Yarmouk University

Faculty of Information Technology and Computer Science

Computer Science Department

Project Title: **Help Me!**
Project Supervisor: **Dr Bilal Abed Alguni**
Project Team: **1. Haitham Nezam Mahmoud Al-Omari** **ID:2015801020** **2. Hazem Zeiad Ahmad Al-Qasem** **ID:2015801035** **3. Bader Sami Shawkat Fayiz** **ID:2015901027**

2018-2019 Second Semester

Table of Contents

Table of Figures

Table of Use Case

Abstract

We have implemented this project to facilitate the process of finding a specific service needed by the community, these services fall under the perspective of professional (manual) services, “Help Me“ is a service application that is based on finding the service that is needed by the community by putting the professionals in their services on this application and presenting it to the users to save them time and effort in finding and utilizing their services. After completion of this project, this project will solve some of the problems of unemployment and will solve these problems in a smooth and easy to use manner.

1. Introduction and Overview

In this section, we briefly describe the current system and the motivation of our work It also includes an overview on the rest of the sections included in this documentation.

1.1 Introduction:

At present, Information communication of technology (ICT) plays an important role in all aspects of life. We have proposed the idea of this project to facilitate the community to find the services they need easily, quickly and in a timely manner and find the right people to complete this service.

Help Me application will find the right person easily according to the criteria of evaluation will be discussed later in detail and also this application solves the problem of unemployment in the sense of professionals (manual) to find work easily also depends on them in the extent of craftsmanship in completion the service.

Help Me application serves all aspects of society (individuals, companies, institutions, hospitals, etc...), we can take advantage of the technological revolution by making it easier for all sectors of the society to find the services they need such as (blacksmith, carpenter, electrician, etc...).

We have recently noticed a problem for people looking for a particular service, which is that it may be very difficult to find the services and people who are most suitable for them, where they need to search in many offices that help to provide these services.

Reasons for this problem It is difficult to find handicraft owners who have the appropriate skill to complete the service professionally, which is not available in all manual professions so we have proposed the idea of this project to solve this problem.

In order to solve this problem we will create an application for smart phones by the professionals to publish their services on this application and their sites and their telephone numbers and explain the services they provide in turn users of this application to search for the service they need and communicate with the right professions and knowledge of their sites through virtual maps and Identify the shortest path between the service provider and the users of the service to complete the communication process quickly and easily and thus we solve this problem using the best and most convenient way for the service provider and the beneficiary of this service.

1.2 Contribution and Objectives:

Help Me application contributes to solving the problem of unemployment by employing the unemployed by putting the services they have and offering them to users in terms of benefiting from them easily.

The main objectives are:

- **Productivity:** Increase productivity and efficiency by providing employment opportunities for people, and organizing them, providing all means of assistance to them.

- **Computing:** Make these practical services available in a computerized manner on smart phones, so that the person does not need long time to find his service.

- **Organization and evaluation:** The application aims to evaluate the available services, in terms of the speed performance and efficiency of the working people, to help users to choose the best service that meets all their needs.

2. Background and Literature review

There are other applications similar to the idea of our application, but these applications were not enough to implement our project and we added other things we will address later.

We have found some applications that are very similar to our idea:

2.1 Open Sooq:

OpenSooq Application operates as an online classified that offers its users the ability to buy and sell products and services online through online classifieds postings. It provides the users to post free classified advertisements and also to browse advertised goods, products, and services using searches and filters to find deals and discounts. The company is headquartered in Amman, Jordan. **[5]**

2.2 Aoun| House Maintenance Service:

Aoun providers go through a selection criterion to maintain Aoun standards, all providers are professional and trained to deliver the best services.

Book the service you need at any time you want, schedule your orders or ask for them urgently, Aoun is always there whenever you need help.

Aoun allows you to choose from a variety of service providers, based on their experience, profile, ratings and reviews! **[6]**

2.3 Thumbtack: Find Local Pros:

The Thumbtack app connects you with local professionals for pretty much any project on your to-do list – contractors, cleaning services, movers, dog walkers, home repair help, wedding photographers, you name it. Get bids, read reviews, message the pros and hire your favorite right from the app. **[7]**

2.4 JustServe:

JustServe app gives you the main features of JustServe.org to help you find volunteer opportunities while on the go. It can link into your phone's location services, calendar, and map apps to make volunteering in the community easier than ever.
- Search for opportunities based on your account location, your current location, or by an inputted location
- Volunteer for opportunities
- Easily add opportunities and locations to your mobile device's calendar and maps apps
- Read success stories from past volunteering projects
- Just Serve! **[8]**

3. System Analysis

In this section, we illustrate:

1. System User.
2. Functional Requirements.
3. Non-Functional Requirements.
4. Use Case diagram.

3.1 System User:

1. Admin, who manages the system.

2. We will have several users use this system:

a. A user to provide his services.

b. A user who reviews and selects the services.

3.2 Functional Requirements:

- **Authentication (Login and Sign up):** The application shall allow the users to create an account by recording the required data such as name, e-mail, password or other user-generated accounts such as Facebook, Gmail directly without the need to enter data in the application.

- **Personal account:** The application shall allow the users to create a personal account for each user through which the user can modify his personal data such as (name, password, etc.) or exit from his personal account.

- **Add services:** The application shall allow the users to add their professional services (manual) accurately and detailed through his personal account.

- **View services:** The application shall allow the users to browse and use the services provided by other users (manual professionals).

- **Search process:** The application shall allow the users to searching for a particular service to search for it and find it easily and quickly by typing the user the name of the service they need and the nearby area (Search bar).

- **Blocking and service availability:** The application shall allow the users who have offered their services to other users to block the service for a specific time specified by the service provider or to grant it at any time and through which the service applicant can know if the services are blocked or available.

- **Ways to communicate with service providers:** The application shall allow the users who want to benefit from the services offered to communicate with the service providers in several ways:

  1. Send massage to publisher of the service with Location using (Google Map API).

  2. Email of the service provider.

  3. The phone number of the service provider.

- **Evaluation of the services offered:** The application shall allow the users who have benefited from the services offered to evaluate the performance of the service providers and thus achieve the objectives of the goals:

  1. Increase the competitiveness between the service providers.

  2. Choosing the best providers by the user.

  3. Sort the services by rating.

- **Send notifications to users:** The administrators of this application send alerts informing users of this application of the latest developments that can occur.

3.3 Non-Functional Requirements:

1. Security: The system shall not allow direct access to the database for any user.

2. The system's code shall be documented and clear in description.

3. The system shall be easy to manage.

4. The system shall be available 24 hours.

5. Easy to use: the system shall have friendly interface for users.

6. Reliability: Ability of a computer program to perform its intended functions and operations in a system's environment, without experiencing failure (system crash).

7. Performance: Provider systems MUST meet the agreed response time performance targets.

8. The system shall be able to handle user interface request and responses immediately.

9. Scalable: The system shall be able to conform high size of data.

3.4 Use Case diagram: [1]

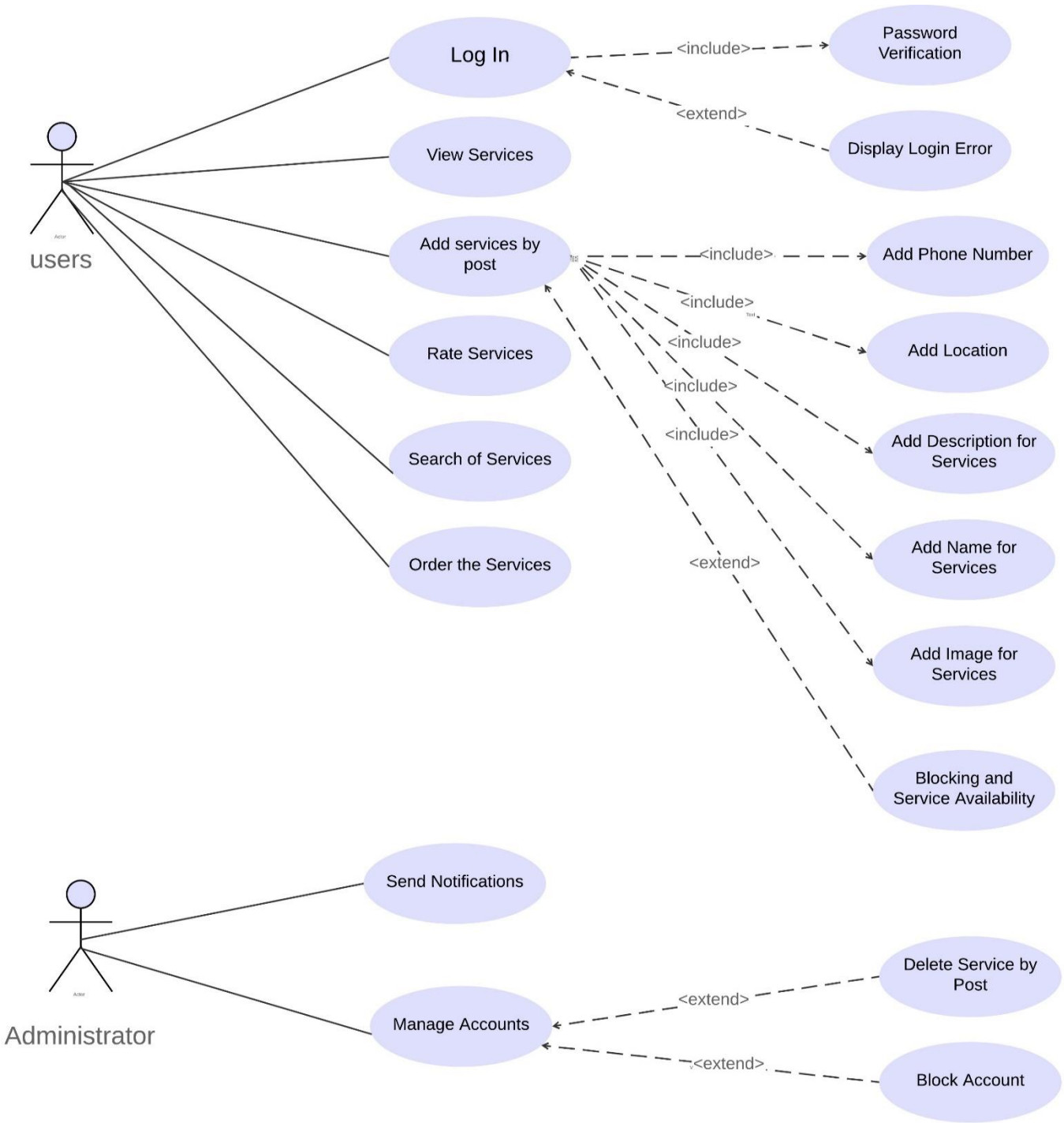


Figure 1 : Use case diagram

Use Case specification:

Use Case ID:	1
Use case name:	Log in.
Actors:	Users.
Description:	• The user login in help me application.
Pre-Conditions:	• The user has account. • System up and running.
Post-Conditions:	• Login successfully. • View the main layout.
Normal Flow:	• Enter username. • Enter password. • Click on login button. • Go to main layout.
Alternative Flow:	• Unsuccessfully login then show message excepted error: **1.** The username was invalid. **2.** Password is incorrect.

Table 1 : Log in

Use Case ID:	2
Use case name:	View Services.
Actors:	Users.
Description:	• The user wants to browse the services in main layout.
Pre-Conditions:	• The user has account. • The user should login.
Post-Conditions:	• Login successfully. • View the main layout. • The user can browse the services.
Normal Flow:	• The user can browse the services.
Alternative Flow:	• Unsuccessfully show the services.

