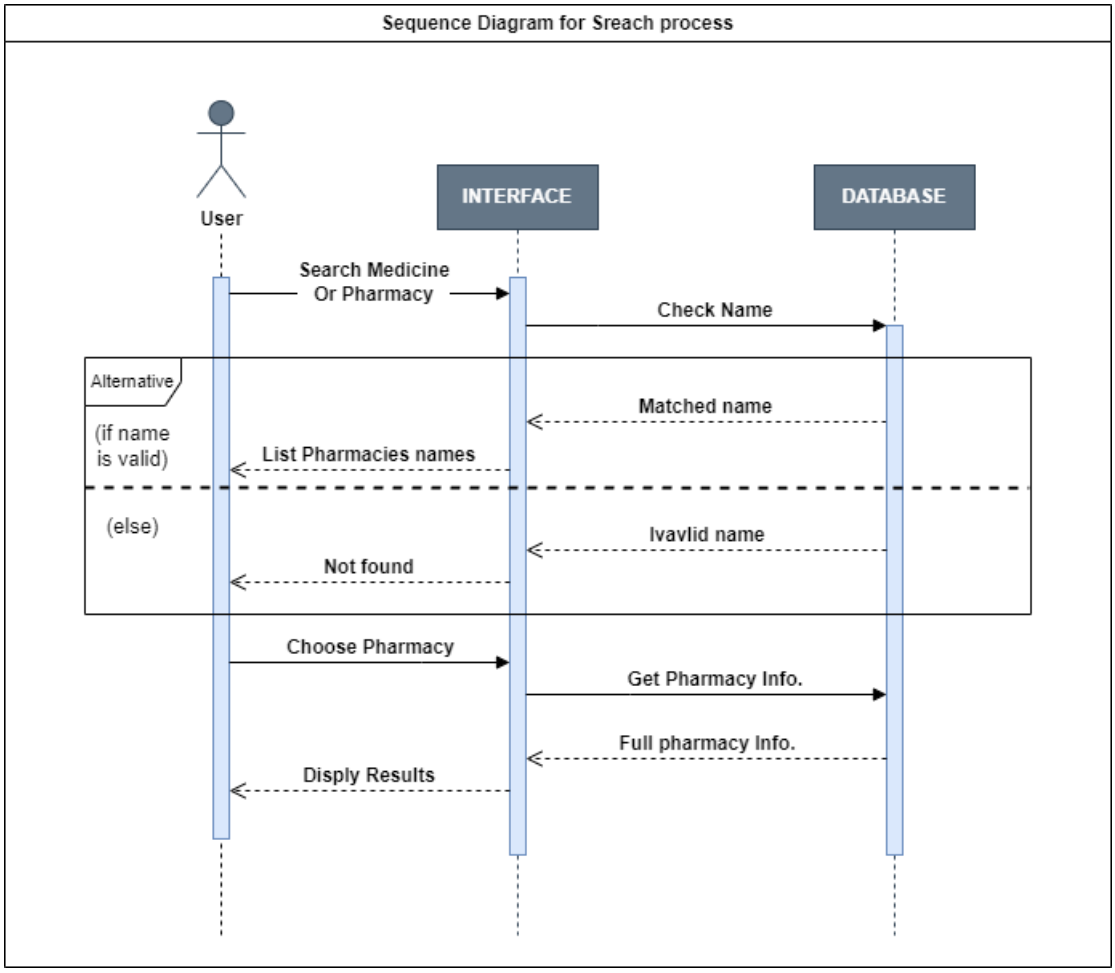


Figure 7: Sequence Diagram for search process.

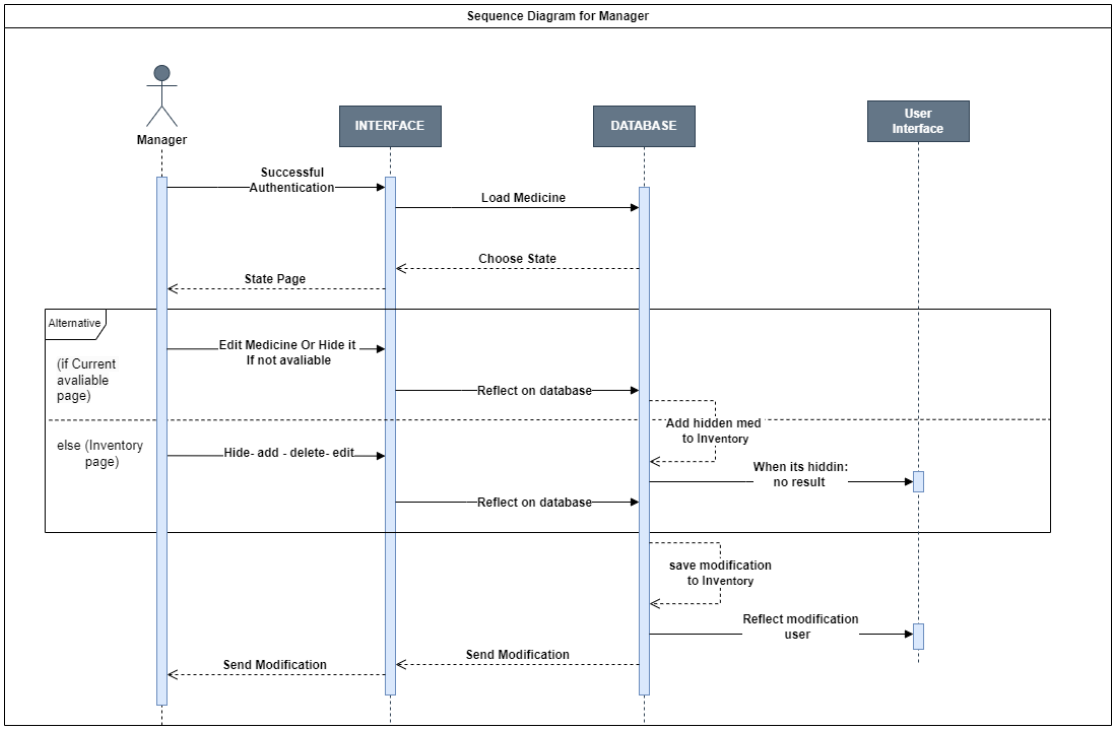


Figure 8: Sequence Diagram for pharmacy manager.

