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

Gas station

A project submitted
in partial fulfillment of the requirements for the
B.Sc. Degree in Software Engineering

By

Nada Suleman Alraei (1934685)
Rand Tayseer Malkawi (1931790)
Maysam Mohammad Almaaytah (1933864)
Abdallah Essam Tahaineh (1938272)
Mais Omar Abdallah (1934033)

Supervised by

Bashar Al shboul

June/2022

CERTIFICATE

It is hereby certified that the project titled <***Gas station***>, submitted by undersigned, in partial fulfillment of the award of the degree of "bachelor's in software engineering" 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.

Nada Alraei (1934685)	**Signature**
Rand Malkawi (1931790)	**Signature**
Maysam Almaaytah (1933864)	**Signature**
Abdallah Tahaineh (1938272)	**Signature**
Mais Omar (1934033)	**Signature**

ABSTRACT

We all face a problem when we want to maintain or wash the vehicle, because of time or effort it takes, waiting until the end of these services or even reaching the station, and worry about being interrupted on long roads. So that this application come to solve this problem, by providing services to clients at their home and get service for emergency and road maintenance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

In this chapter, we will provide a general overview of our project, then we will clarify the main idea of the project in the following subsections

1.1 Overview

In the recent period, we have noticed the great reliance on technology and the demand for mobile applications, and how people's lives have become easier and more comfortable with the presence of such applications. Therefore, the idea of our project is to save people's time and effort by providing the following basic services:

1-Gasoline and diesel filling for heating systems and cars.

2- Car wash and dry clean.

3- Flat tires change.

4-Oil change

5- Battery change.

1.2 Project Motivation

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

In our country, there are many applications that provide services, such as online shopping, online billing, food delivery, money transactions, and taxi service.

With the huge increase in cars numbers, some issues and problems became a headache for people such as car maintenance, gas filling, and traffic so we came up with an idea why not to provide these services online to save people's time, effort, and to avoid the traffic.

Q2. Why your project is important?

Car maintenance and gas filling are essential things that all people need so we decided to give people an easy and convenient way to get them.

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

We were not able to find an application that provides the services we are proposing in a convenient fashion.

1.3 Problem Statement

The only way to fill gas, flat tires change, car oil changes, and battery change is to head to the gas station and car maintenance workshops which causes traffic and waste of time\effort

1.4 Project Aim and Objectives

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

-Reduce pressure at gas stations.

-Reducing customer effort and time.

-Providing services in difficult weather conditions.

-To make our life easier and more comfortable, especially since we live in a huge digital revolution.

Q2. How can this project achieve this goal?

By using the application, in a simple way, the clients will sign up at any time and in specific cities and make an order for the needed service

1.5 Project Scope

Responsibilities in this project are divided between three sections:

1) Clients: clients should sign into the app if it was the first time to use, after that they can access the app using log in only. They can choose the service or product needed from the desired station; they can edit their information whenever they want. Expressing satisfaction through rating.

2)Gas station manager: he should sign into the app prior first time to use, after that he can access to the app using log in only. Reject or accept the order. He inspects the list of available delivery providers to decide whether to accept or cancel the request. He selects one delivery provider from those who are available and sends order to deliver it.

3)Delivery man: he should sign into the app if it was the first time to use, after that their can access to the app using log in only. He receives the request from the station and confirm it.

1.6 Project Software and Hardware Requirements

- For programming and documentation:
  - Computer or laptop.
  - Visual studio code.
  - Android studio.
  - Excel.
  - Marvel.
  - Lucid.
- For use:
  - Smart phone.
  - Google play.

1.7 Project Limitations

1)The delivery service may be delayed sometimes, either due to the pressure and a large number of requests in some events such as bad weather.

2)There are some locations deliveries hard to reach because it is fare away from city centres, and then we will deliver the order as close as possible.

3)Few users at a time due to the limited number of delivery cars.

1.8 Project Expected Output

Expect to build a mobile application that will enable customers to order gas station services online.

1.9 Project Schedule

Table (1): Project schedule

Activates	Number of weeks	Start day
Introduction	1 Week	3/18
Literature Review	2 Week	3/25
Requirement Engineering and Analysis	1 Week	4/8
Architecture and Design	1 Week	4/15
Implementation Plan	1 Week	4/22
Testing Plan	2 Week	4/30
Conclusion And Results	1 Week	5/14

1.10 Project, product, and schedule risks

There are no big risks, there are some fears of attracting cassette owners in station and delivery workers

1.11 Report Organization

The rest of the report is organized as follow, **Chapter 1** introduction about project, **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** talks about testing plan and software testing, **Chapter 7** presents future work and concludes the report.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

In this chapter, we will list the Existing systems that provide service similar to the subject of our project, show the problem for this system and suggest solutions of this problem from our point of view.

2.2 Existing systems

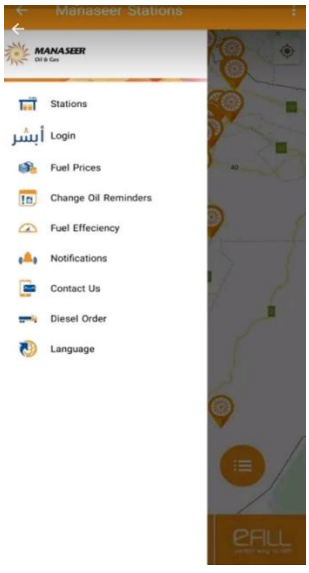
There are many existing systems of the same subject as our project some of them are in Arab world and other are in Jordan, we list the existing system in Jordan as example:

1.Manaseer Station:

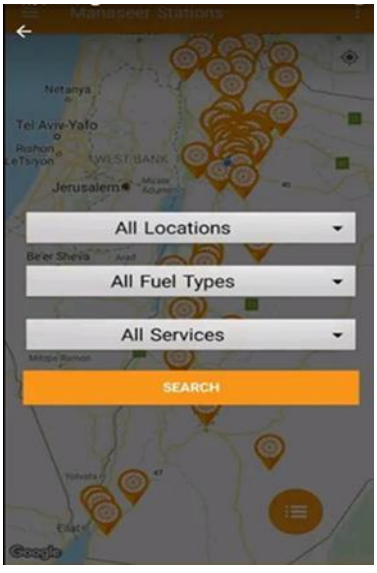
This application provides many services most of them based on notifications and give information about vehicles not the delivery of services, these services are:

- Nearest station.
- Latest fuel prices.
- Change oil reminder.
- Fuel efficiency for vehicles.
- Diesel order.
- Can determine nearest station depending on the fuel types and service need.

Figure(1): Nearest Manaseer Stations



Figure(2): Manaseer Service Menu



Figure(3): Manaseer Search

2.3.1 Overall problems of Manaseer Station:

1. Limited delivery service, that provides it only for **Diesel order**.
2. Do not provide daily or emergency services to vehicles.

2.4.1 Overall solution approach of Manaseer Station:

1. Increasing services to include the largest percentage of vehicles services.
2. Dedicate a team for emergency and daily service (after adding this type of services).

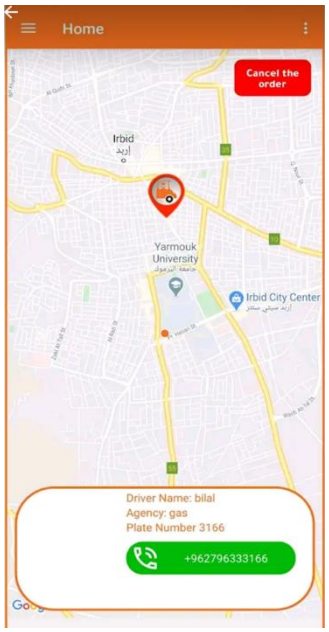
2.GazDel:

This application specialist in gas cylinder order only, this app services

Are:

- Gas cylinder order.
- View previous orders.
- Determine required quantity.
- View the drive`s name and phone number after order confirmation (to call him when needed).
- Rate the service after driver arrive.

Figure (4): GazDel Determine Order



Figure(5): GazDel Order Information

Figure(6): GazDel Order Done

2.3.2 Overall problems of GazDel:

1. The service is only in the city of Irbid.
2. It does not provide any advantage, that can be dispensed with by taking the phone number of any gas cylinder delivery man.

2.4.2 Overall solution approach of GazDel:

1. Gradual expansion of services delivery and increase number of drivers, to be able to achieve this expansion.
2. Add advantage such as deliver the service faster, the more people use the app.

CHAPTER 3: REQUIREMENT ENGINEERING AND ANALYSIS

3.1 Stakeholders

Figure 7 shows the primary stakeholders of the Gas Station App: Developers, Supplier, and Client. **Application admin** is the owners of the application and the project, **supplier** is the person responsible for the delivery and processing of the order from the station, **client** is the person receiving the service.

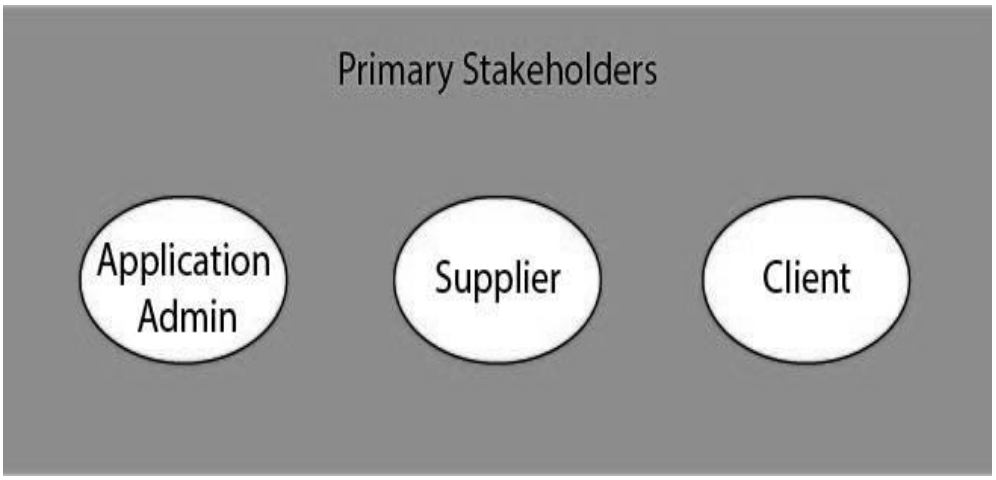


Figure (7): Primary Stakeholders

Figure 8 Show the second stakeholders of gas station application: **the ministry of energy and mineral resources (MEMR)** which is responsible for monitoring the quality of fuel for fuel stations, **Agent Persons** responsible for other services.