Table 2 : View Services

Use Case ID:	3
Use case name:	Add services by post.
Actors:	Users.
Description:	• The user add service by post to the main layout.
Pre-Conditions:	• The user has account. • The user should login.
Post-Conditions:	• Login successfully. • View the main layout. • The user can add service by post.
Normal Flow:	• Enter service name. • Enter service deception. • Enter your phone number. • Add your location. • Click on add button. • The service will preview in service page.
Alternative Flow:	• Unsuccessfully to add your service if the user doesn't enter all information.

Table 3 : Add services by post

Use Case ID:	4
Use case name:	Rate Services.
Actors:	Users.
Description:	• The user can evaluate the service after they have benefited.
Pre-Conditions:	• The user has account. • The user should login. • The user has requested the service from which you have benefited.
Post-Conditions:	• Login successfully. • View the main layout. • The evaluation process was successful.
Normal Flow:	• After the user has requested the service, he can do the following: **1.** Click the Like button if the service is excellent. **2.** Click the Do Not Like button if the service is not excellent.
Alternative Flow:	• The user has the option to evaluate.

Table 4 : Rate Services

Use Case ID:	5
Use case name:	Search of Service.
Actors:	Users.
Description:	• The user can search for the service they want.
Pre-Conditions:	• The user has account. • The user should login.
Post-Conditions:	• Login successfully. • View the main layout. • View available services.
Normal Flow:	• After the user has identified the service he wants, he can search for it by: **1.** Enter the name of the service. **2.** Press the search button. **3.** View available services.
Alternative Flow:	• The service he searched for was not available.

Table 5 : Search of Services

Use Case ID:	6
Use case name:	Order the Service.
Actors:	Users.
Description:	• The user can request the service he wants.
Pre-Conditions:	• The user has account. • The user should login. • The service that the user wants is exist.
Post-Conditions:	• Login successfully. • View the main layout. • View available services. • Order the service.
Normal Flow:	• After the user has identified the service he wants, he can order the service by: **1.** Send massage to publisher of the service with Location using (Google Map API). **2.** Email of the service provider. **3.** The phone number of the service provider.
Alternative Flow:	• If the service owner rejects the request sent by the user.

Table 6 : Order the services

Use Case ID:	7
Use case name:	Send notification.
Actors:	Administrator.
Description:	• The administrator sends notifications to all users of the application.
Pre-Conditions:	• Users must exist in the application.
Post-Conditions:	• The administrator sends notifications to all users of the application.
Normal Flow:	• The administrator sends alerts to users of the application by: **1.** Access to the database. **2.** Send notifications to all users.
Alternative Flow:	• Don’t send notifications to users.

Table 7 : Send notifications

Use Case ID:	8
Use case name:	Manage accounts.
Actors:	Administrator.
Description:	• The administrator can manage user accounts.
Pre-Conditions:	• Users must exist in the application.
Post-Conditions:	• The administrator can manage user accounts.
Normal Flow:	• The administrator can manage user accounts by: **1.** Delete user accounts. **2.** Delete the infringing services.
Alternative Flow:	• **Don't send notifications to users.**

Table 8 : Manage accounts

Explanation of use case diagram [Figure 1]

This diagram is divided into two sections:

- Users
- Administrator

➢ Users:

The user creates an account and through this account can access the screen (Services screen)
And then the user can add his service by introducing the following services:

1. Service Name
2. Description of the service
3. Phone number
4. Location
5. Photos about the service

When adding a post, the service owner can hide it or show it.
The user can also browse the services and request the service that he needs by searching for them and sending a request to the owner of the service and the applicant can reject or accept the request and also the user who requested the service to assess.

➢ Administrator:

1. Send notification to users.
2. Manage accounts by blocking user or deleting the service.

4. Data flow diagram

The system will be illustrated on two different level of DFD which are:

4.1 Data Flow Diagram (Level 0): [1]

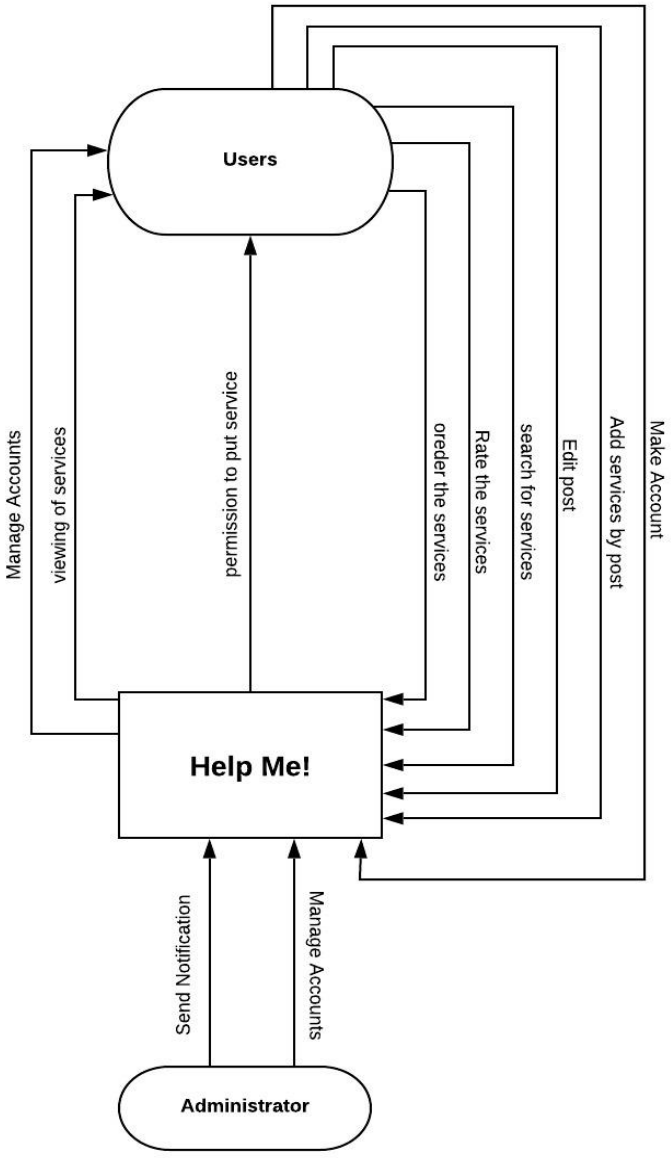


Figure 2 : Data flow diagram (level 0)

Explanation of Data flow diagram (level 0) [Figure 2]

This diagram is divided into three sections:

- Users
- Application
- Administrator

➢ **Users:**

In this section we will talk about the user's relationship with the application by:

1. The user can create an account in this application.

2. The user can add services by the post.

3. The user can modify the post.

4. The user can search for the service he wants.

5. The user can request the service he wants.

➢ **Application:**

In this section we will talk about the relationship of the application with users by:

1. The application allows the user to add the service.

2. The application represents the services to users.

3. The application allows users to manage their accounts.

➢ Administrator:

In this section we will talk about the relationship of the administrator with the application by:

1. Administrator can manage user accounts through the application.

2. Send notifications to users through the application.

4.2 Data Flow Diagram (Level 1): [1]

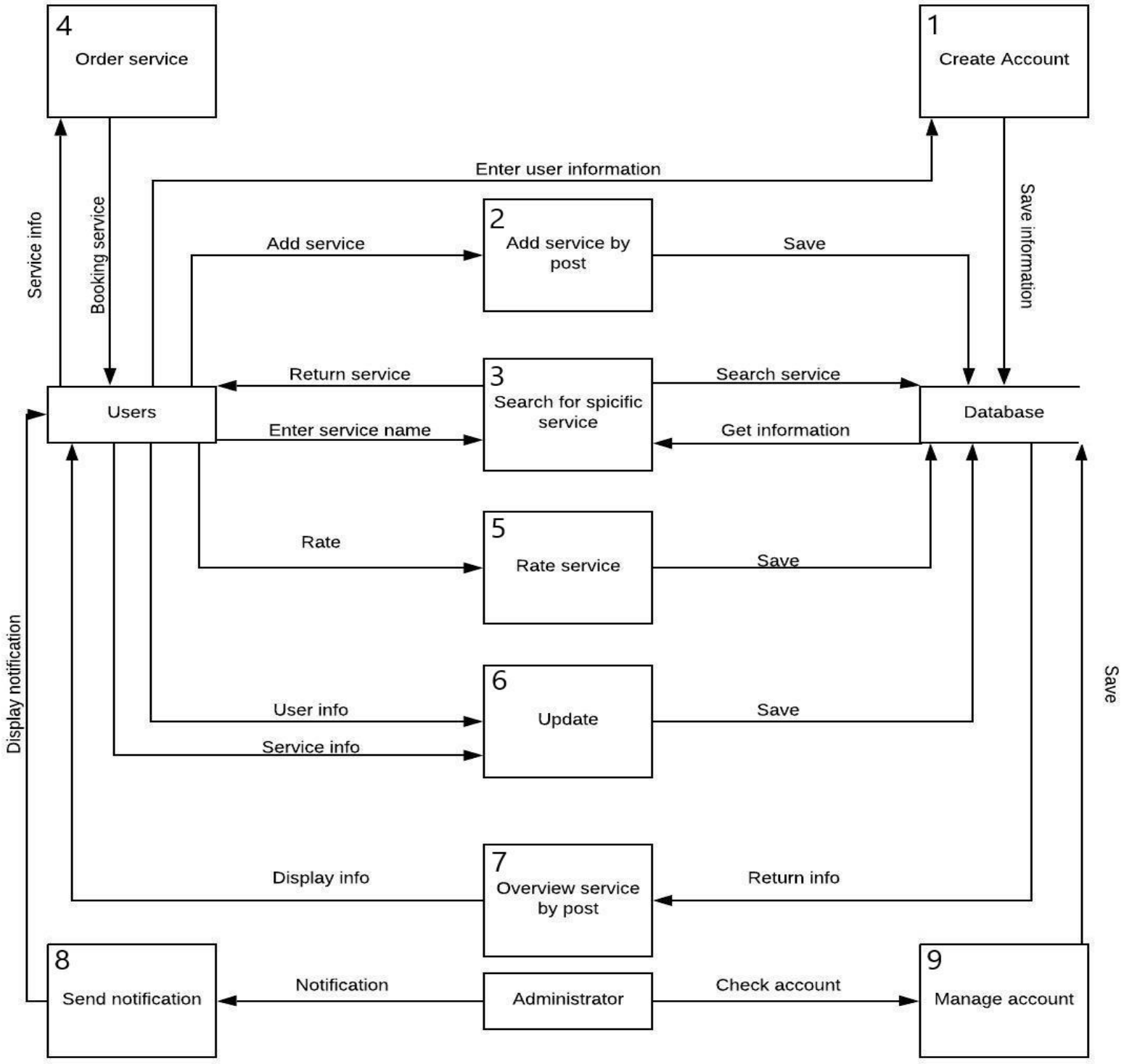


Figure 3 : Data flow diagram (level 1)

Explanation of **Data flow diagram (level 1)** [Figure 3]

This diagram is divided into two sections:

- Users
- Administrator

➢ **Users:**

First, the user creates an account by:

1. The user enters the required data.

2. The application stores the previous data in the database.

3. The user can access the main screen by entering it (Email, Password) in the login screen.

4. The application sends the data to the database for verification.

Second, the user will add his service through the post by:

1. Enter the data of service to be provided.

2. The application saves the data within its database.

Third, display the services to users in the display screen of services by:

1. Retrieve the service data that the user has entered on the database.

2. Display these data in the form of a publication on the display of services.

Fourth, search for services by:

1. Enter the service name in the search bar.

2. The application sends it to the database to search for it.

3. If the service is found it will be displayed to the user.

Fifth, the user will request the service by:

1. Sending an application to the owner of the service.

2. A message to the owner of the service.

3. The owner of the service to accept or reject the request.

4. If the service request is accepted, the location of the service applicant will be displayed and the route will be determined and accessed.