4.2.2 Class diagram

The class diagram shows classes' relationship, internal classes data, and methods. This should be based on the use case scenarios, problem description, and use case scenarios sequence/communication diagrams.

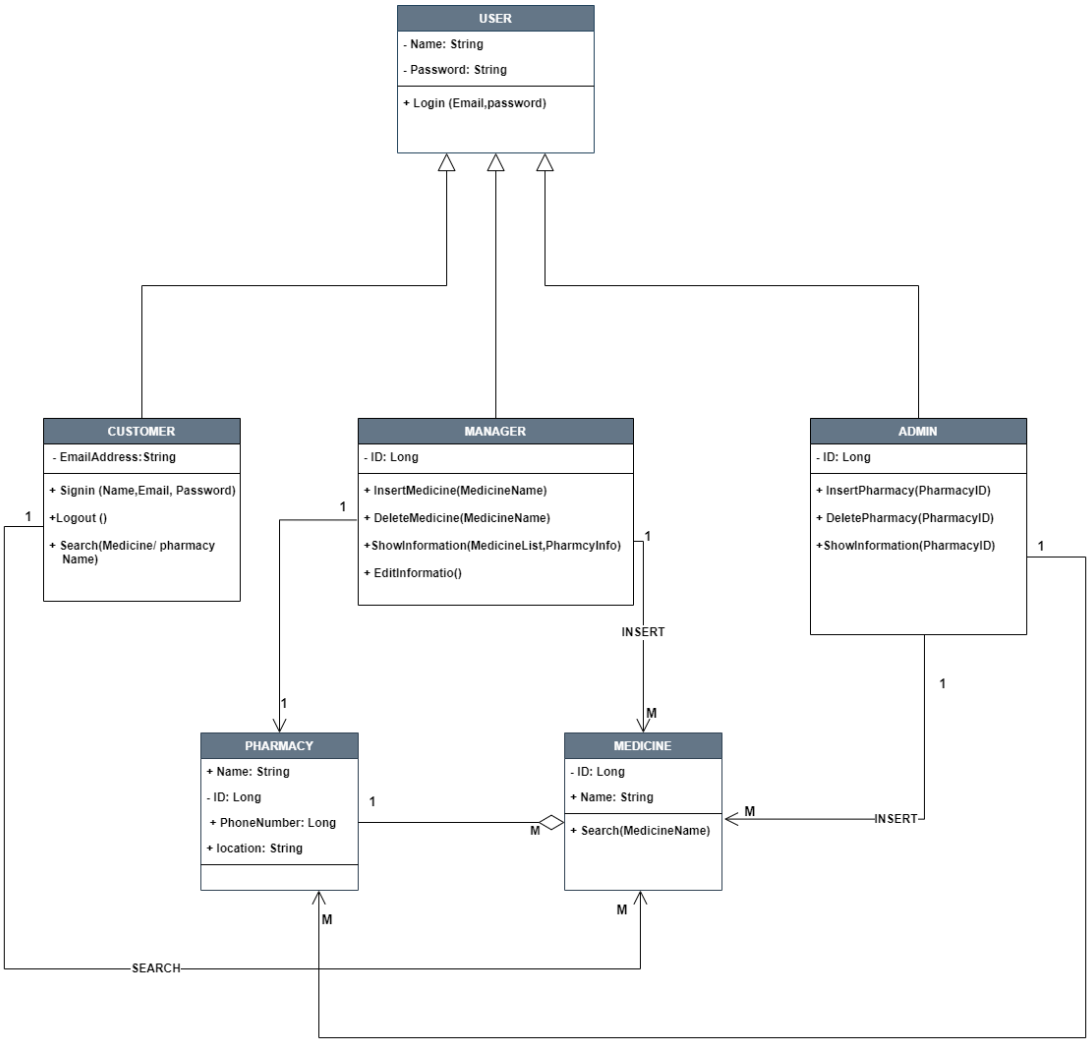


Figure 9: Class diagram.

4.2.3 ER diagram (if any)

The ER diagram provides the data structure in the database.