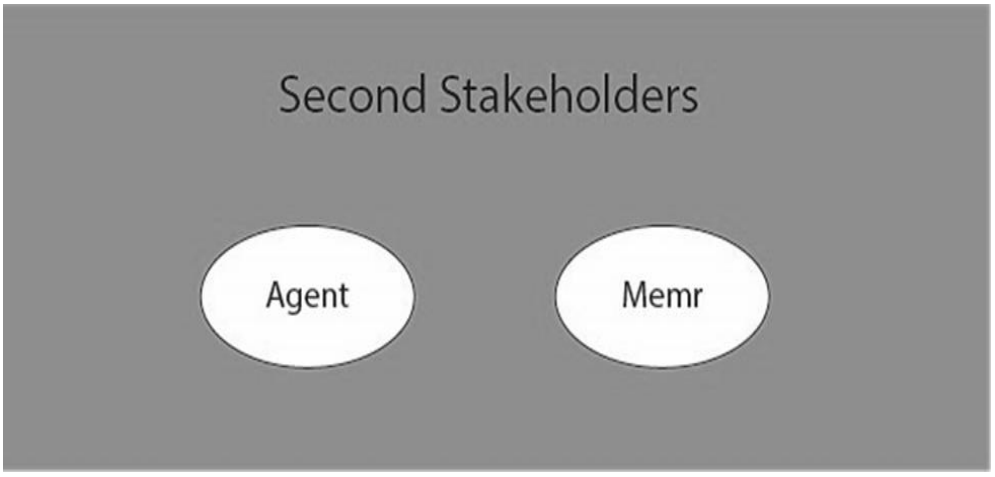


Figure (8): Second Stakeholders

3.2 Use Case Diagram

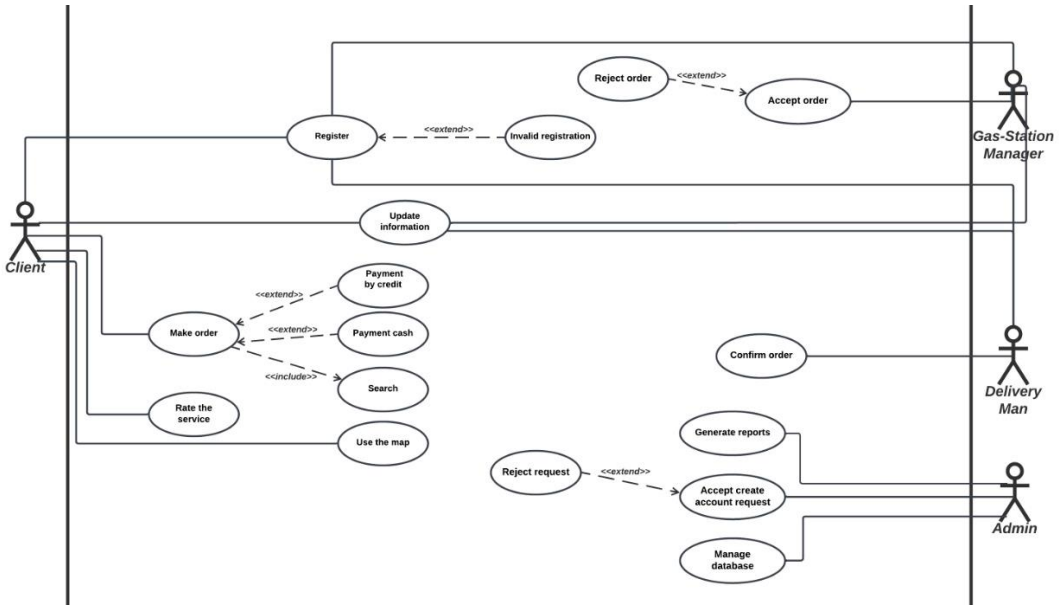


Figure (9): Use Case Diagram

3.2.1 Use Case Section and alternative flows

Table (2): Add gas station

Use case	Description	Action	Pre-condition	Post-condition
1.*Add gas station*	**1.** Admin manually verify the station information. **2.** Admin enter station information into the app.	**1.** Admin press add gas station button. **2.** Admin enter station information **3.** Station add to database **4.** Admin get gas station username and password	Request to create account from station.	Gas station has account

Table (3): Add delivery man

Use case	Description	Action	Pre-condition	Post-condition
1.*Add gas station*	**1.** Admin manually verify the delivery man information. **2.** Admin enter delivery man information into the app.	**1.** Admin press add delivery man button. **2.** Admin enter delivery man information **3.** Delivery man add to database **4.** Admin get delivery man username and password	Request to create account from delivery man.	Delivery man has account

Table (4): Report

Use case	Description	Action	Pre-condition	Post-condition
3.*Reports*	The admin monitors the activities in the application.	Admin select app activity.	Register as admin.	Monitor the app and determine if it need any update.

Table (5): Register

Use case	Description	Action	Pre-condition	Post-condition
4. *Register* ***《extend》*** *Invalid registration*	**1.** Create account for a new client. **2.** Log in for registered user using needed information.	**1.** For create account, client enters require information. enter sent code. **2.** For log in, user enters their Username and password.	User must install the application	Have an account and use the application features.

Table (6): Update information

Use case	Description	Action	Pre-condition	Post-condition
5.*Update Information*	Allow user to change their information	User press on update information button if he wants to update his information	User must be registered	Information has been updated

Table (7): Map

Use case	Description	Action	Pre-condition	Post-condition
6.*Map*	User Know the nearest gas station depending on their location.	User press on map button	User entered his location	User Know the nearest gas station

Table (8): Make Order

Use case	Description	Action	Pre-condition	Post-condition
7.*Make Order* ***《extend》*** *Payment by credit, Payment cash* ***《include》*** *Search*	Allow clients to select order, gas station and payment method. Also search for specific station and using map to know the location of station.	User press on map button	User must be registered	The request has been submitted to the station's system

Table (9): Rate the services

Use case	Description	Action	Pre-condition	Post-condition
8.*Rate the services*	Allow clients to rate the services after getting it	Choose from 1-5 displayed options	User must be registered	Obtaining feedback from the client on the level of services

Table (10): Accept Order

Use case	Description	Action	Pre-condition	Post-condition
9.*Accept Order* ***《extend》*** *Reject order*	Allow manager to accept or reject an order	**1.**Managers check the list of available delivery man. **2.** If there are available one, accept the order and select the suitable one. **3.** If no available delivery man, the order will reject.	User must be registered	Order must deliver to selected delivery men

Table (11): Confirm order

Use case	Description	Action	Pre-condition	Post-condition
10.*Confirm order*	Allow delivery man to confirm order	Select the order to deliver it and press on complete button	User must be registered	Deliver the order to client

3.3 Non-Functional User Requirements

Execution qualities:

- ✓ Security: the system must protect users' information from disclosure or loss.
- ✓ Usability: the app should be simple and easy to use for all considered end users.
- ✓ Availability: the system must be available anytime the users need it (24/7).

Evaluation qualities:

- ✓ Testability: should be able to test in any step of system development life cycle.
- ✓ Maintainability: should be able to modify and add any option according to user needs.
- ✓ Extensibility: should be able to expand and increase the system service such as, add cities or add new station.
- ✓ Scalability: should be able to increase a performance such as speed, when the number of users increase.

Constraints

1. Internet connection is required for all type of users.
2. The client should have account to be able to use the app.
3. The station should have account to be able to provide its service through this app.
4. The delivery man should have account to be able to receive orders from the station.

CHAPTER4: ARCHITECTURE AND DESIGN

4.1 Overview

In this chapter we will view Software Architecture (class diagram, deployment diagram) and Software design (Sequence diagram), to show classes' relationship based on the use case scenarios