Sixth, the user shall evaluate the service after its request by:

1. Press on like button or dislike button.

2. Send the results of the evaluation to the database and display on the data screen.

- **Administrator:**

1. Admin sends out notices to the user.

2. Verification of the accounts and modification in the database.

5. Algorithmic Design

5.1 Pseudo Code:

```
if (User has Account ()) {
    Enter (Email, Password);
    Press (Log in button);
If (is correct (Email, Password)) {
    Open (overview service activity);
If (User press on Add service button ()) {
    Open (add service page);
    Enter Service information (Service name, Phone Number, Description of
    service, location, Image);
    Add Information to database ();
    Show post on overview Activity (Service information);
    }//end add post ()
If (User Press on search bar ()) {
    Enter (Service name);
    Press (search);
    SearchOnDB (service name);
```

```
    Show Services (on overview Activity);}
Else
    Show massage (“Service not found”);
    }//end search bar ()
If (user press on like ()) {
    Service was evaluated ();
    }//end on like ()
If (user press on dislike ()) {
    Service was evaluated ();
    }//end on dislike ()
If (User Press on ordering service button ()) {
    Send massage (to publisher of the service (Accept, Ignore)));
If (publisher of the service pressing on Accept button ())

    Bring the location of the person who requested the service ();
    Send approval message to user ();
else
    send reject massage to user ();
    }//end order user ()
If (update information of post ()) {
```

```
    Press on menu bar ();

    Press on my post ();

    Update your information for your post ();

    Press submit ();

    Change information in database ();

    }//end update ()

If (user want to change password ()) {

    Press on menu bar ();

    Press on change password ();

    Edit your password ();

    Press submit ();

    Change password in database ();

    } //end change pass ()

    }//end is correct

Else

    Show massage (“Email or password is not correct”)

    }//end if has account

Else
    {
    Press (signup);
    Open (signup activity);
```

Enter (name, email, password);
Press (signup);
Save data on (DB);
Open (overview service activity);
}//end if has account

5.2 Flow Charts: [1]

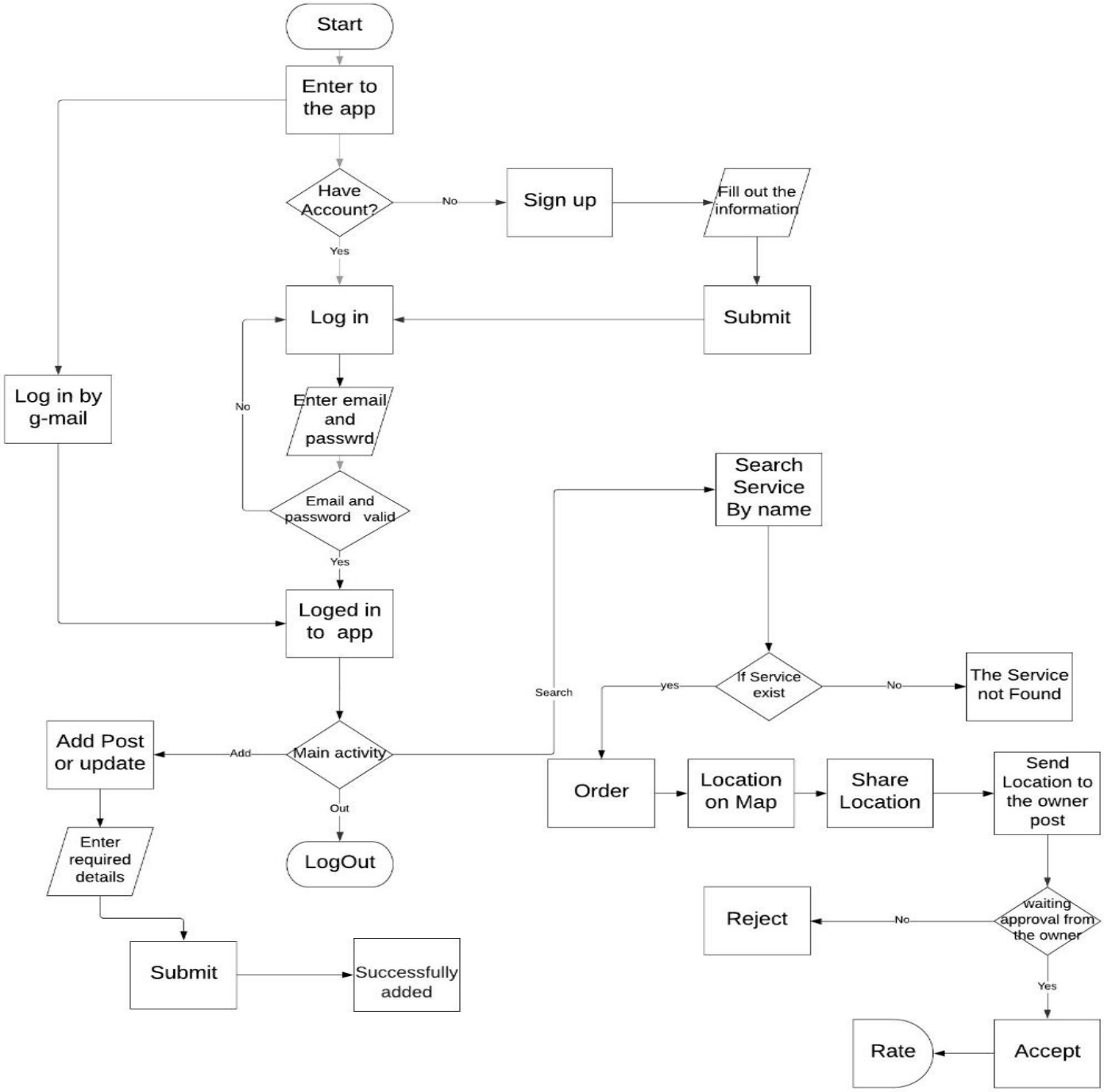


Figure 4: Flow Charts

Explanation of Flow Charts [Figure 4]

- **First, the user opens the application**

- **Second, move it to the recording screen.**

  - If the user has an account.
    - ✓ Enter your email address and password.
    - ✓ Confirm email and password.

  - If the email and password are correct.
    - ✓ Scroll to the Services view.

  Else

  - If the user does not have an account.
    - ✓ You access the Account creation screen.
    - ✓ Enter your email and password to create an account.

  - Access the Services display.

- **Third, the user can add a service.**

  - Enter the required data.
  - Move the publication to the Services view.
  - The publication has been successfully published.

- **Fourth, the user can then search for a specific service.**

  - If the service exists.
    - ✓ The service is requested by location.
    - ✓ The site is sent to the person who posted the service.
    - ✓ The service publisher can accept or reject the request.

- **Fifth, the user can log out of the application**.

6. Equipment’s

This section includes all the hardware and software that we used to make Help Me! application.

6.1 Software:

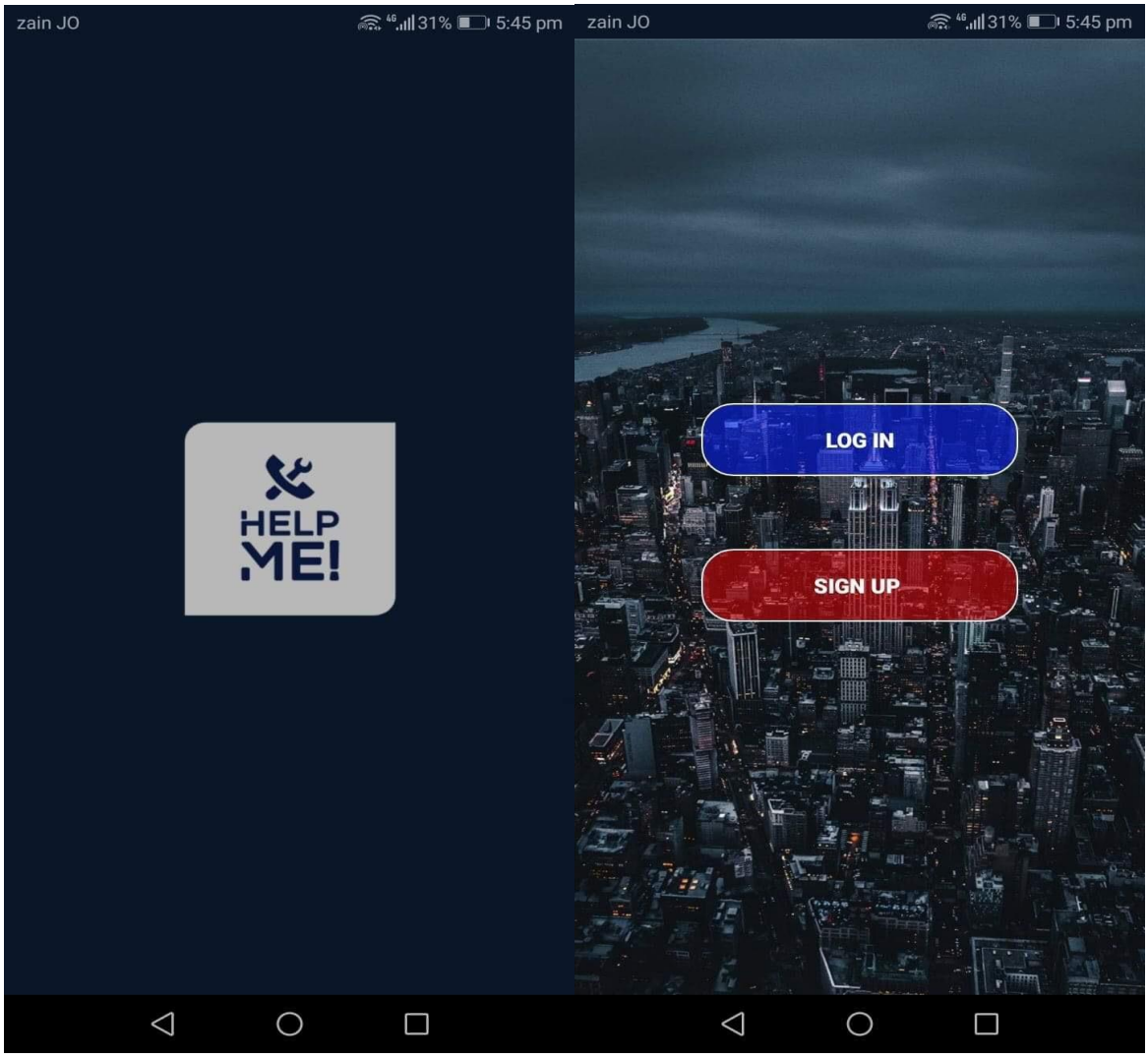
1. Android studio.
2. Eclipse.
3. Microsoft office: word 365 to write the documentation.
4. Programing language: JAVA.
5. Photoshop, to create the application’s logo.
6. Prezi, to make the presentation.