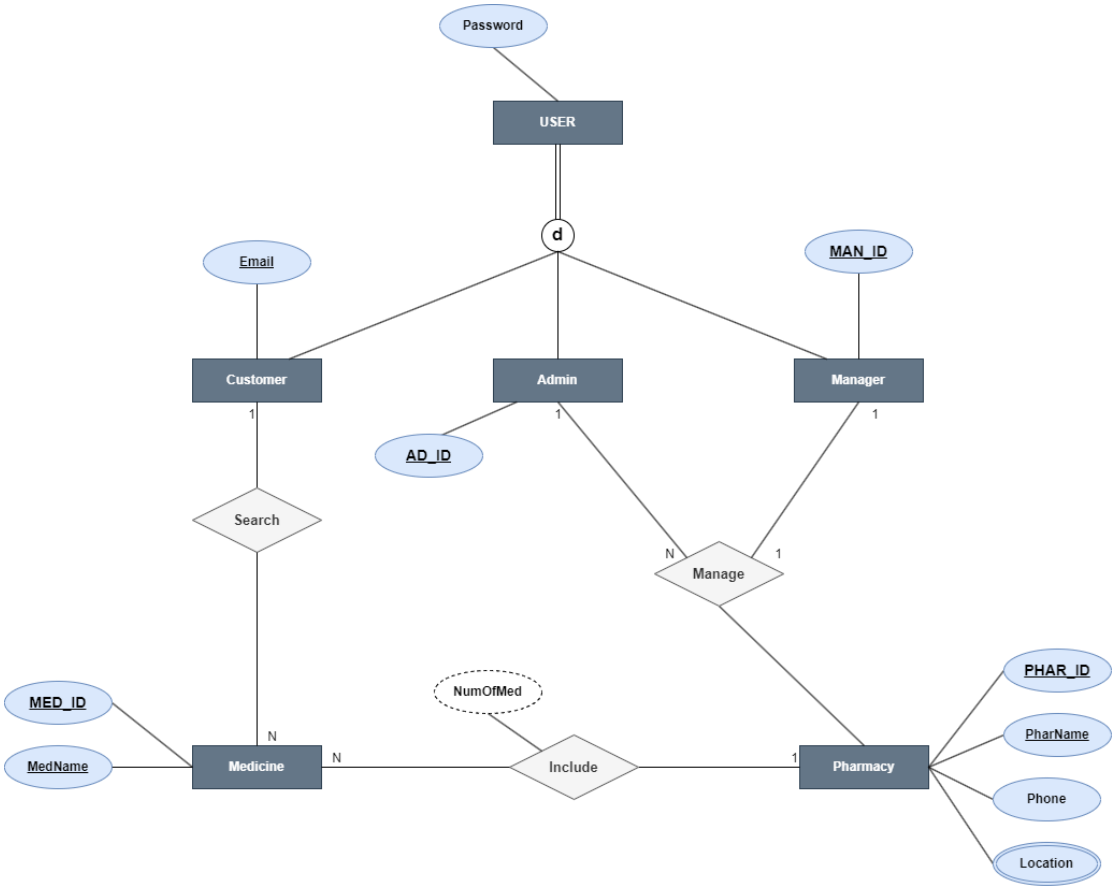


Figure 10: ER Diagram.

4.3 User interface design (prototype)

Figure 11: Application Icon.