4.2 Software Architecture

4.2.1 Logical view

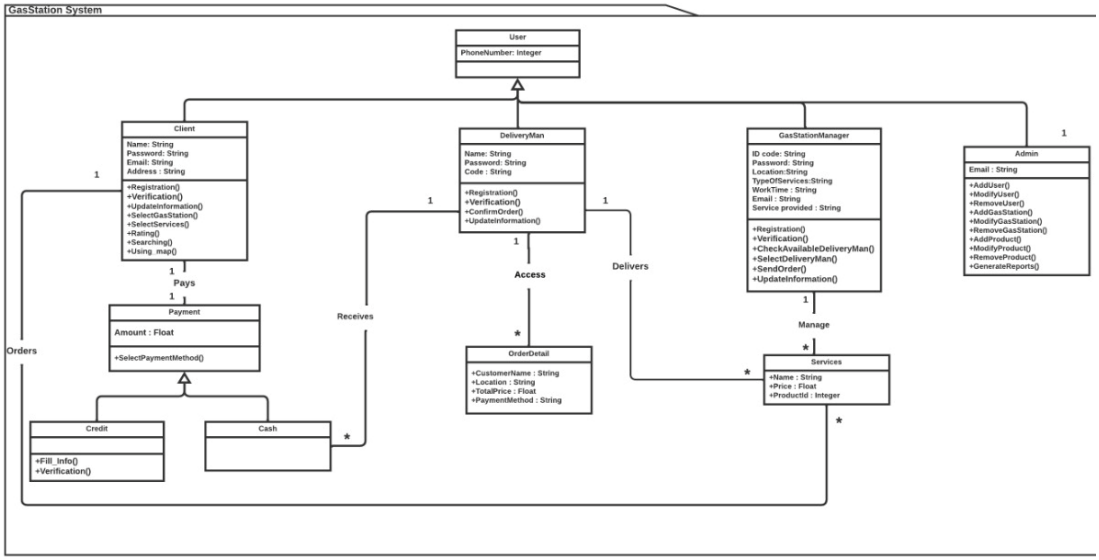


Figure (10): Class diagram

4.2.2 Physical view

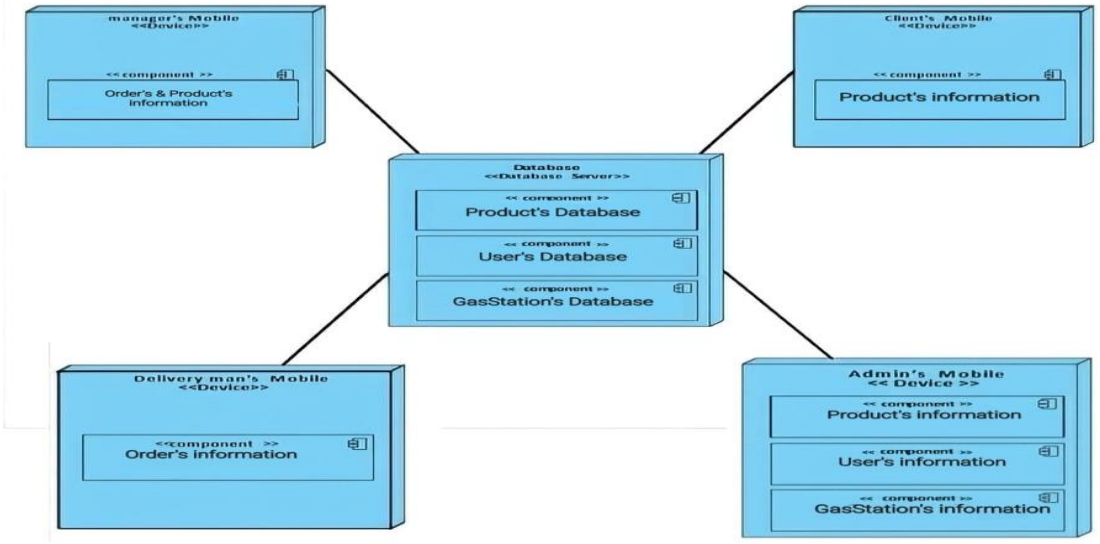


Figure (11): Deployment diagram

4.3 Software design

4.3.1 UML sequence/communication diagram

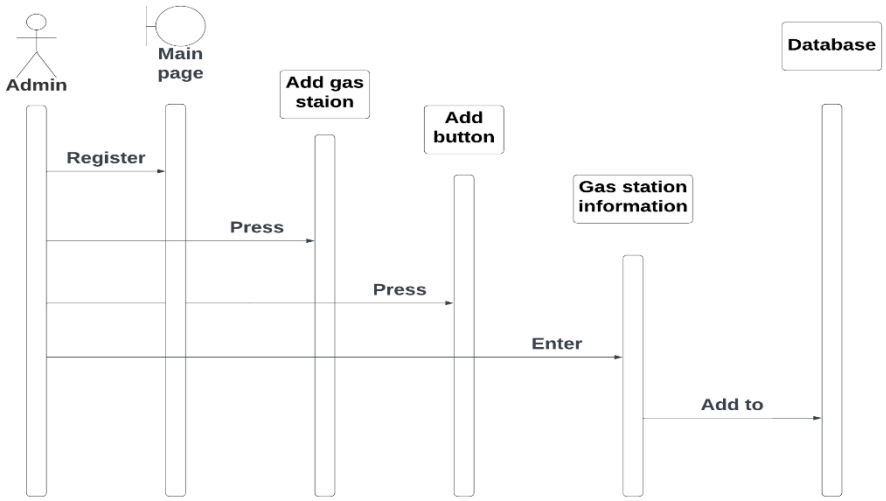


Figure (12): Add gas station

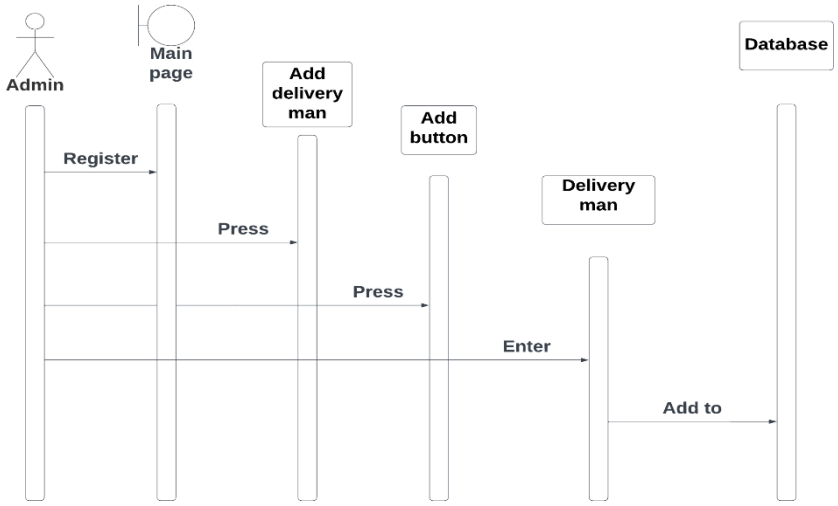


Figure (13): Add delivery man

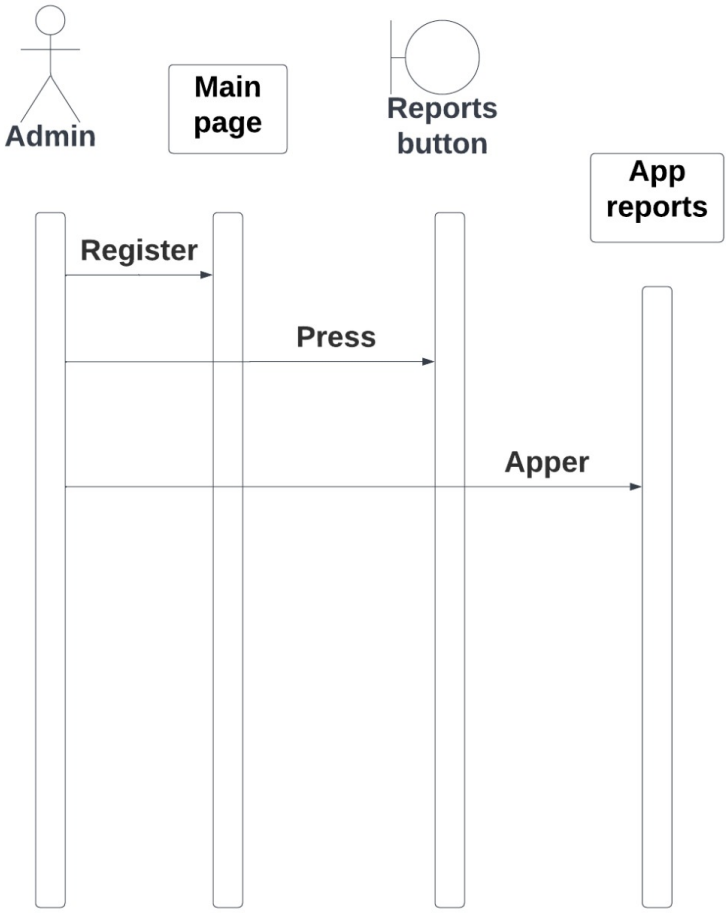


Figure (14): Report

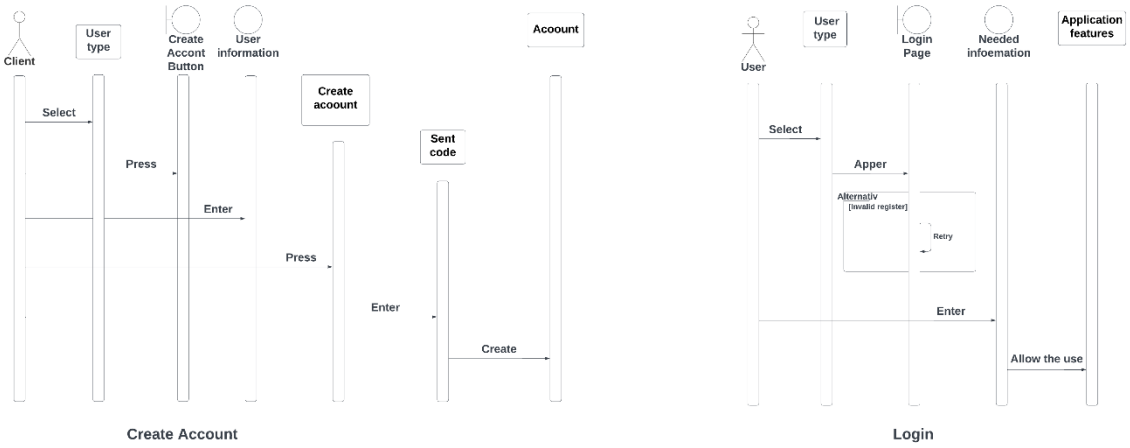