6.2 Hardware:

1. Dell laptop.
2. smart mobile phones.

7. Mechanism of use and operation

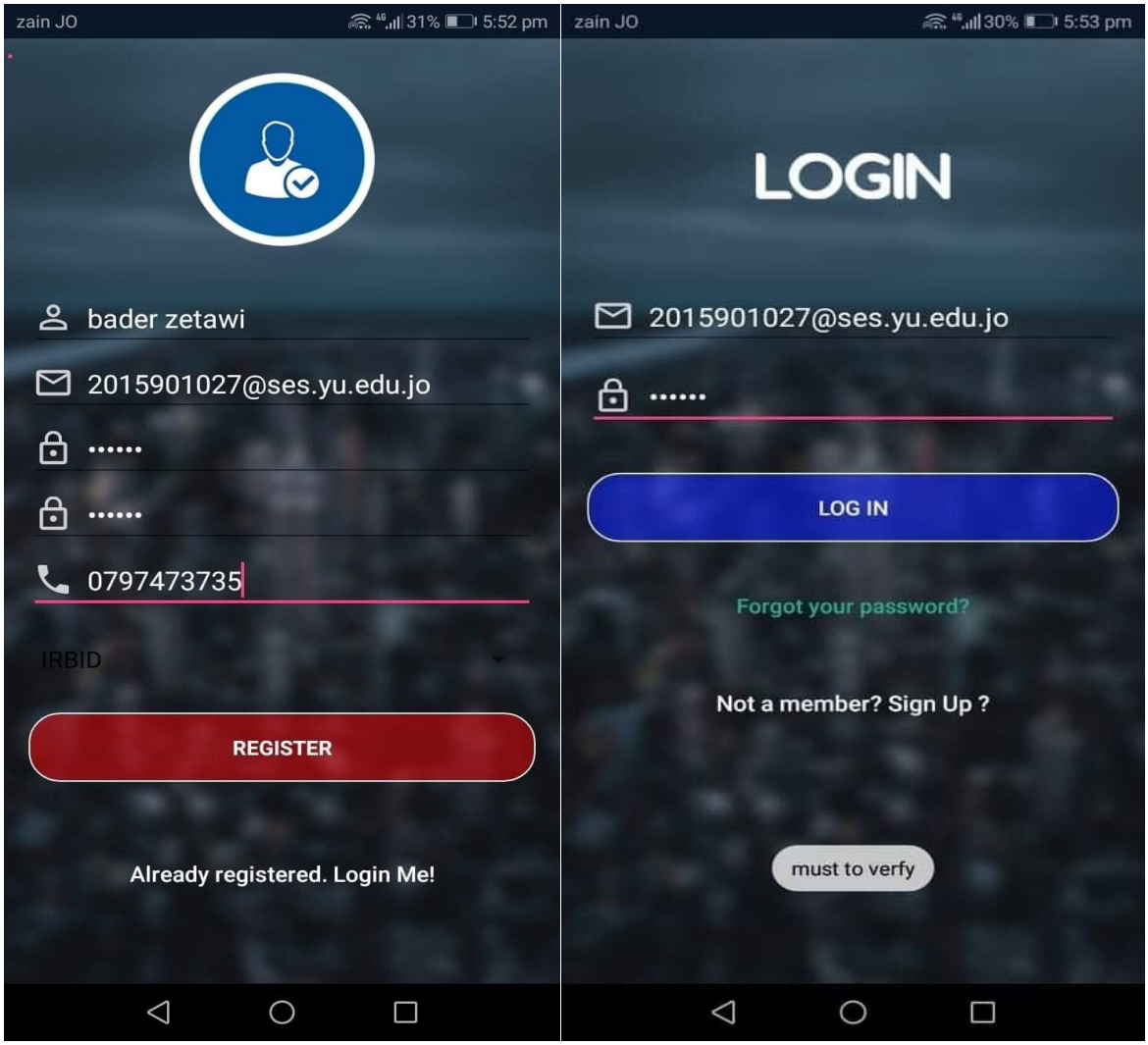
(1) **(2)**



1-Splash screen: Appear with the application running for 4 seconds and then disappear and then move to next page.

2-The login and registration screen: On this page you can go to the login page or create account page.

(1) **(2)**

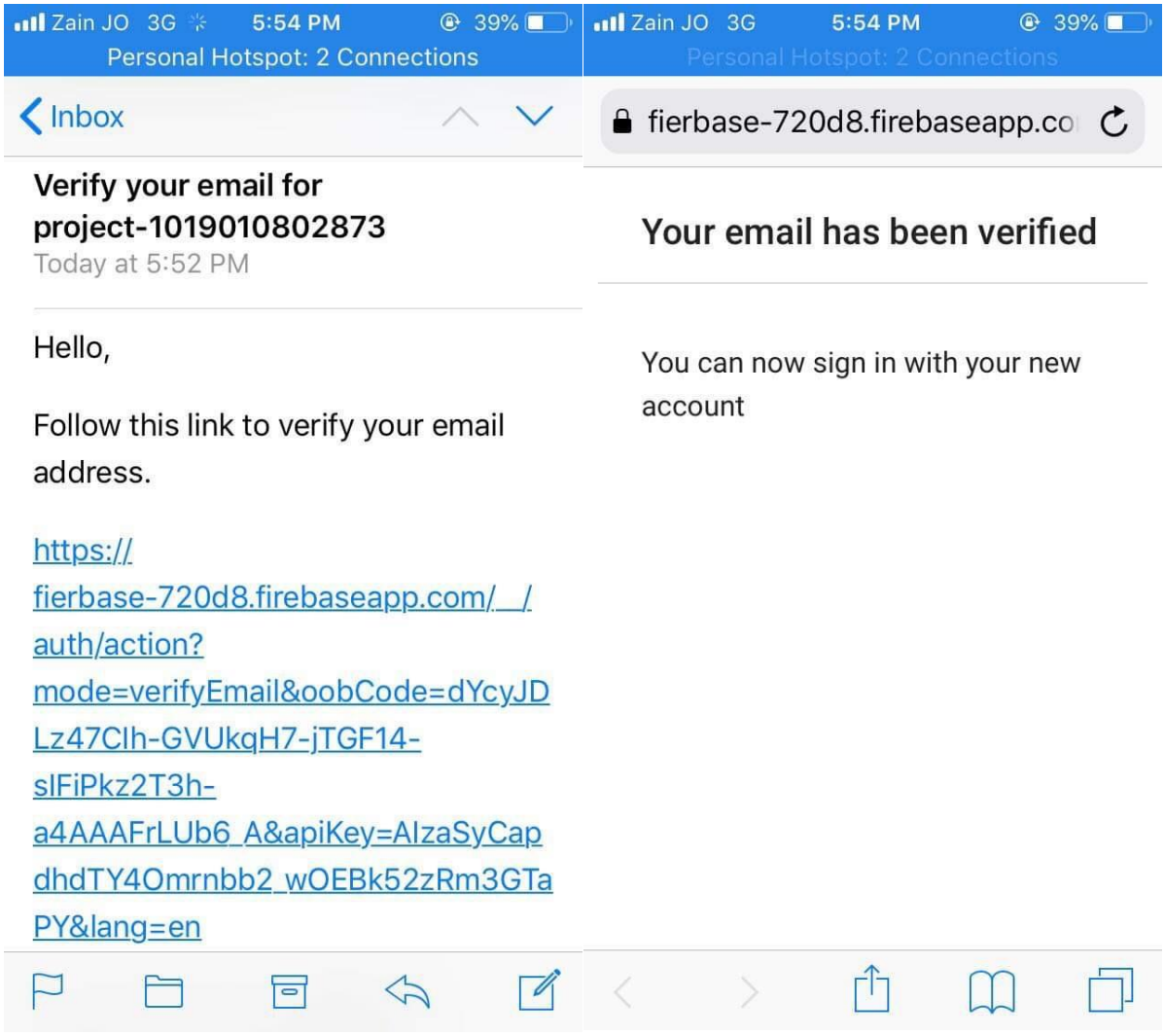


1-Registration page: The user enters his or her own information to create a new account.

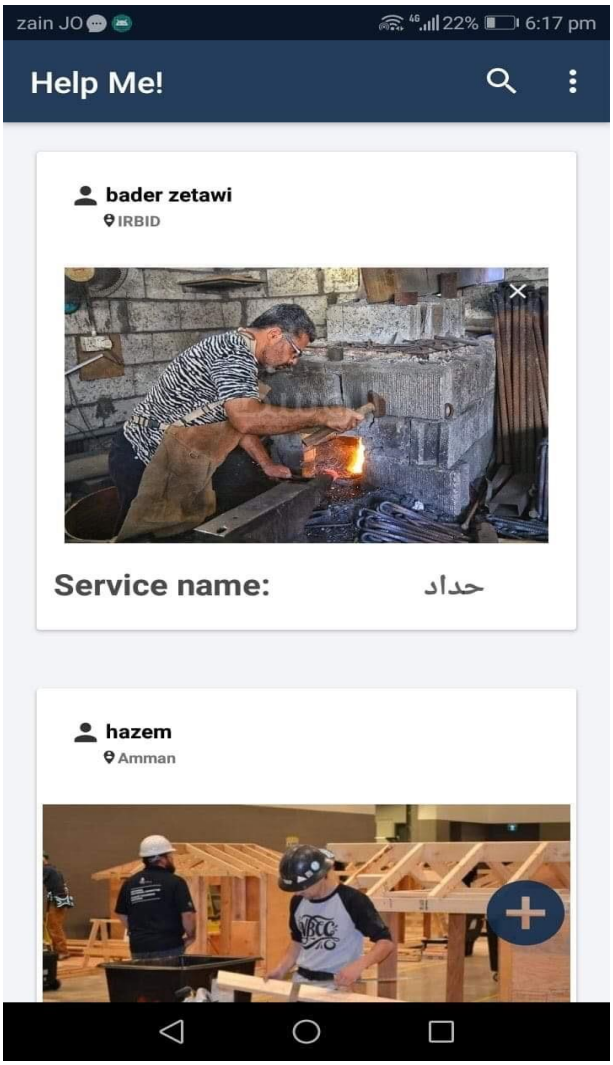
2-Login page : The user enters the email and password to access his account.

*Note: The user must verify his / her account via email.

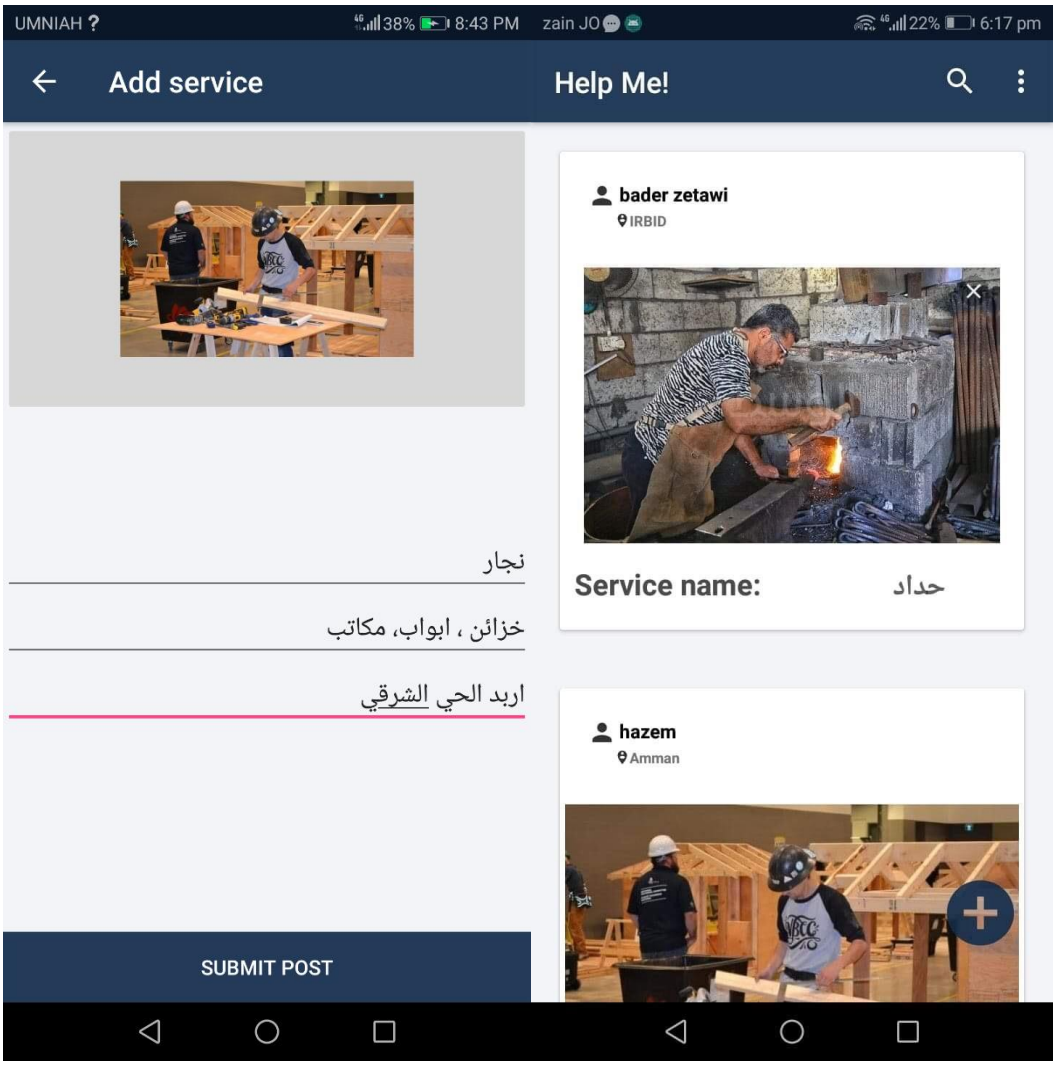
(1) **(2)**



(1 & 2) - Account verification process.