`

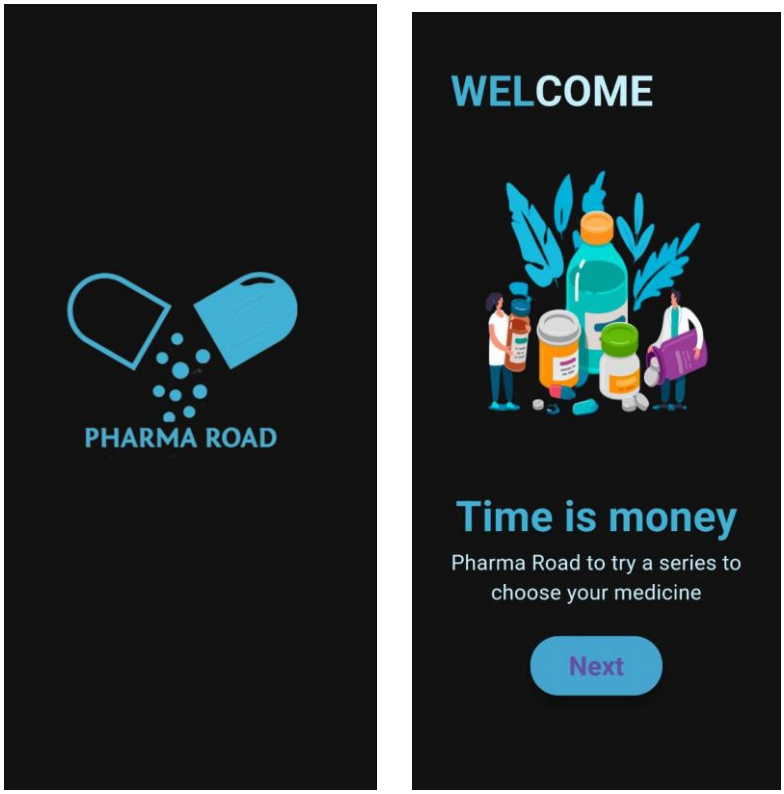


Figure 12: Splash and start Screen for the Application

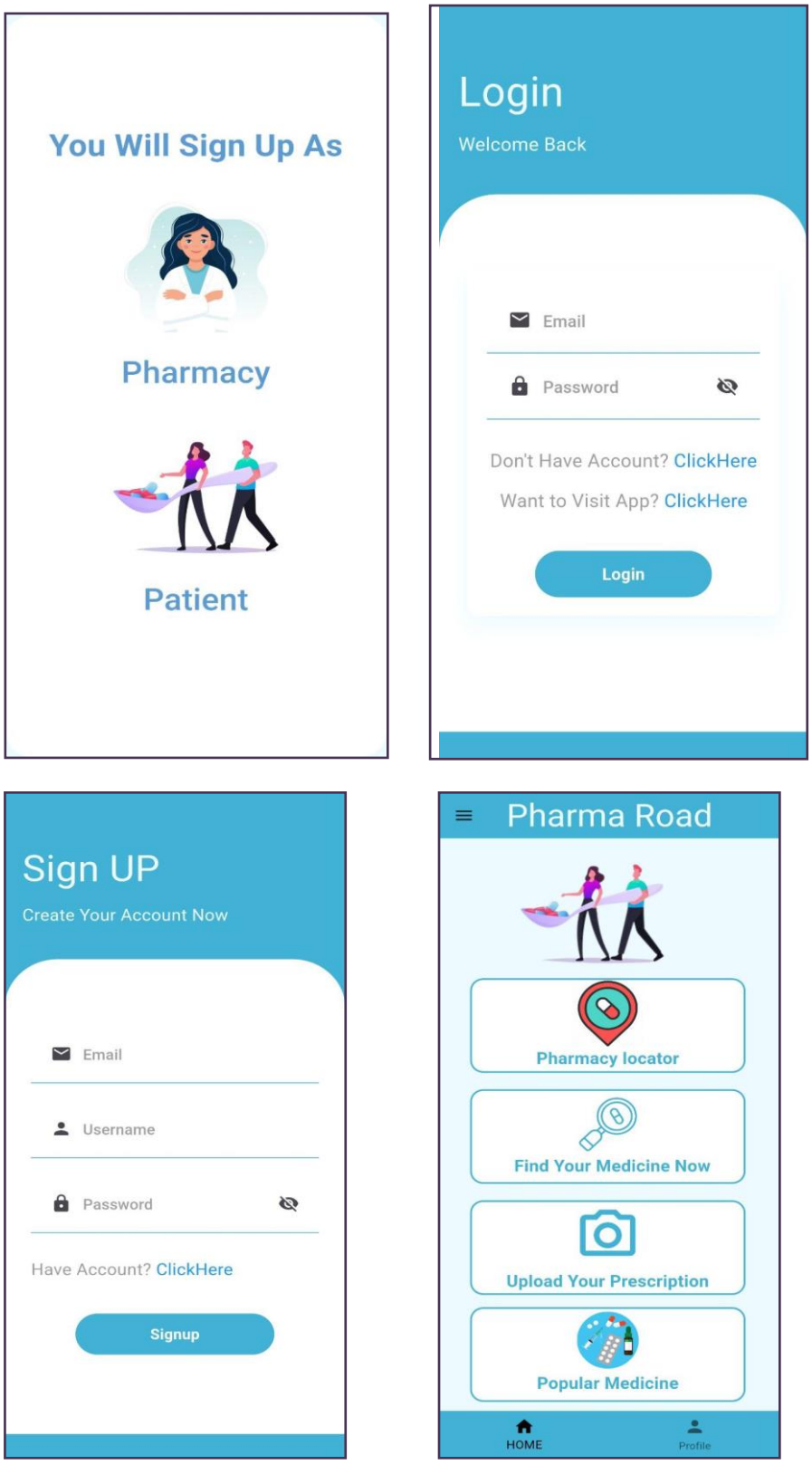


Figure 13: Users page, LogIn&Sign and Home Page for Patient.

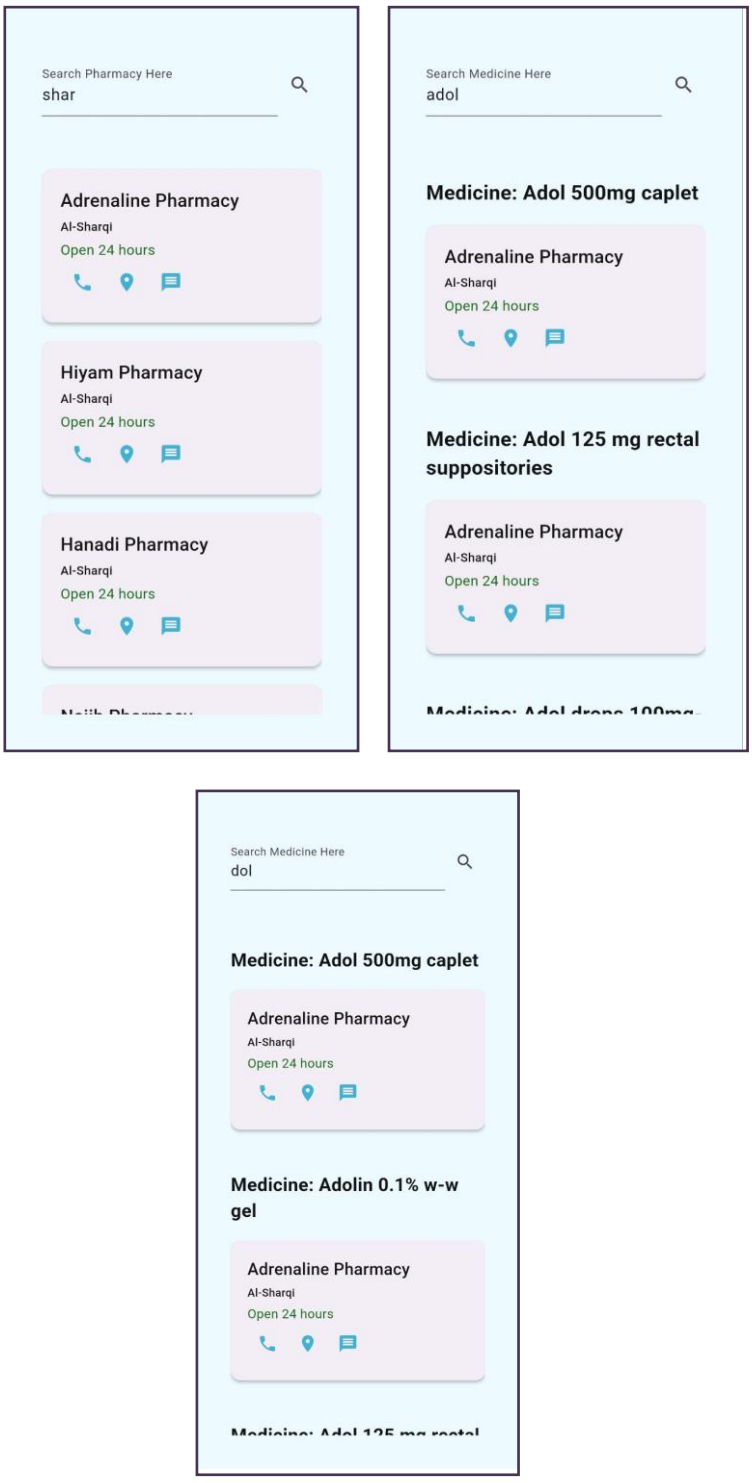


Figure 14: Search pages for medicine and pharmacy.