Figure (15): Registration

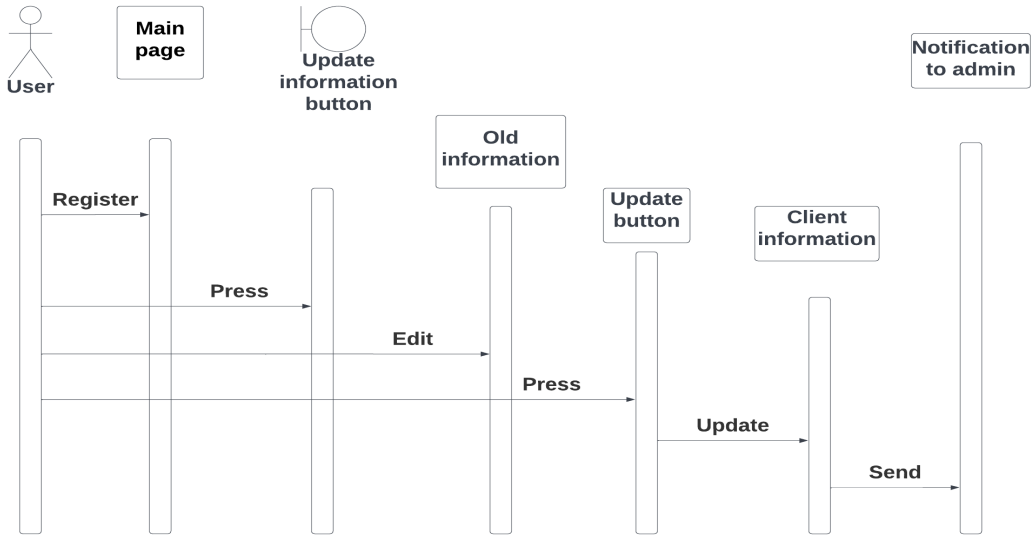


Figure (16): Update information

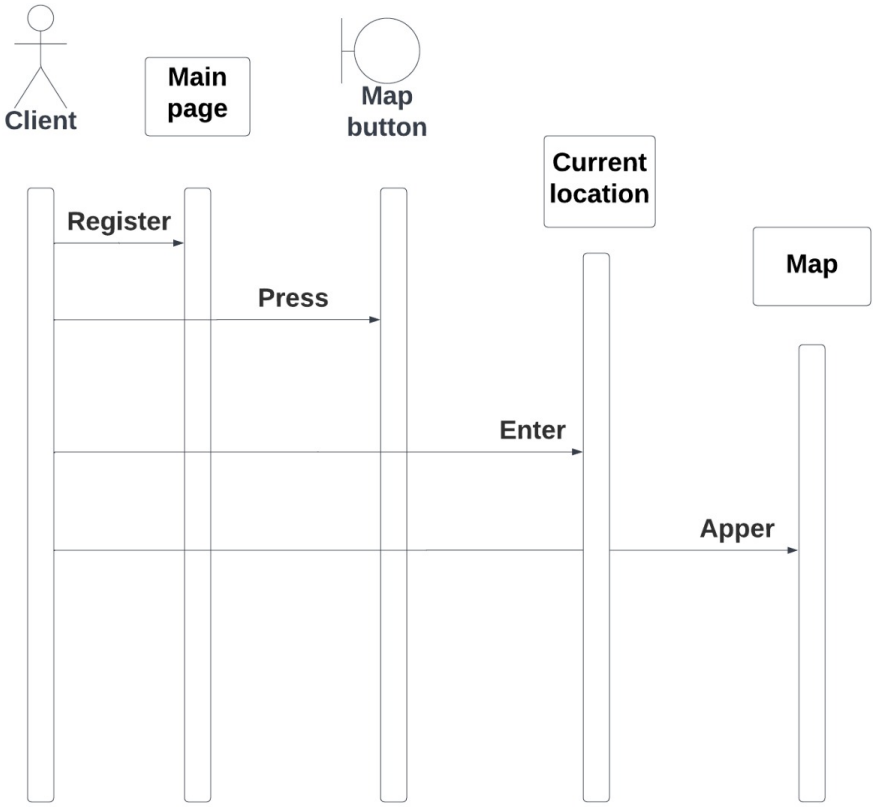


Figure (17): Map

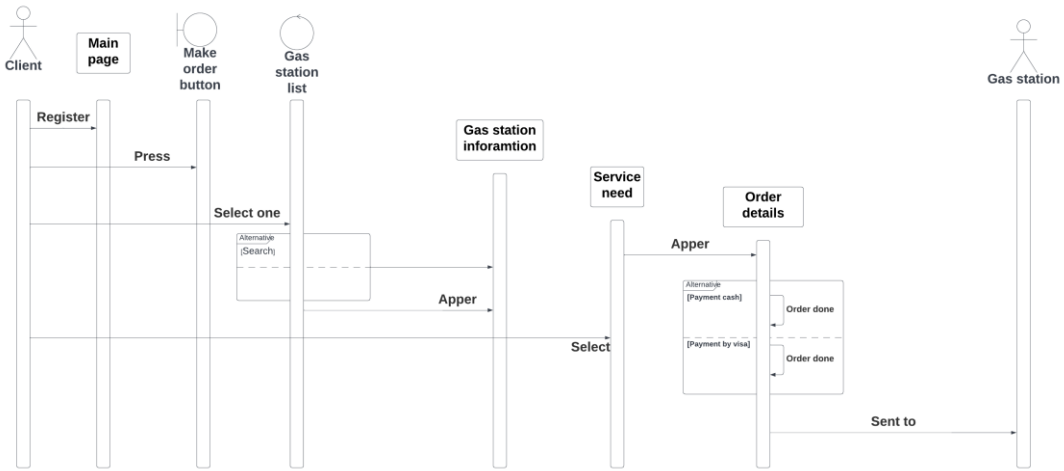


Figure (18): Make Order

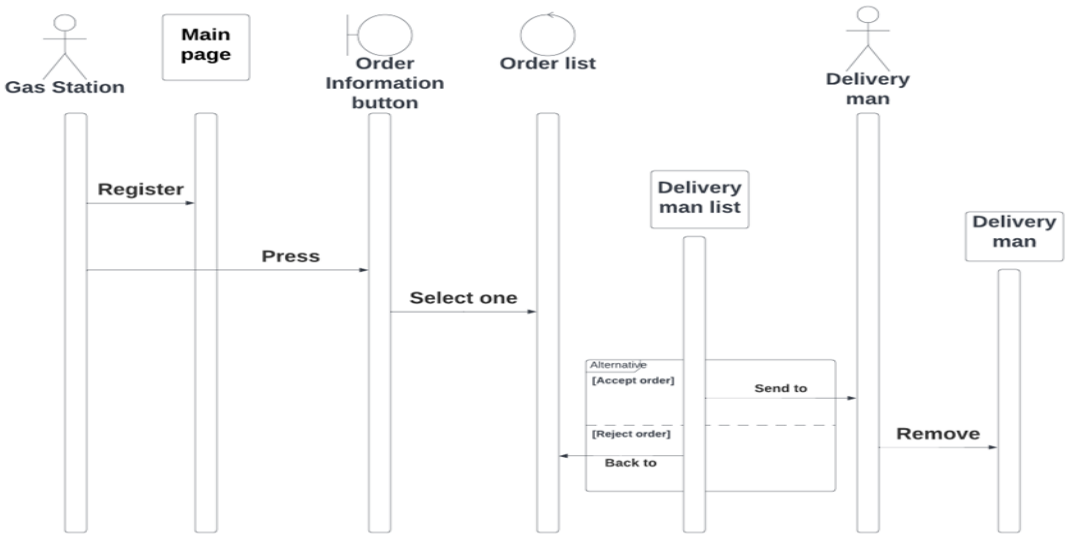


Figure (19): Accept order

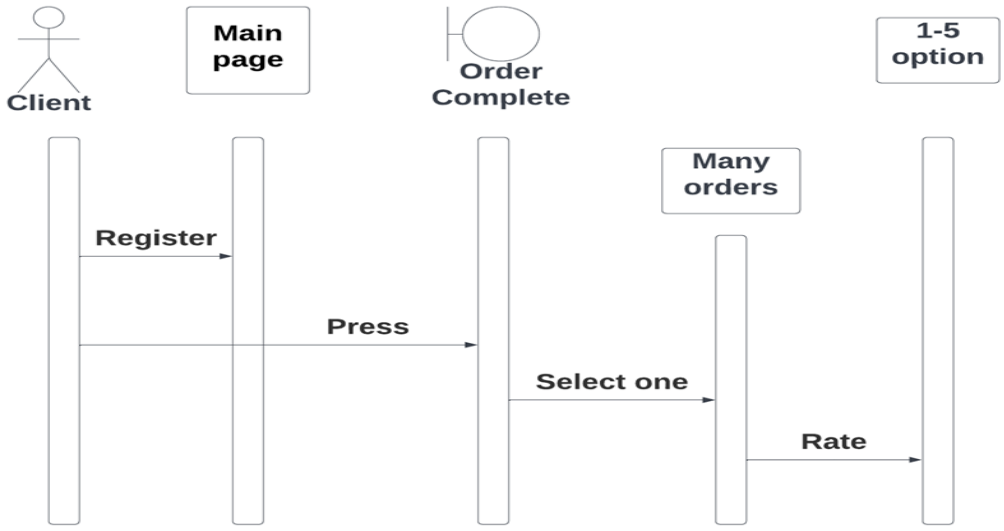


Figure (20): Rate Service

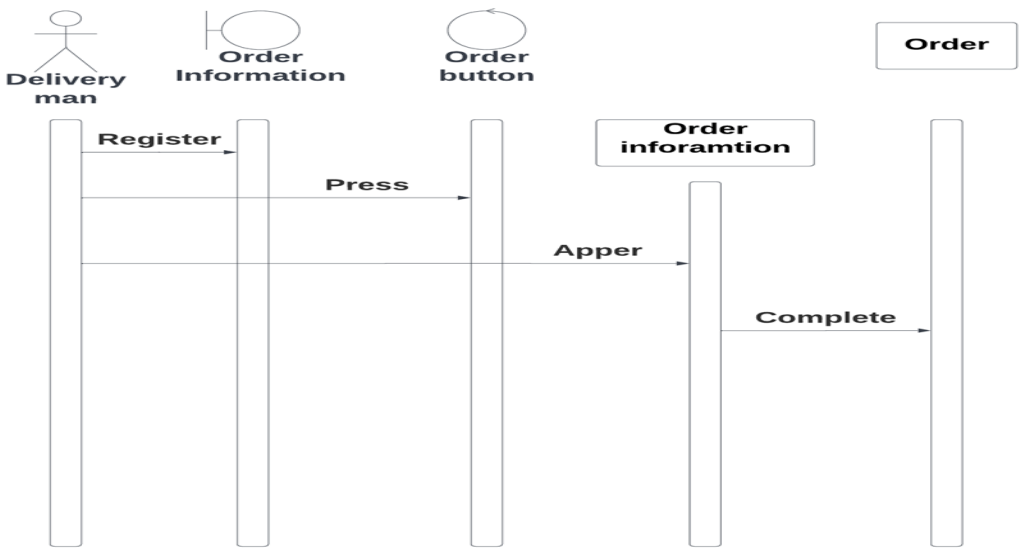


Figure (21): Confirm order

4.4 User interface design (prototype)

Main Page Screens:

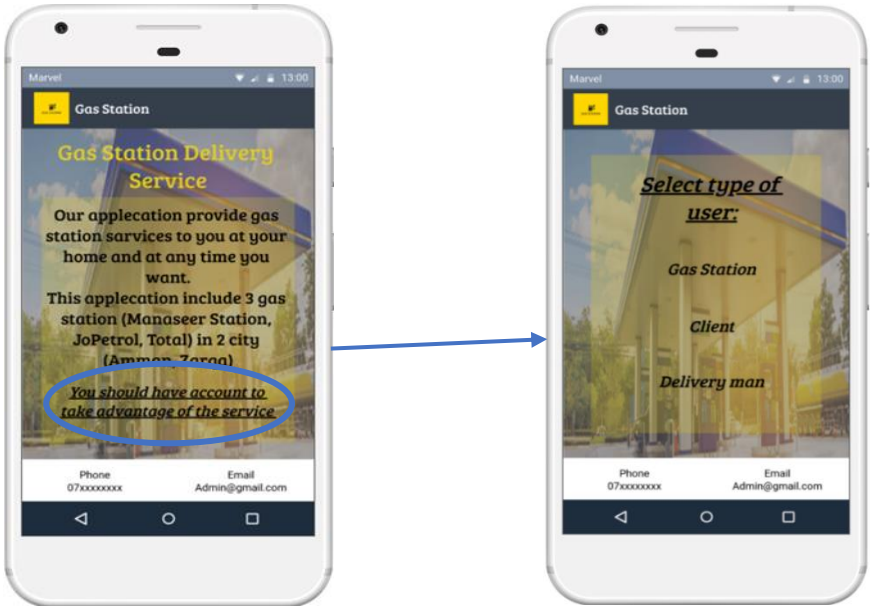


Figure (22): Main page

Figure (23): Select type of user

Gas Station Screens:

Accept orders:

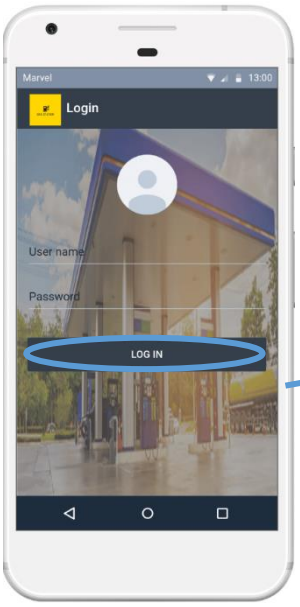


Figure (24): Login

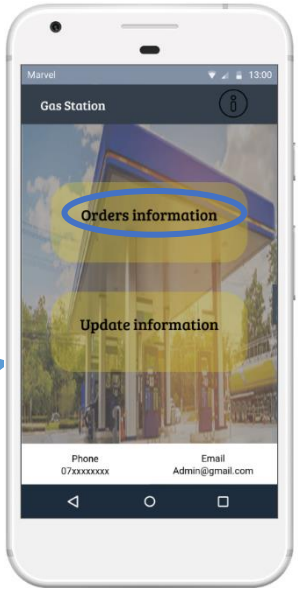


Figure (25): Home page

Figure (26): Orders information

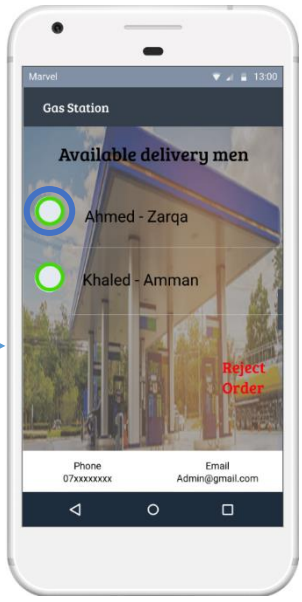


Figure (27): Available delivery men

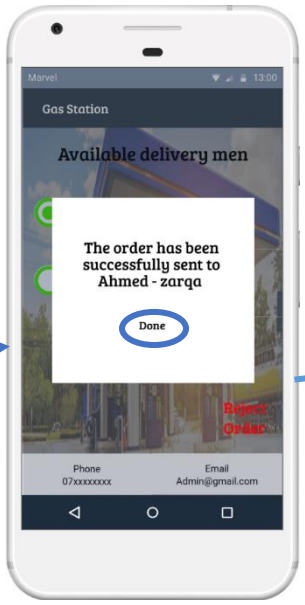


Figure (28): Send order dialog

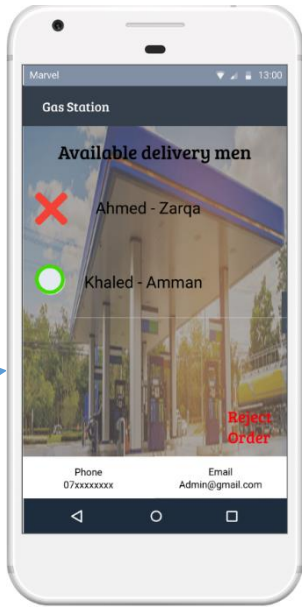


Figure (29): Remove delivery man

Reject order:

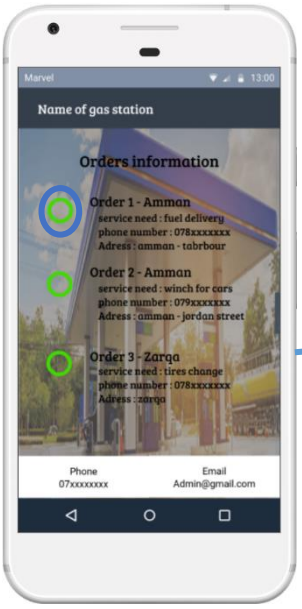


Figure (30): order information

Figure (31): Reject message

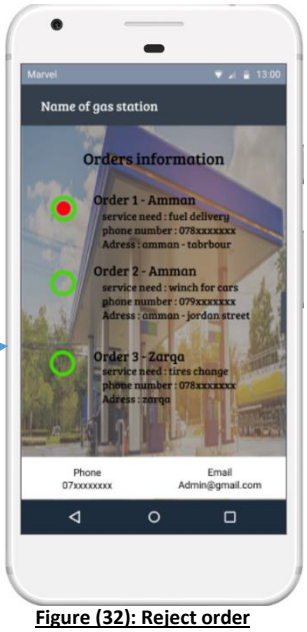


Figure (32): Reject order

Update information:

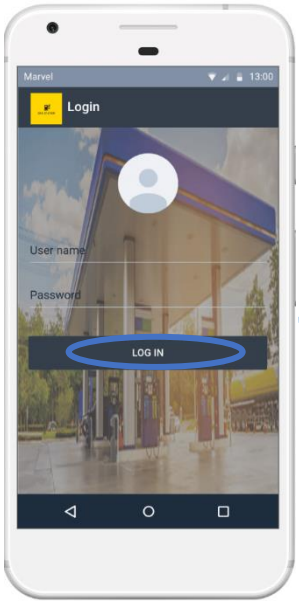


Figure (33): Login

Figure (34): Home page

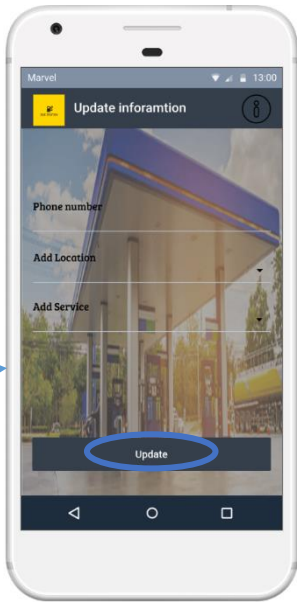


Figure (35): Update information

Invalid registration:

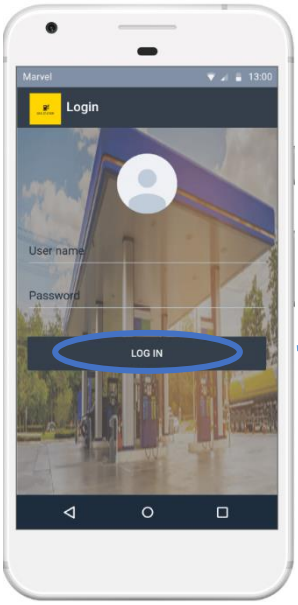


Figure (36): Login

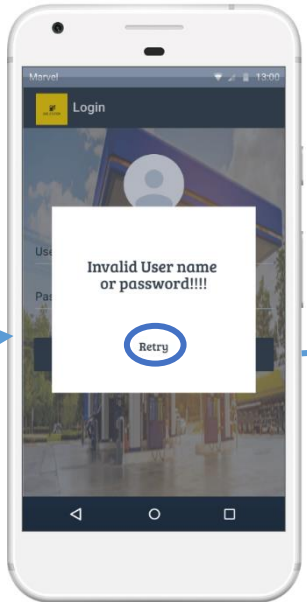


Figure (37): Invalid Registration

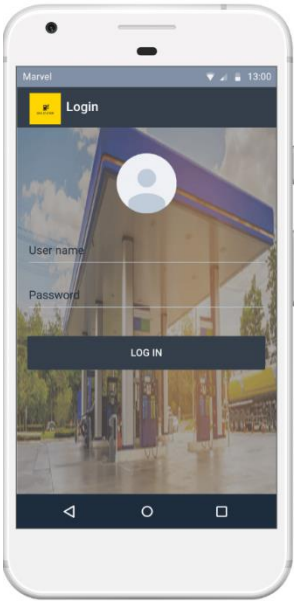


Figure (38): Login

Delivery man:

Orders information:

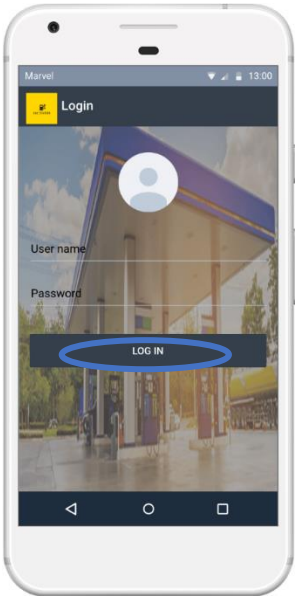


Figure (39): Login

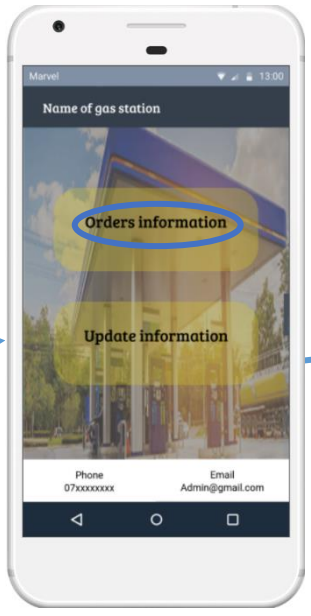


Figure (40): Home page

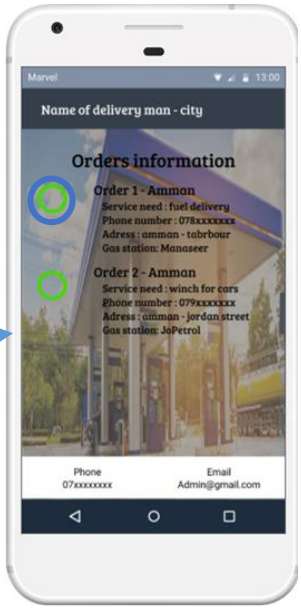
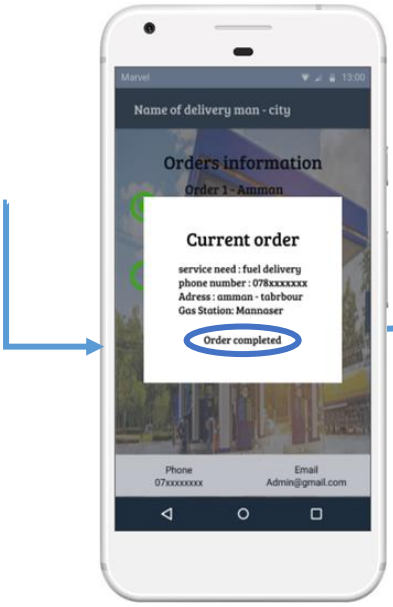


Figure (41): Order information



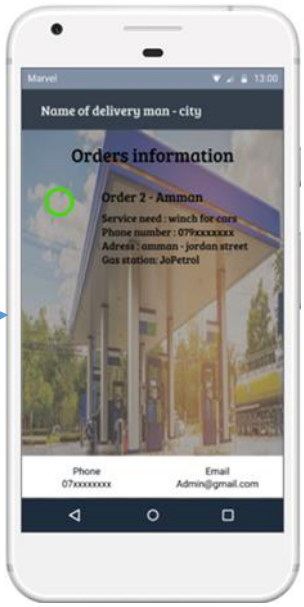
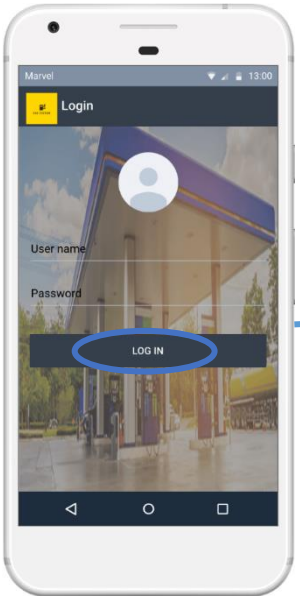
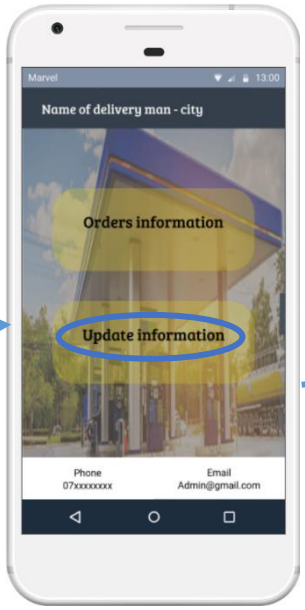


Figure (42): Current order information **Figure (43): Remove completed order**

Update information:





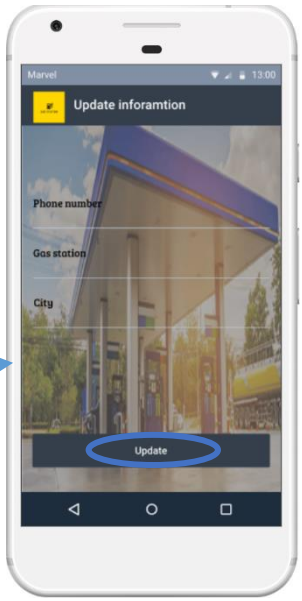


Figure (44): Login **Figure (45): Home page** **Figure (46): Update information**

Invalid registration:

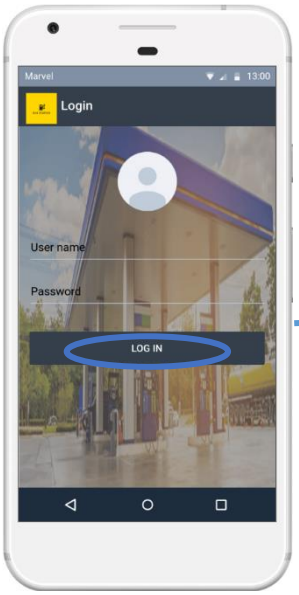


Figure (47): Login

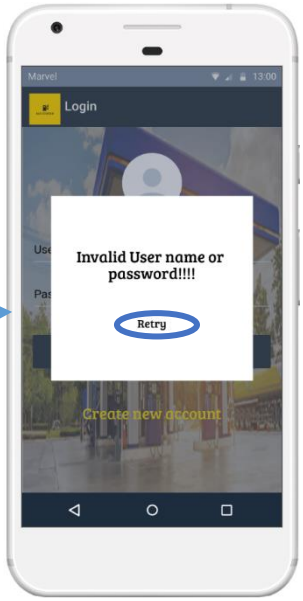


Figure (48): Invalid delivery man

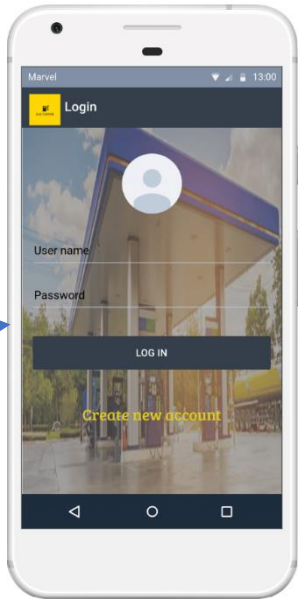


Figure (49): Login

Clients:

Create account:

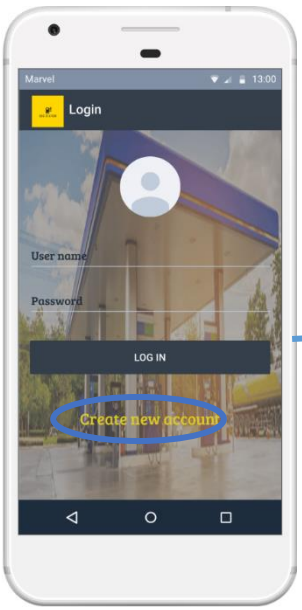


Figure (50): Login

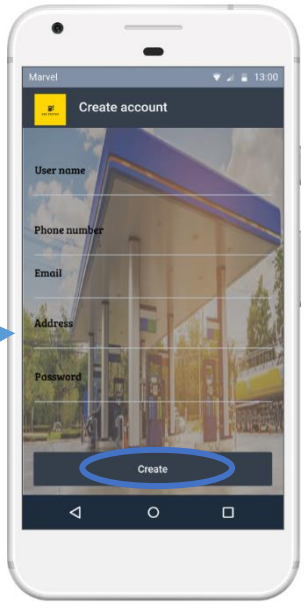


Figure (51): Create account

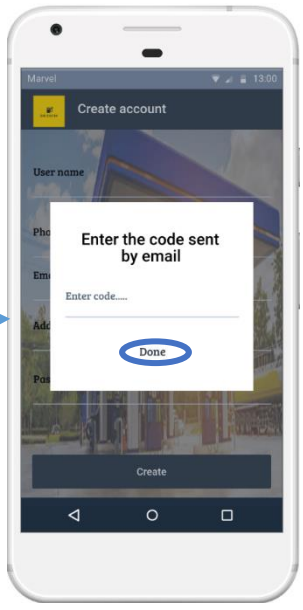


Figure (52): Enter the code

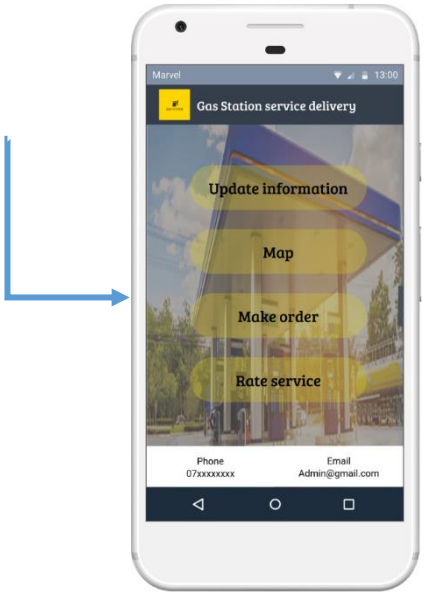


Figure (53): Home Page

Account creation error:

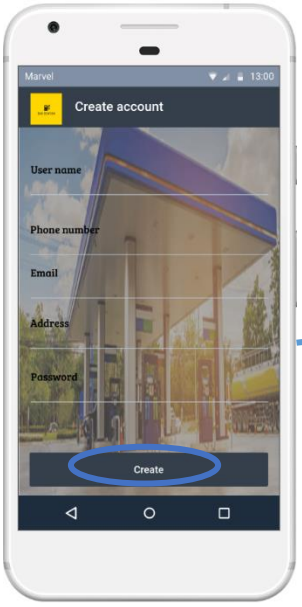


Figure (54): Create account

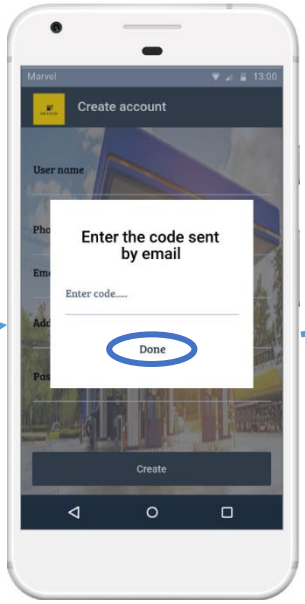


Figure (55): Enter the code

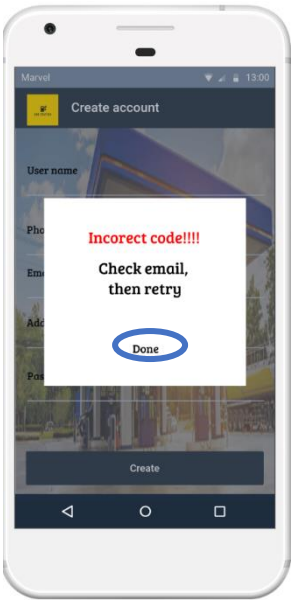


Figure (56): Error message

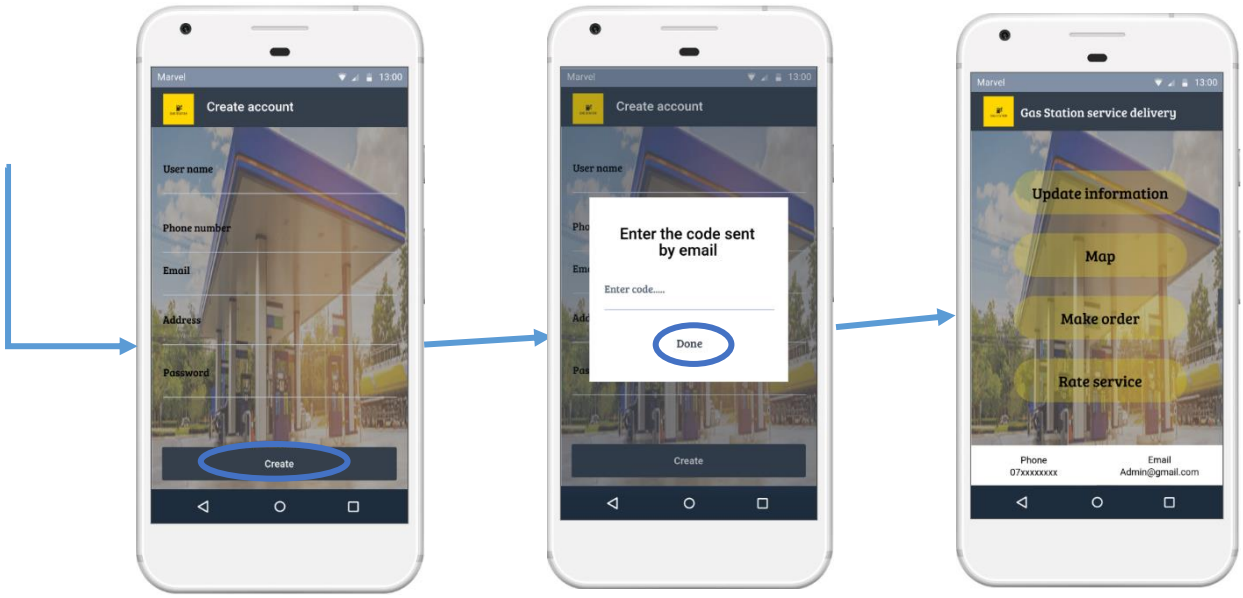


Figure (57): Create account

Figure (58): Reenter code

Figure (59): Home Page

Update information:

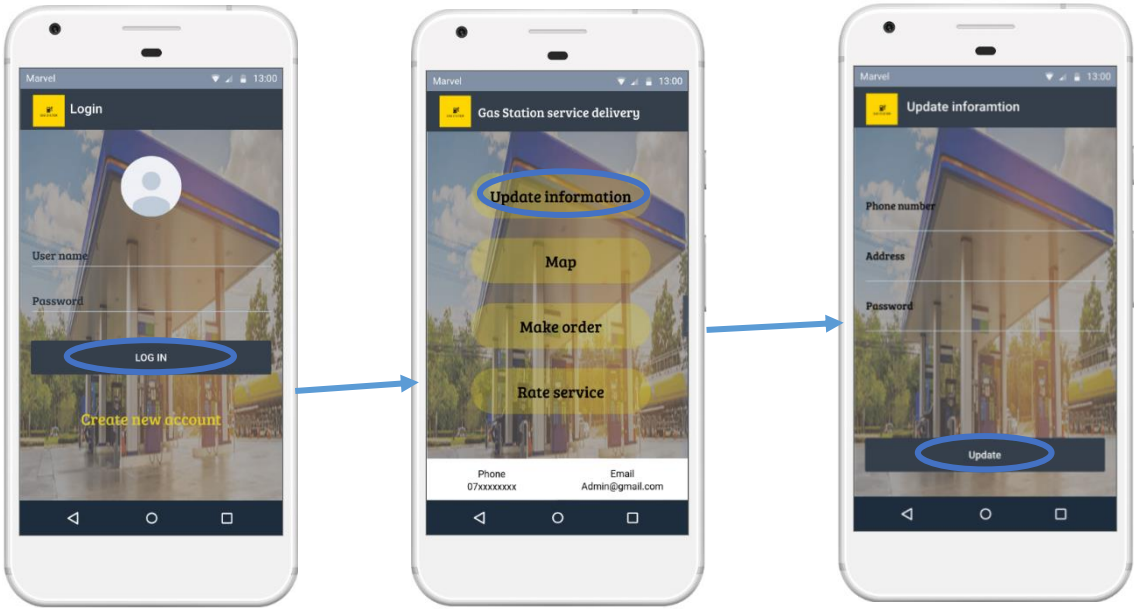


Figure (60): Login

Figure (61): Home Page

Figure (62): Update page

Using map:

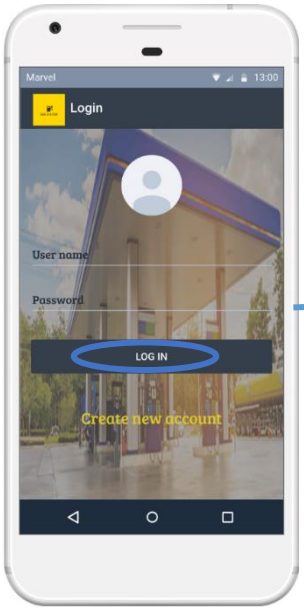


Figure (63): Login

Figure (64): Home Page

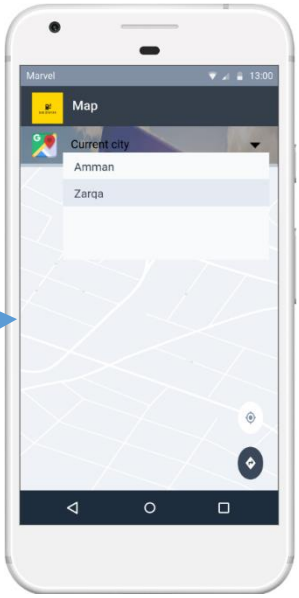


Figure (65): Nearest gas station by map

Make order (using list):

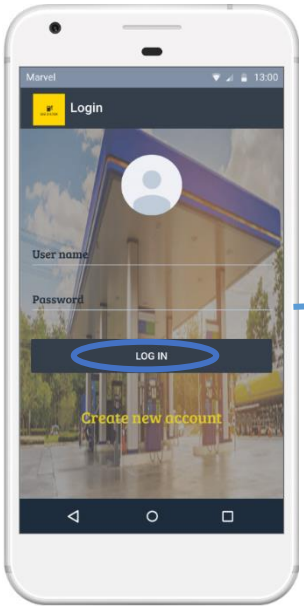


Figure (66): Login

Figure (67): Home Page

Figure (68): Select gas station from list

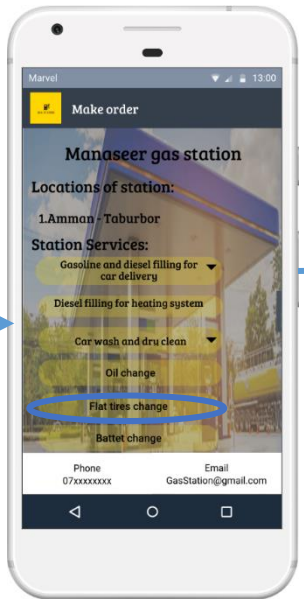


Figure (69): Select service

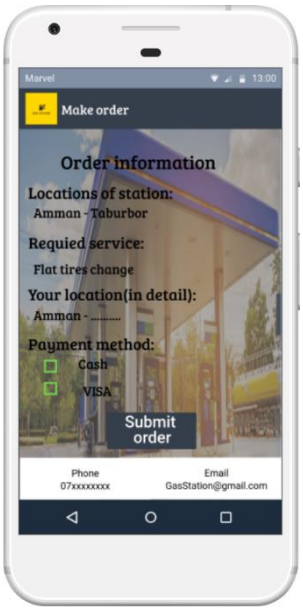


Figure (70): Order information

Make order (using search):