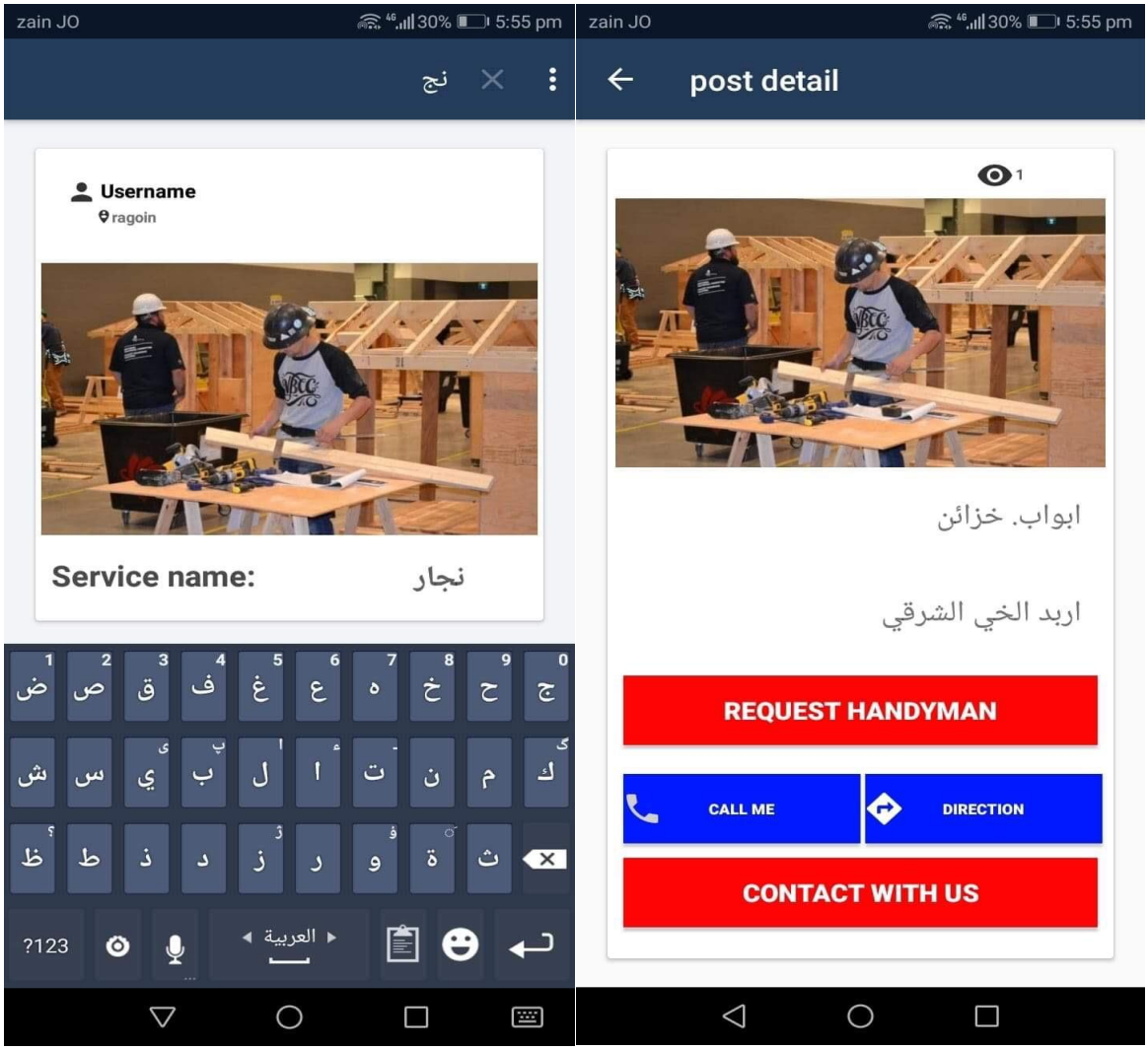


Home page: This page reviews all the services that users have added.



Add the service page:

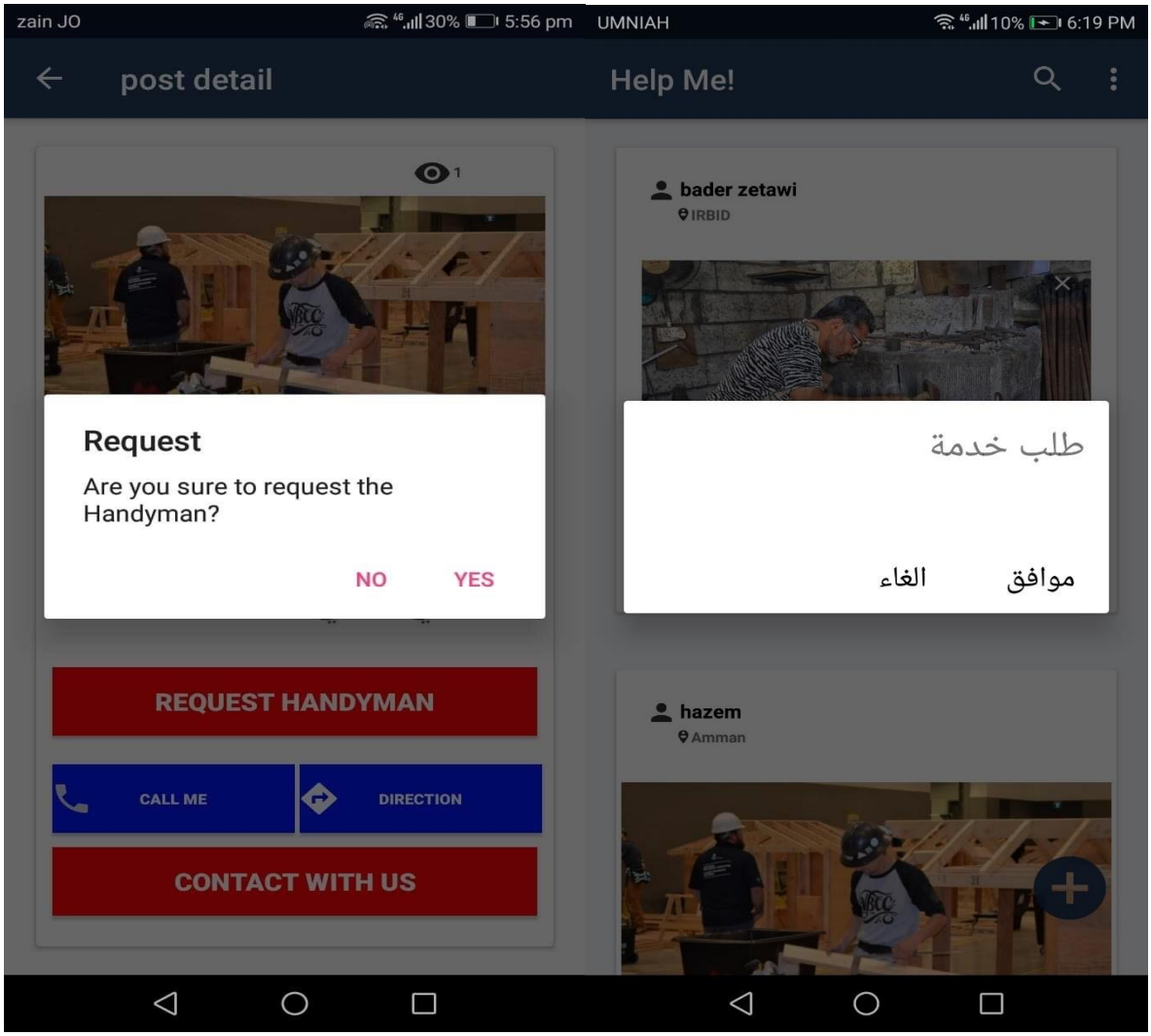
The user adds information about his / her service and the service appears on the home page.



Search for the service: The user searches for the service he wants and then presses the service and move to the service page.

Views: User can see views to participate.

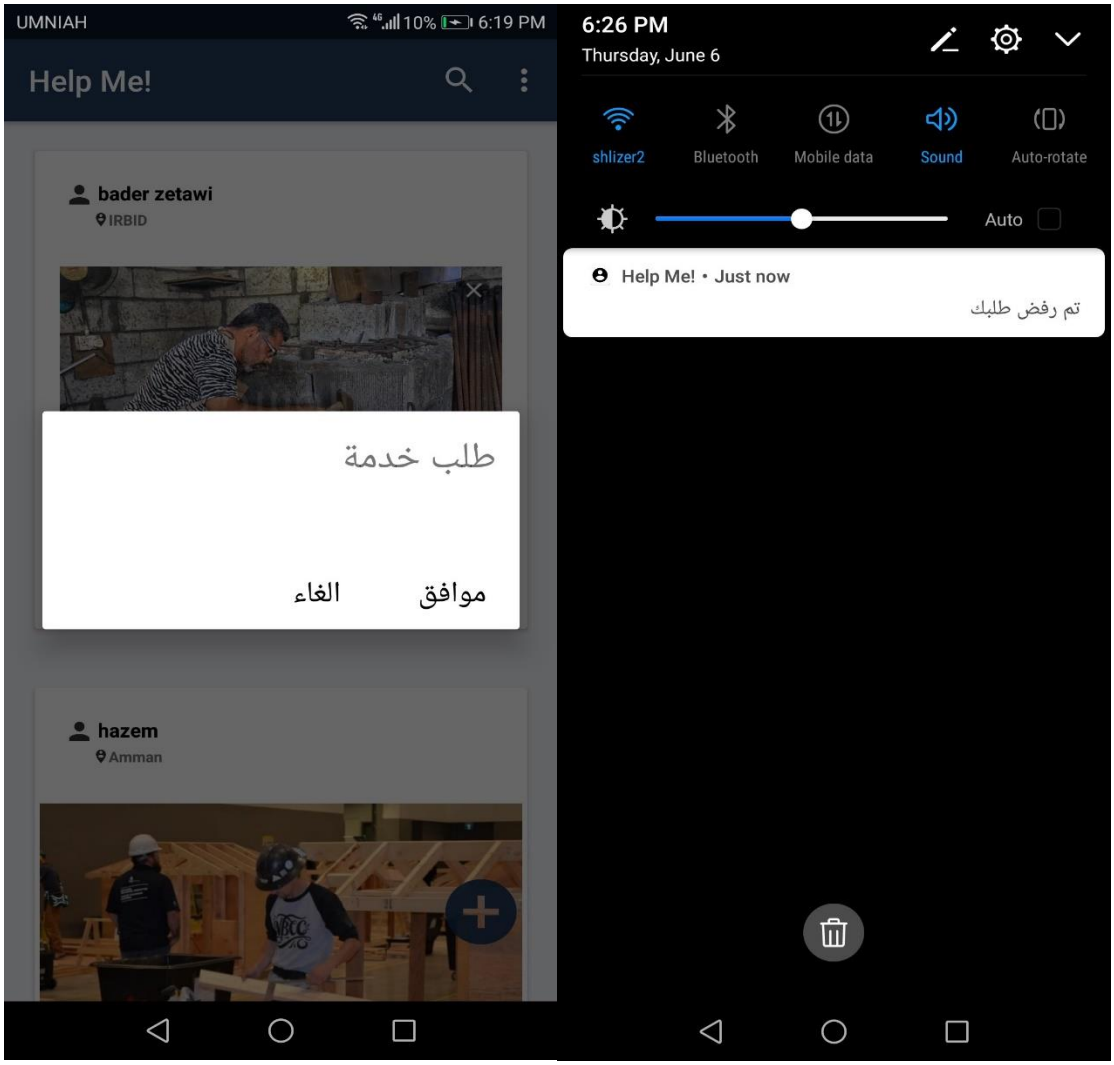
(Device 1) **(Device 2)**



Device 1: The service is requested by pressing the service request button and when pressed, it sends a notification to the service owner (device 2).