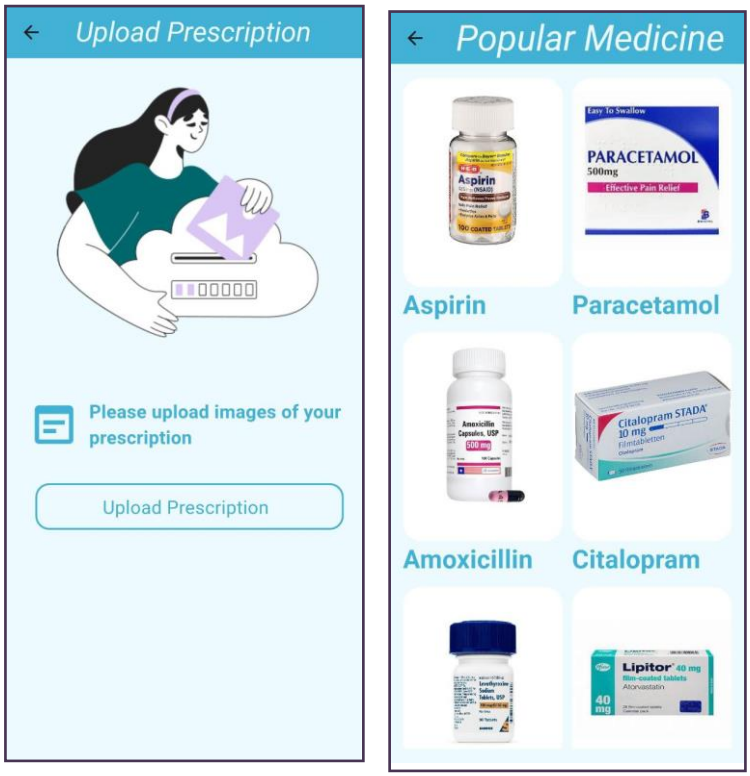


Figure 15: Upload Prescription and Popular Medicine Pages.

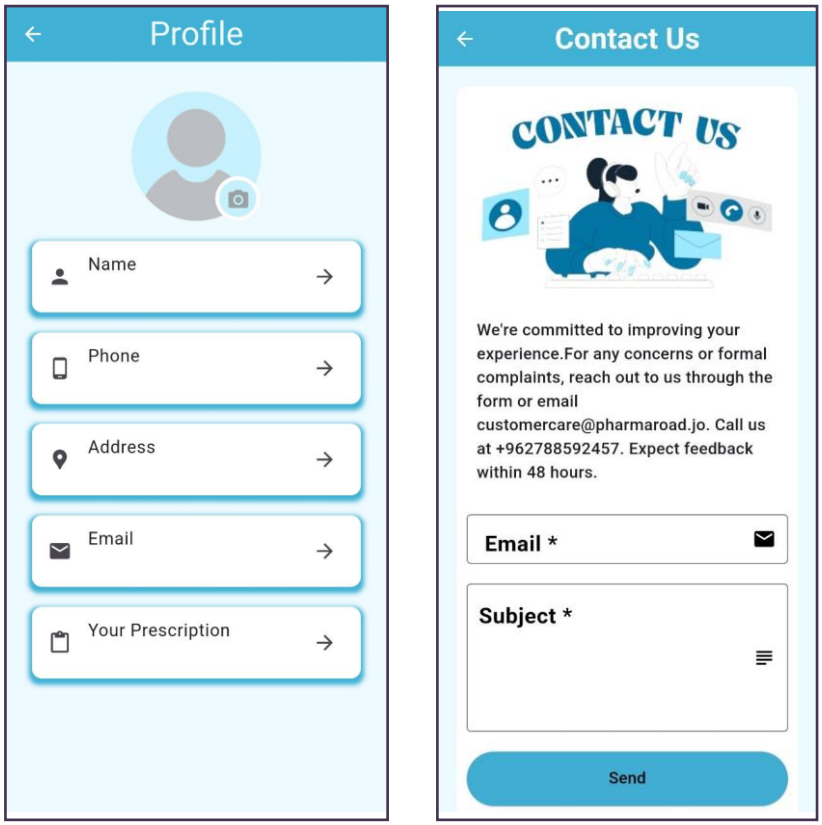


Figure 16: Patient Profile Page, Contact Us Page.

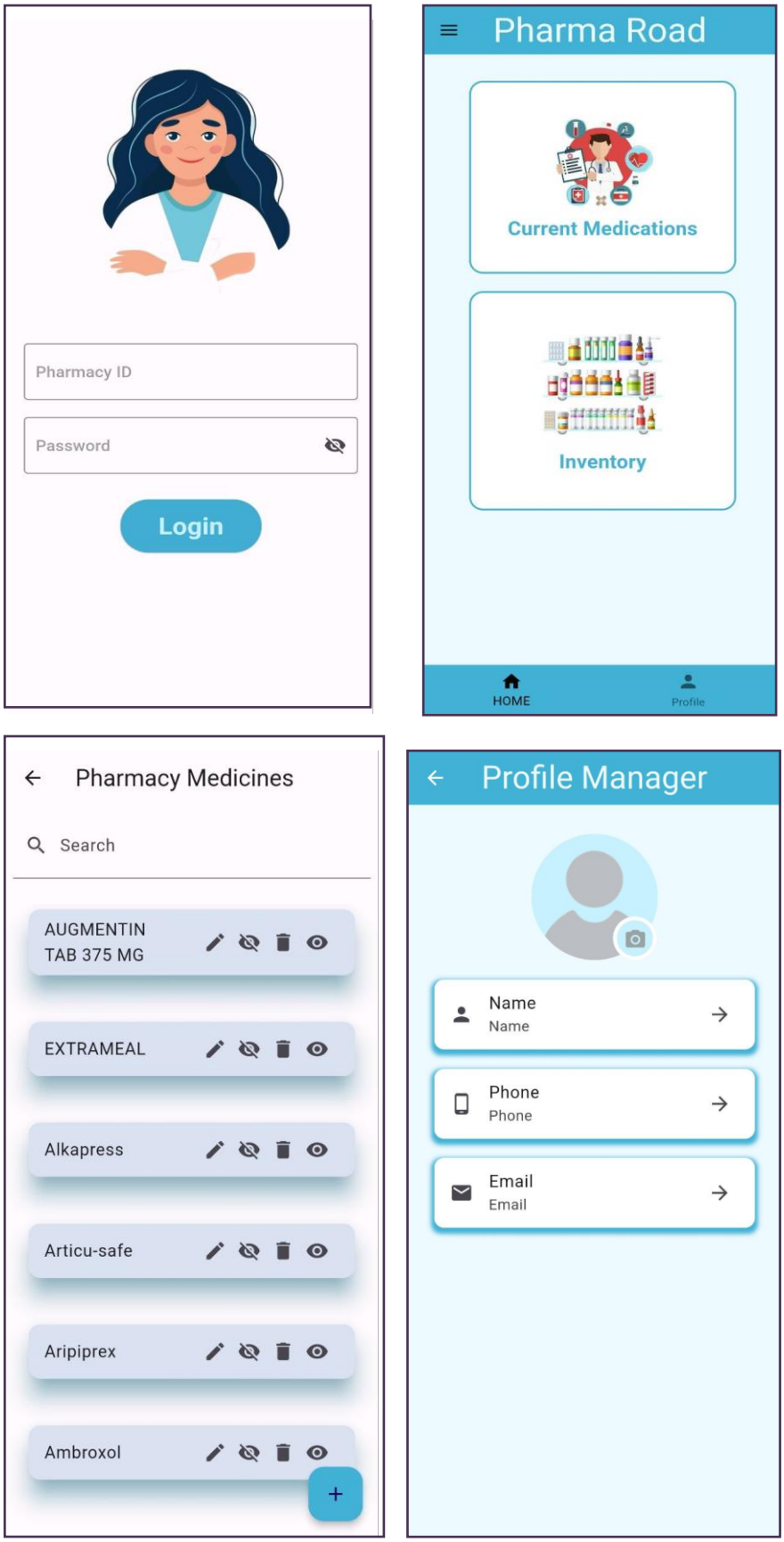


Figure 17: Pharmacy Manager Pages.

CHAPTER 5: IMPLEMENTATION PLAN

5.1 Description of Implementation

We have created two users **(Customers, Managers of pharmacy).**

Each of them has certain permissions in terms of the method of entry, where Manager of pharmacy can enter through the pharmacy id and password, as for customers, they are required to create a new account with the inclusion of the email, username and password When registering for the first time.

5.2 Programming language and technology

Flutter: This year, mobile applications continued to become more and more popular. Fortunately, there are many programming tools available to developers who want to create them. Among these tools there is Flutter, which has distinguished itself lately. Flutter is a free and open-source mobile UI framework created by Google and released in May 2017. In a few words, it allows you to create a native mobile application with only one codebase. This means that you can use one programming language and one codebase to create two different apps (for iOS and Android).

Dart: To develop with Flutter, you will use a programming language called Dart. The language was created by Google in October 2011, but it has improved a lot over these past years. Dart focuses on front-end development, and you can use it to create mobile and web applications.

Its goal is to offer the most productive programming language for multi-platform development, paired with a flexible execution runtime platform for app frameworks.

Fire Base: Firebase is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps, grow their user base. It is built on Google's infrastructure.

Firebase is categorized as a NoSQL database program, which stores data in JSON-like documents.

5.3 part of implementation if possible

```
Run | Debug | Profile
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  await FirebaseAppCheck.instance.activate();

  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
        debugShowCheckedModeBanner: false,
        home: FirebaseAuth.instance.currentUser == null
            ? const splach()
            : const drawer(),
        theme:
            ThemeData(primaryColor: const Color.fromRGBO(115, 144, 215, 0.86)),
        routes: {
          "Signup": (context) => const Signup(),
          "Users": (context) => const Users(),
          "Searchdrug": (context) => const SearchDrug(),
          "Welcome": (context) => const Welcome(),
```

```
class splach extends StatefulWidget {
  const splach({super.key});
  @override
  State<splach> createState() => _SplashScreenState();
}

class _SplashScreenState extends State<splach> {
  @override
  void initState() {
    FirebaseAuth.instance.authStateChanges().listen((User? user) {
      if (user == null) {
        print('User is currently signed out!');
      } else {
        print('User is signed in!');
      }
    });
    super.initState();
    Future.delayed(const Duration(seconds: 5), () {
      Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (context) => const Welcome()),
      );
    }); // Future.delayed
  }
```

Figure 18: main dart section.

```dart
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:email_validator/email_validator.dart';

class AuthService {
  final FirebaseAuth _auth = FirebaseAuth.instance;

  Future<User?> signInUser(String email, String password) async {
    try {
      UserCredential credential = await _auth.signInWithEmailAndPassword(
        email: email,
        password: password,
      );
      return credential.user;
    } on FirebaseAuthException catch (e) {
      // Handle authentication exceptions
      print("Authentication error: ${e.message}");
      return null;
    }
  }
}

class Log extends StatefulWidget {
  const Log({Key? key}) : super(key: key);

  @override
  _LogState createState() => _LogState();
}

class _LogState extends State<Log> {
  var obscureText = true;
  TextEditingController email = TextEditingController();
  TextEditingController password = TextEditingController();
  final _formKey = GlobalKey<FormState>();
  final AuthService _authService = AuthService();

  // Function to trim input strings
  String _trimInput(String input) {
    return input.trim();
  }
```

```dart
        ), // Border
      ), // BoxDecoration
      child: TextFormField(
        controller: email,
        validator: (value) {
          if (value == null || _trimInput(value).isEmpty) {
            return 'Please enter your email';
          } else if (!EmailValidator.validate(_trimInput(value))) {
            return 'Please enter a valid email';
          }
          return null;
        },
        decoration: const InputDecoration(
          prefixIcon: Icon(Icons.email),
          hintText: "Email",
          hintStyle: TextStyle(color: Colors.grey),
          border: InputBorder.none,
        ), // InputDecoration
      ), // TextFormField
    ), // Container
    Container(
      padding: const EdgeInsets.all(10),
      decoration: const BoxDecoration(
        border: Border(
          bottom: BorderSide(color: Color(0xff41b2d6)),
        ), // Border
      ), // BoxDecoration
      child: TextFormField(
        controller: password,
        validator: (value) {
          if (value == null || _trimInput(value).isEmpty) {
            return 'Please enter your password';
          }
          return null;
        },
        obscureText: obscureText,
        decoration: InputDecoration(
          prefixIcon: const Icon(Icons.lock),
          suffixIcon: GestureDetector(
            onTap: () {
              setState(() {
                obscureText = !obscureText;
              });
```