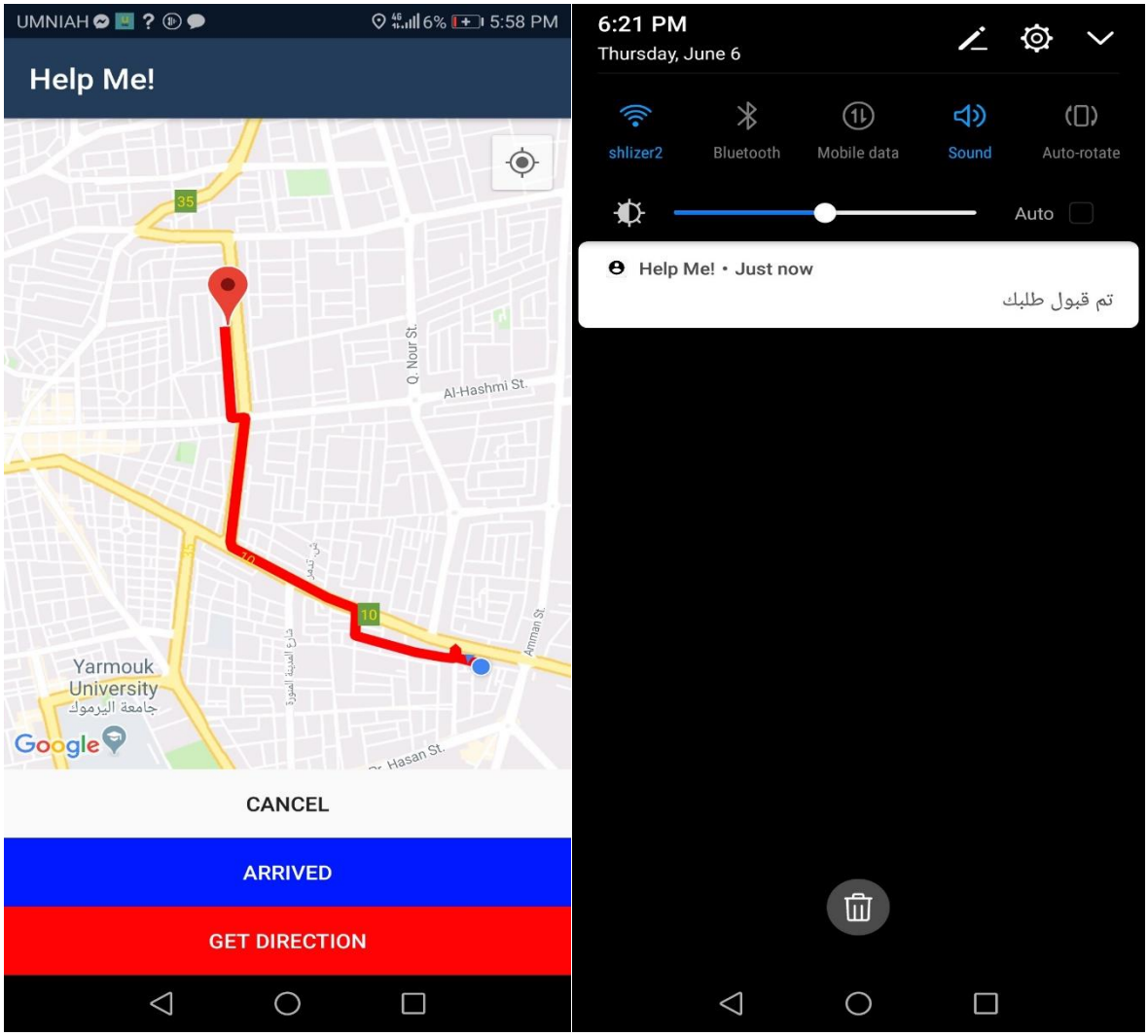
Device 2: The service owner receives (Device 2) a notification from the service applicant (Device 1) who can accept or reject it.

(Device 2) **(Device 1)**



If the service owner (Device 2) rejects the request, a notification will be sent to the applicant (Device1) (“your request was denied”)

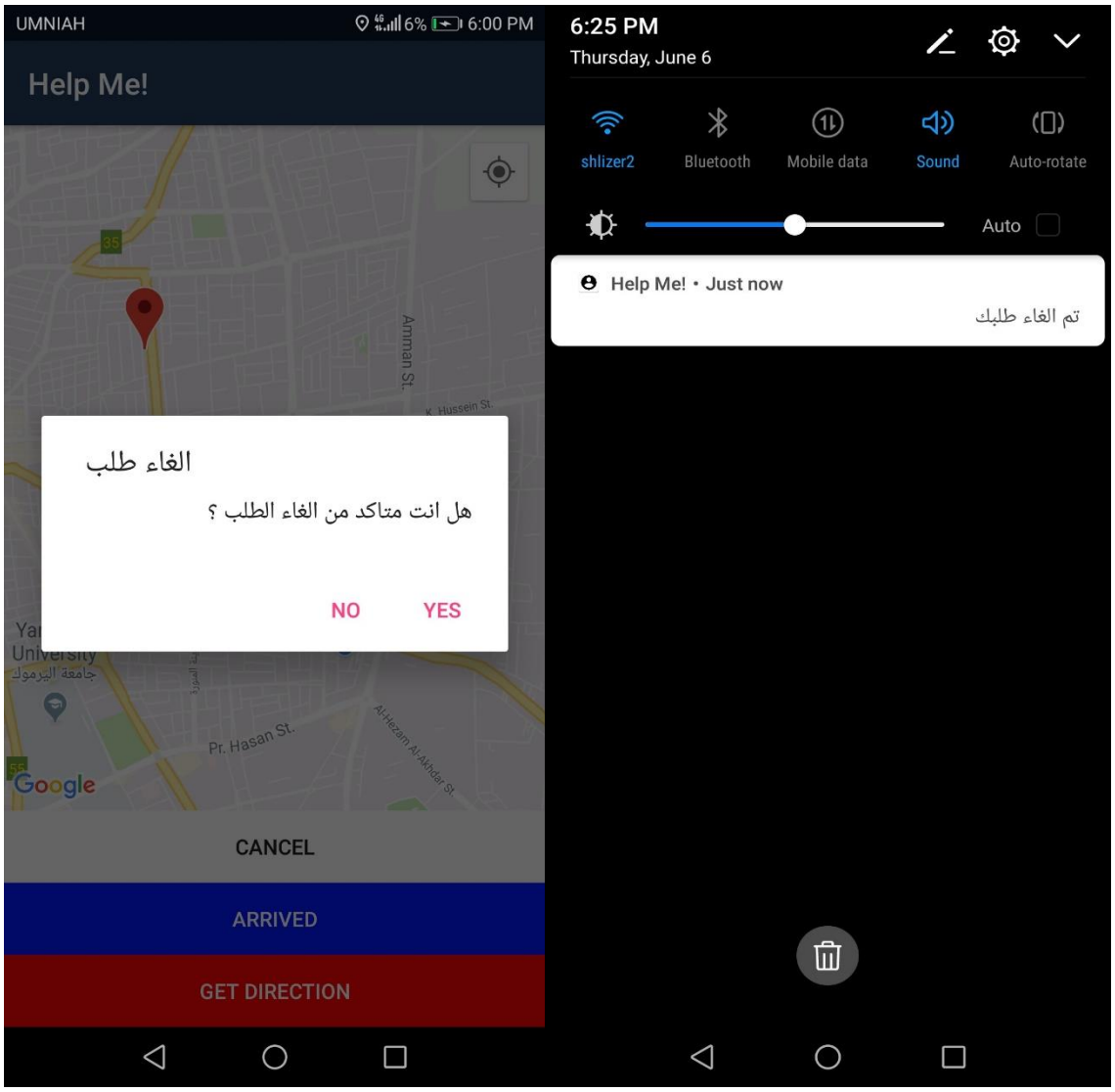
(Device 2) **(Device 1)**



Device 2: When the service owner (Device2) approves the service request from (Device1), the map and coordinates of the location are displayed and the shortest routes are determined between them.

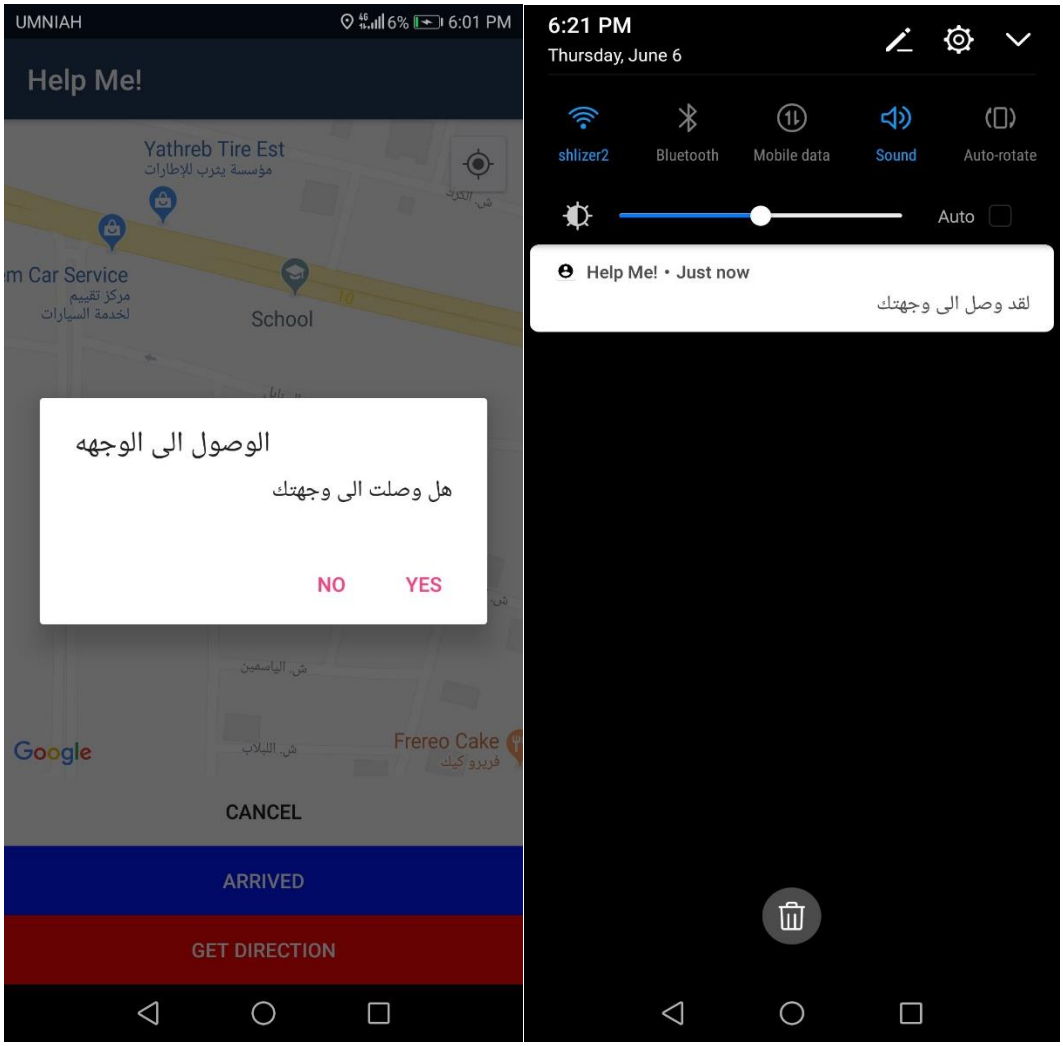
Device 1: Receives a notification from the service owner to accept the service request.

(Device 2) **(Device 1)**



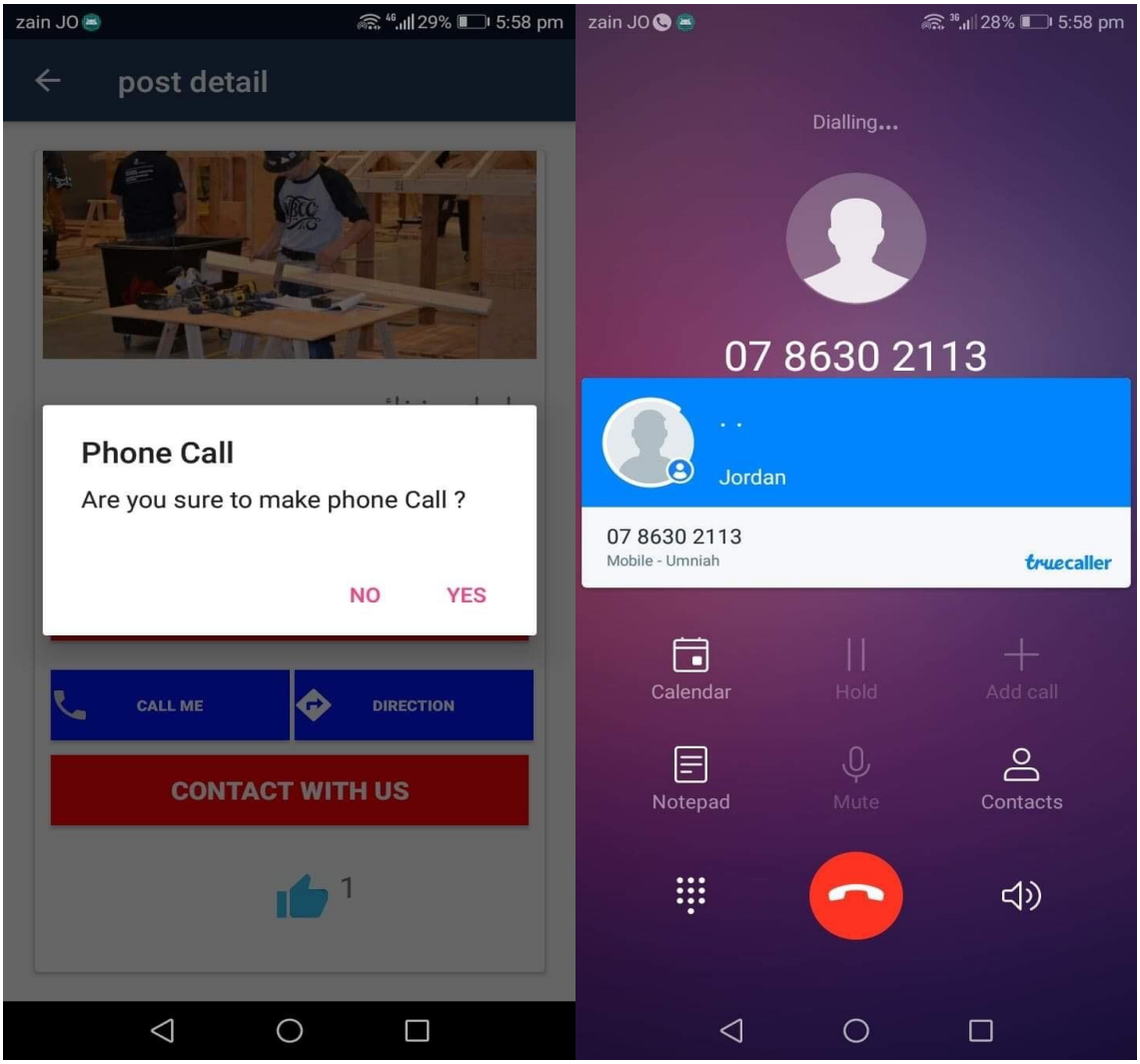
After accepting the service request, the service owner (Device 2) can cancel the service request by pressing the cancel button and then send notification to the service applicant (Device 1) that the request has been canceled.

(Device 2) **(Device 1)**



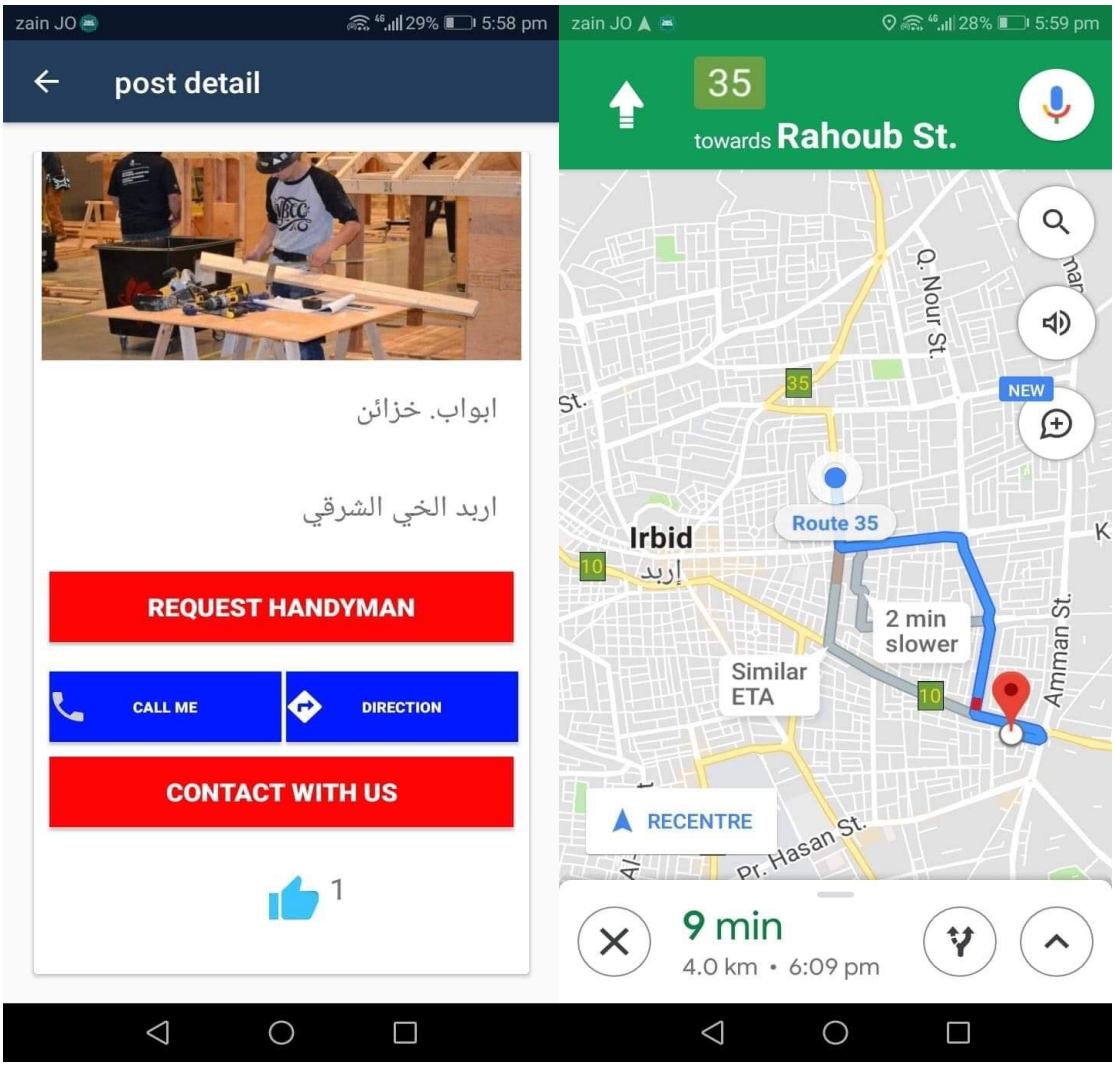
After acceptance of the service request and arrival at the intended destination, a notification is sent to the applicant that it has been reached.

(Device 1)



The user can contact the owner of the service by calling him from the phone number he entered during the registration process.

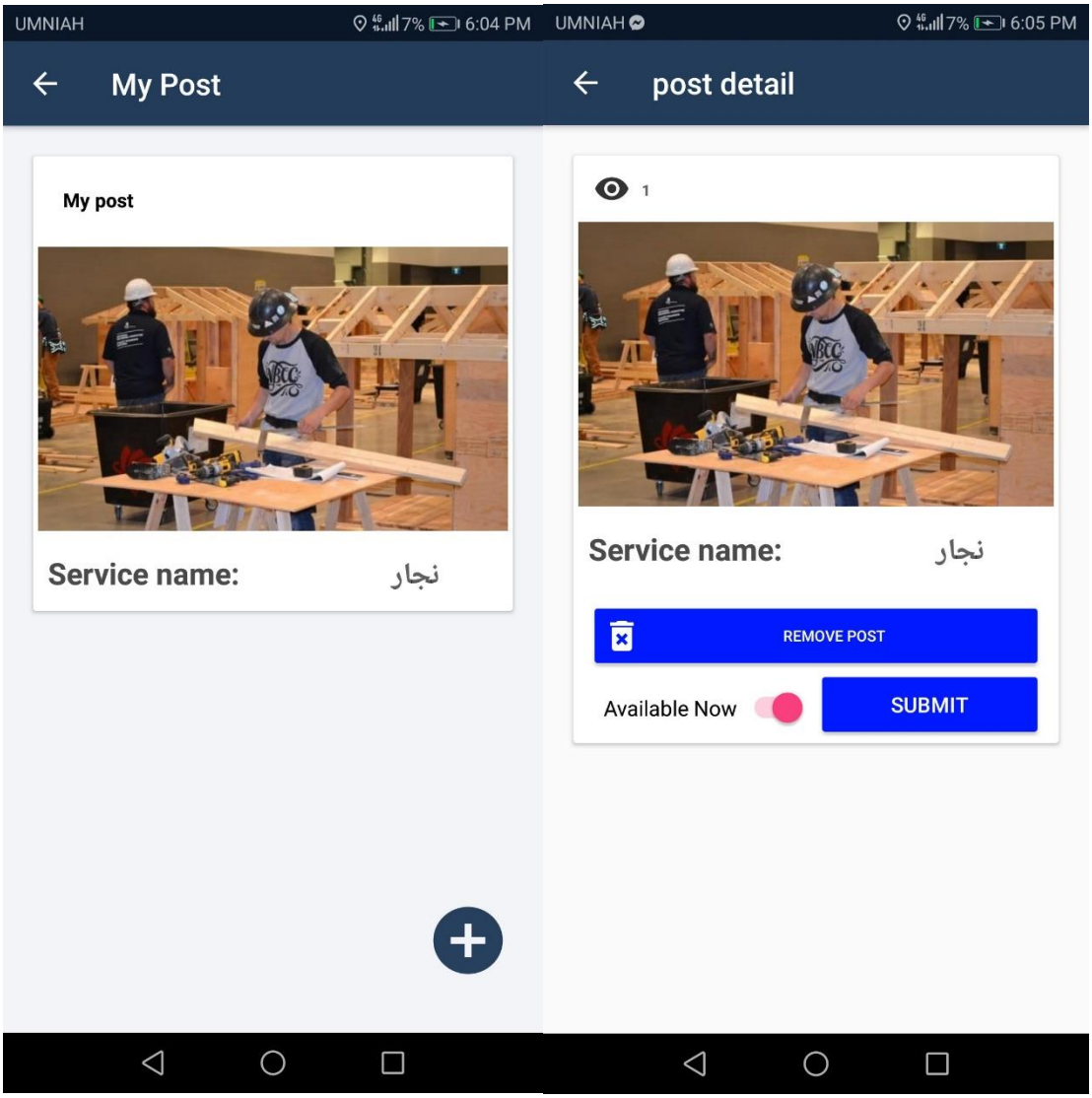
(Device 1)