Figure 19: login for User.

```dart
Widget buildSignupButton() {
  return Container(
    height: 50,
    margin: const EdgeInsets.symmetric(horizontal: 50),
    decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(50),
      color: const Color(0xff41b2d6),
    ), // BoxDecoration
    child: Center(
      child: TextButton(
        onPressed: () async {
          email.text = email.text.trim();
          name.text = name.text.trim();
          password.text = password.text.trim();

          if (email.text.isEmpty || name.text.isEmpty || password.text.isEmpty) {
            showSnackBar('Please fill in all fields.');
          } else if (!EmailValidator.validate(email.text)) {
            showSnackBar('Please enter a valid email address.');
          } else {
            try {
              await FirebaseAuth.instance.createUserWithEmailAndPassword(
                email: email.text,
                password: password.text,
              );
              Navigator.of(context).pushReplacementNamed("drawer");
            } on FirebaseAuthException catch (e) {
              if (e.code == 'email-already-in-use') {
                showSnackBar('The account already exists for that email.');
              } else if (e.code == 'weak-password') {
                showSnackBar('The password provided is too weak.');
              }
            } catch (e) {
              print(e);
            }
          }
        }
```

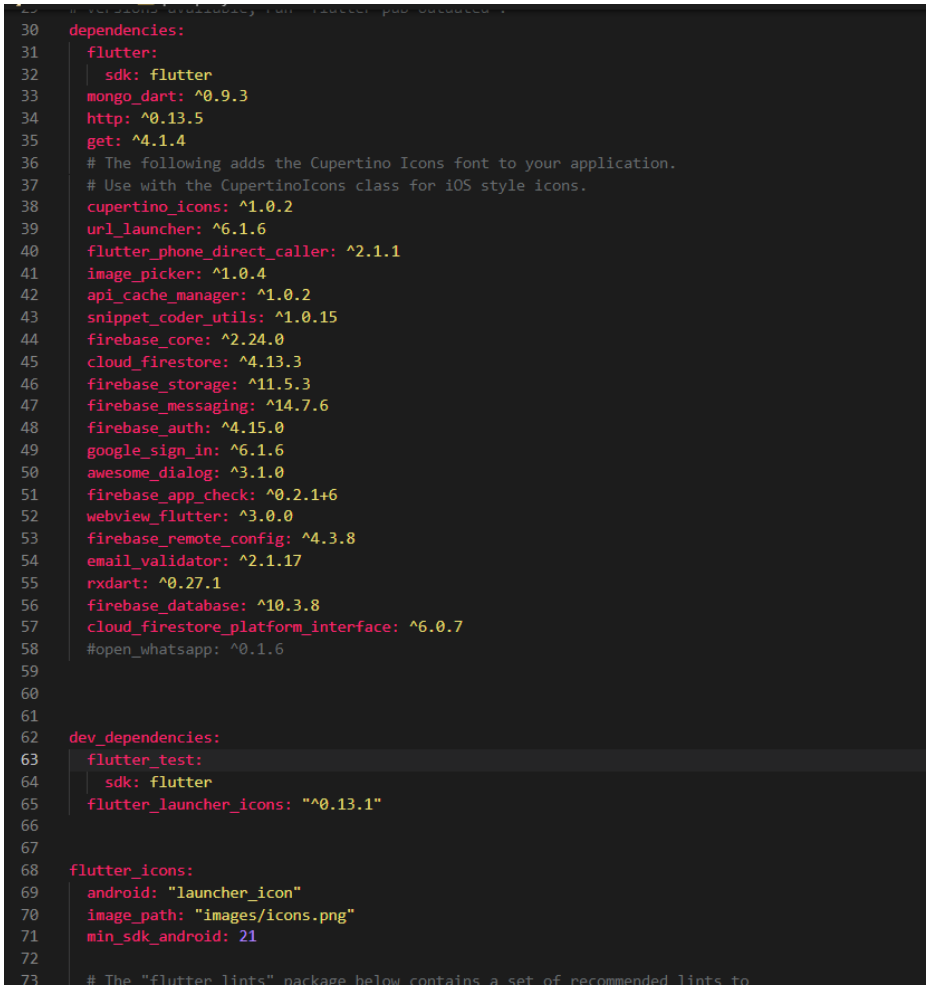
Figure 20: signup for User.

```dart
class SearchDrug extends StatefulWidget {
  const SearchDrug({Key? key}) : super(key: key);

  @override
  State<SearchDrug> createState() => _SearchDrugState();
}

class _SearchDrugState extends State<SearchDrug> {
  TextEditingController searchController = TextEditingController();
  Map<String, List<Map<String, dynamic>>> pharmaciesByMedicine = {};
  final StreamController<String> _searchController = StreamController<String>();
  bool isLoading = false;

  CollectionReference pharmacy = FirebaseFirestore.instance.collection("Pharmacies");

  @override
  void initState() {
    super.initState();
    _searchController.stream.listen((searchTerm) {
      retrievePharmacyData(searchTerm);

      print(retrievePharmacyData);
    });
  }

 void retrievePharmacyData(String searchTerm) async {
  print('Searching for: $searchTerm');
  setState(() {
    isLoading = true;
  });

  pharmaciesByMedicine.clear();
  setState(() {});

  searchTerm = searchTerm.toLowerCase();

  if (searchTerm.isEmpty) {
    setState(() {
      isLoading = false;
    });
```

```dart
  QuerySnapshot pharmaciesSnapshot = await pharmacy.get();

  for (QueryDocumentSnapshot pharmacyDoc in pharmaciesSnapshot.docs) {
    String pharmacyId = pharmacyDoc.id;

    if (pharmacyDoc.data() is Map<String, dynamic>) {
      Map<String, dynamic> pharmacyData = pharmacyDoc.data() as Map<String, dynamic>;

      if (pharmacyData.containsKey('name') &&
          pharmacyData.containsKey('location') &&
          pharmacyData.containsKey('neighborhood') &&
          pharmacyData.containsKey('phone')) {
        String pharmacyName = pharmacyData['name'];
        String pharmacyLocation = pharmacyData['location'];
        String pharmacyNeighborhood = pharmacyData['neighborhood'];
        String pharmacyPhone = pharmacyData['phone'];

        // Fetch medicines for the pharmacy
        QuerySnapshot medicinesSnapshot = await FirebaseFirestore.instance
            .collection("Pharmacies")
            .doc(pharmacyId)
            .collection("medicine")
            .where('isVisible', isEqualTo: true)
            .get();
```

```dart
        QuerySnapshot medicinesSnapshot = await FirebaseFirestore.instance
            .collection("Pharmacies")
            .doc(pharmacyId)
            .collection("medicine")
            .where('isVisible', isEqualTo: true)
            .get();

        for (QueryDocumentSnapshot medicineDoc in medicinesSnapshot.docs) {
          if (medicineDoc.data() is Map<String, dynamic>) {
            Map<String, dynamic> medicineData = medicineDoc.data() as Map<String, dynamic>;

            if (medicineData.containsKey('Mname')) {
              String medicineName = medicineData['Mname'].toLowerCase();

              // Check if the medicine contains the search term
              if (medicineName.contains(searchTerm)) {
                Map<String, dynamic> pharmacyInfo = {
                  'id': pharmacyId,
                  'name': pharmacyName,
                  'location': pharmacyLocation,
                  'neighborhood': pharmacyNeighborhood,
                  'phone': pharmacyPhone,

                };

                // Add the pharmacy to the result for the specific medicine
                if (!pharmaciesByMedicine.containsKey(medicineName)) {
                  pharmaciesByMedicine[medicineName] = [];
                }

                // Check if the pharmacy is not already added for this medicine
                if (!pharmaciesByMedicine[medicineName]!
                    .any((pharmacy) => pharmacy['id'] == pharmacyId)) {
                  pharmaciesByMedicine[medicineName]!.add(pharmacyInfo);
                }
              }
            }
```