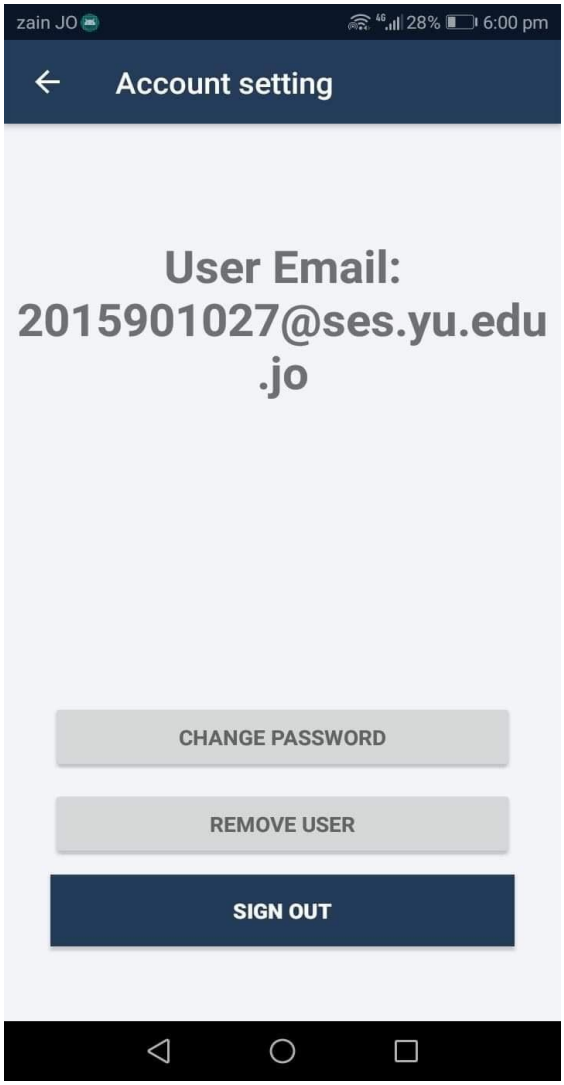
When the service user presses the Direction button, it goes to the map and shows the shortest route between him and the service owner (Device 2).

The service applicant can evaluate the service they have benefited by pressing the Like button.

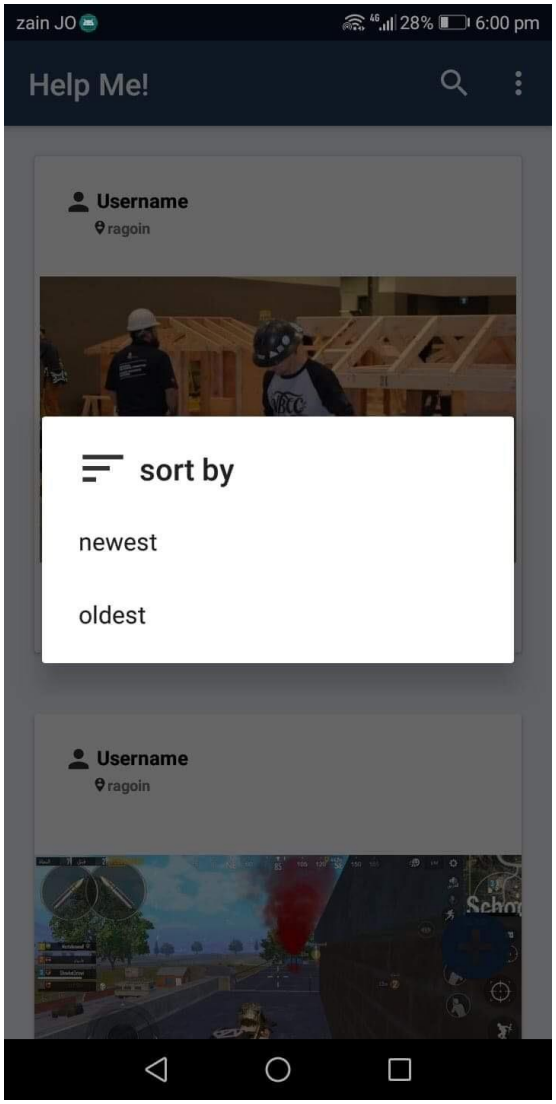
(Device 2)



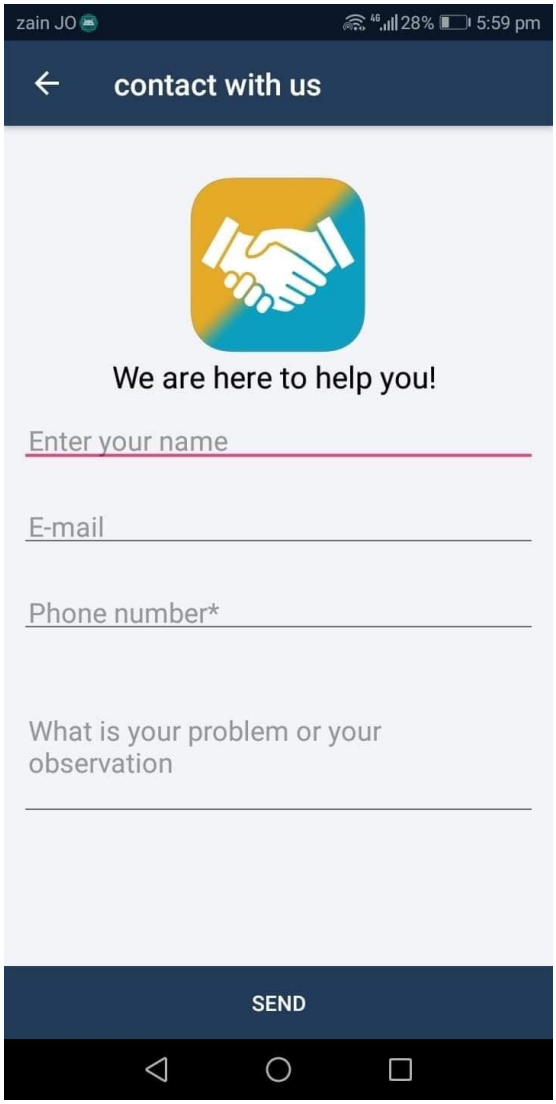
The user can see the services he has listed and can delete, enable or block the service.



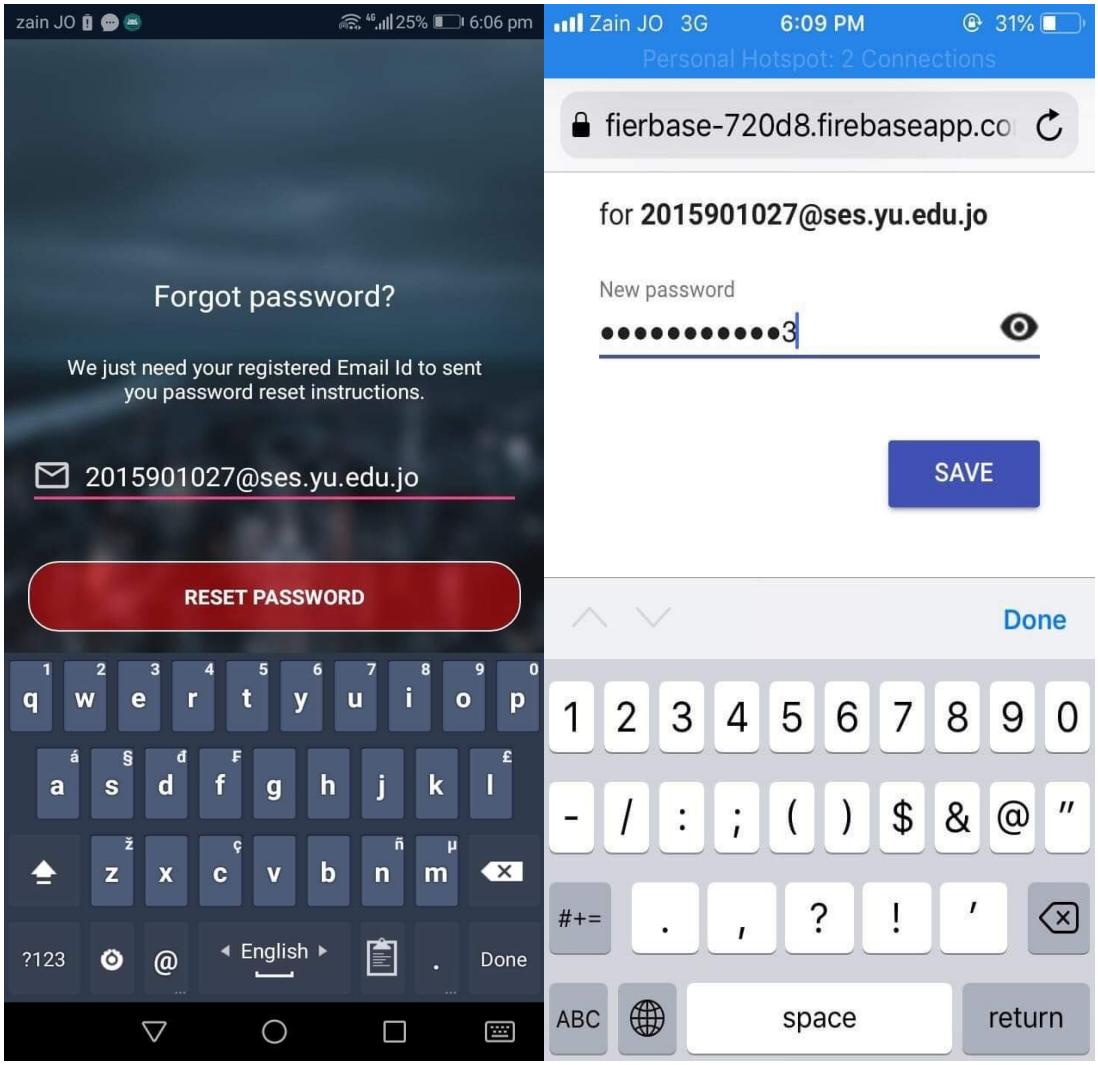
User can manage their account by changing their password, logging out or deleting their account ...



The user can arrange published services from oldest to newest or vice versa.



The user can send the problems he or she faces to the application manager and this process is helpful in the evaluation process.



When a user forgets the password, the user can reset it by placing the e-mail and a message is sent to the e-mail to change the password.

8. References

1. https://www.lucidchart.com
2. https://stackoverflow.com
3. https://www.udemy.com
4. https://developer.android.com
5. https://jo.opensooq.com
6. https://www.linkedin.com/company/3oun
7. https://itunes.apple.com/us/app/thumbtack-find-local-pros/id852703300?mt=8
8. https://www.justserve.org/