Figure 21:search function.



```
dependencies:
  flutter:
    sdk: flutter
  mongo_dart: ^0.9.3
  http: ^0.13.5
  get: ^4.1.4
  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.2
  url_launcher: ^6.1.6
  flutter_phone_direct_caller: ^2.1.1
  image_picker: ^1.0.4
  api_cache_manager: ^1.0.2
  snippet_coder_utils: ^1.0.15
  firebase_core: ^2.24.0
  cloud_firestore: ^4.13.3
  firebase_storage: ^11.5.3
  firebase_messaging: ^14.7.6
  firebase_auth: ^4.15.0
  google_sign_in: ^6.1.6
  awesome_dialog: ^3.1.0
  firebase_app_check: ^0.2.1+6
  webview_flutter: ^3.0.0
  firebase_remote_config: ^4.3.8
  email_validator: ^2.1.17
  rxdart: ^0.27.1
  firebase_database: ^10.3.8
  cloud_firestore_platform_interface: ^6.0.7
  #open_whatsapp: ^0.1.6



dev_dependencies:
  flutter_test:
    sdk: flutter
  flutter_launcher_icons: "^0.13.1"


flutter_icons:
  android: "launcher_icon"
  image_path: "images/icons.png"
  min_sdk_android: 21

  # The "flutter_lints" package below contains a set of recommended lints to
```

Figure 22: Plugins Used for the Application

CHAPTER 6: TESTING PLAN

The test plan in this chapter includes testing for both user and pharmacy manager authentication.

6.1 Testing

6.1.1 Authentication Testing

Table (3): Authentication Testing

Function // Case	Wrong/weak password	Wrong Email/id	Empty Feilds
User login	`Please enter a valid password`	`Invalid email, enter valid one.`	`Please enter your email` `Please enter your password`
User Sign up	`Please enter strong password`	`Email already exists.`	`Please enter your email` `Please enter your password`
Manager Login	`Wrong password`	`Invalid ID, enter valid one.`	`Please enter your ID` `Please enter your password`

CHAPTER 7: CONCLUSION AND RESULTS

The conclusion is a required part that closes the document with a brief summary of the study including the problems found and the proposed solution. Most importantly, it should recommend to the readers the benefits of pursuing the project based on the researcher's analysis.

7.1 Summary of accomplished project

Conclusion:

In conclusion, "Pharma Road" is a location-based mobile application designed to simplify the process of finding and obtaining medication. The application addresses the challenges faced by individuals in locating their required medication by providing a user-friendly interface that allows users to search for medicines within their local vicinity, view medication prices, and purchase medication online. By offering comprehensive information on medication availability, pricing, and contact information for local pharmacies, "Pharma Road" aims to enhance access to healthcare services and improve medication management for citizens.

Results:

The development of "Pharma Road" offers several key results and benefits. Firstly, it provides a modern and efficient solution for individuals who struggle to find their required medication across multiple pharmacies and areas. By utilizing technology, the application helps users save valuable time and effort by quickly locating nearby pharmacies that carry the specific medication they need. This not only simplifies the process for users but also contributes to improving inventory management and sales for pharmacies.

Moreover, "Pharma Road" introduces a new concept to the Jordanian market by utilizing technology to track medication availability and provide up-to-date information. This innovation has the potential to enhance the demand for online services in the country and improve the efficiency of pharmacies. The user-friendly interface of the application aligns with current trends and offers a convenient option for managing medication needs.

While the project is currently limited to serving the Sweileh - Amman area , it is expected that there will be a great demand for the application due to its unique concept and rarity in Jordan. However, one limitation of the project is the inability to find alternatives to the required medicine, which could be a potential area for future development.

Overall, the "Pharma Road" application aims to enhance accessibility and efficiency in obtaining medication, contributing to better healthcare and medication management for individuals in need.

7.2 Future Work

The "**Pharma Road**" application has successfully addressed the challenges faced by individuals in locating and obtaining their required medication. However, there is still ample room for future work and enhancements to further improve the application's functionality and expand its impact. The following section discusses potential areas of focus for future development.

1. Add buying, selling, messaging features, to directly purchase medication and other functions.
2. Expansion to More Regions.
3. Integration of Alternative Medicine Options.
4. Deployment on (iOS) platform.