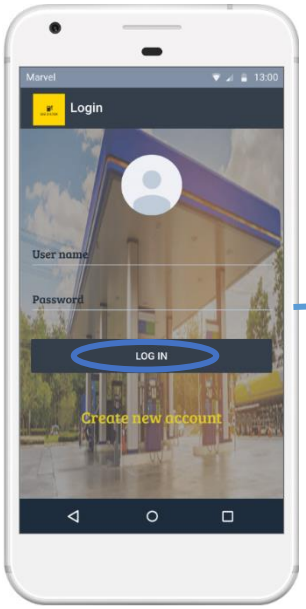
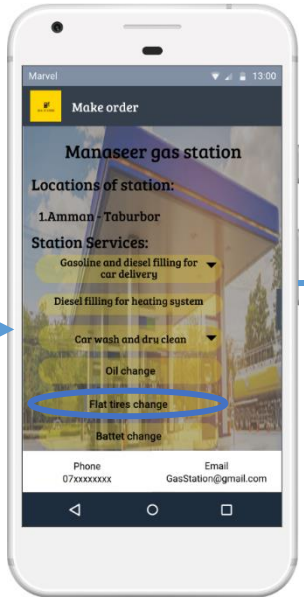


Figure (71): Login

Figure (72): Home Page

Figure (73): Select gas station by search



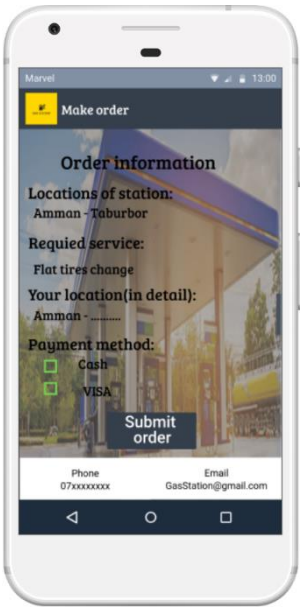


Figure (74): Select service

Figure (75): Order information

Cash payment:

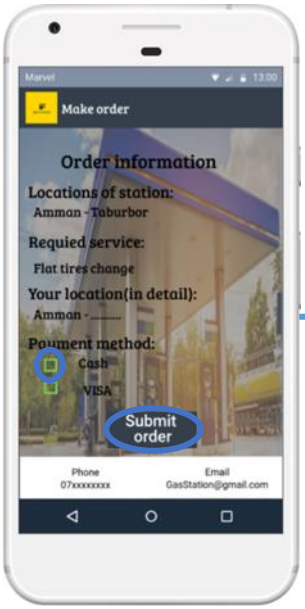


Figure (76): Order information

Figure (77): Sent order

Card payment:

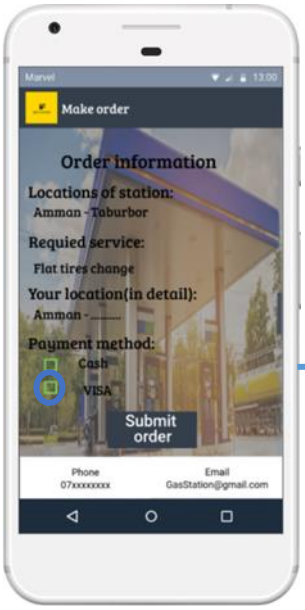


Figure (78): Order information

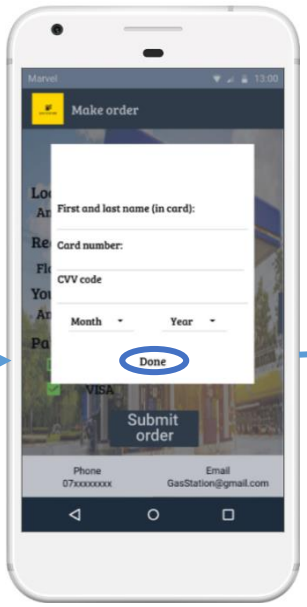


Figure (79): VISA information

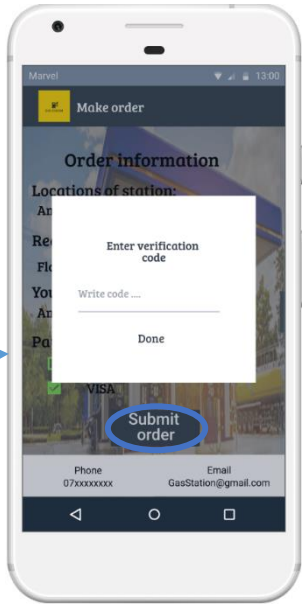


Figure (80): Enter verification code

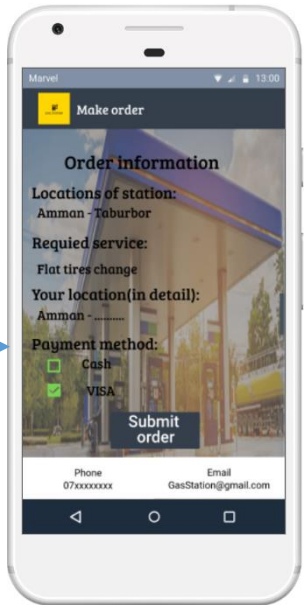
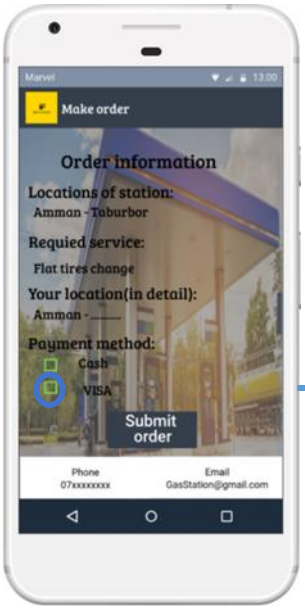
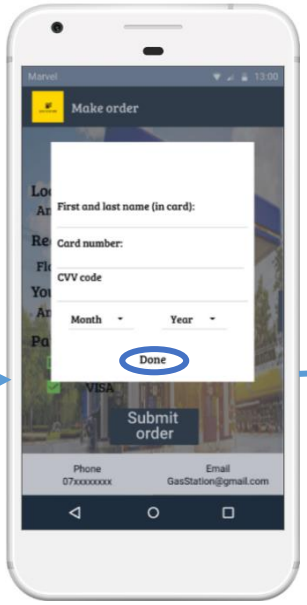


Figure (81): Order information

Figure (82): Send order

Card payment error:





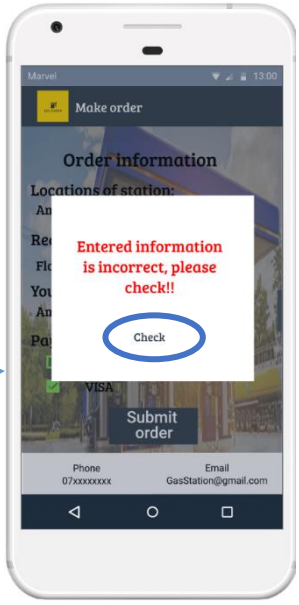
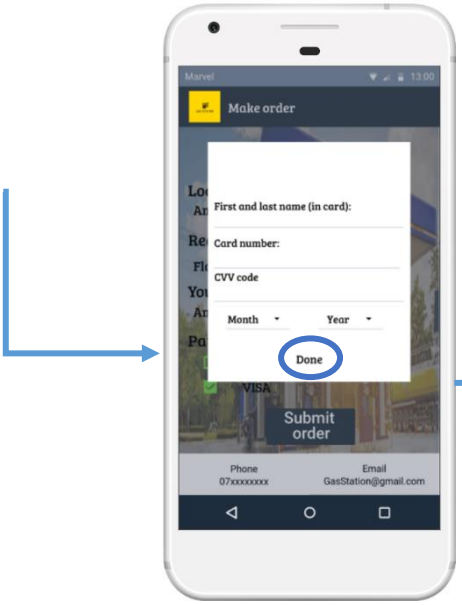


Figure (83): Order information

Figure (84): VISA information

Figure (85): Error in VISA information



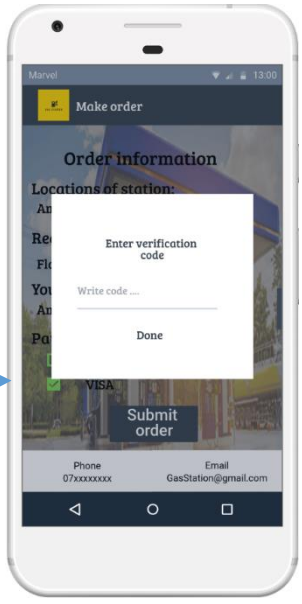


Figure (86): VISA information

Figure (87): Reenter verification code

Rating:

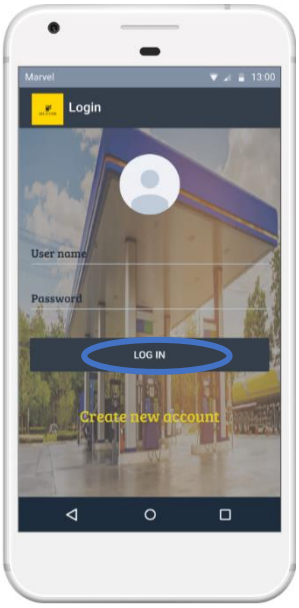


Figure (88): Login

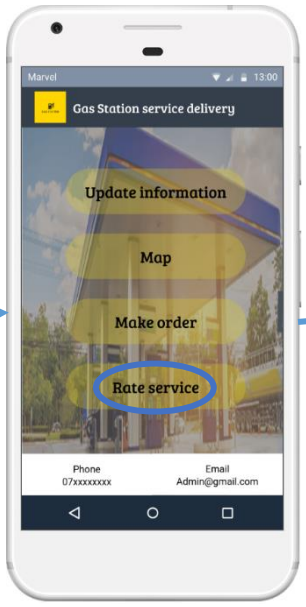


Figure (89): Home page

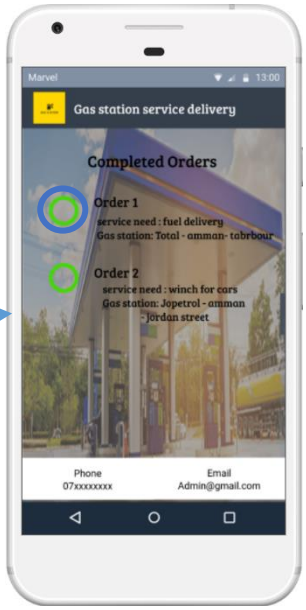


Figure (90): Complete order

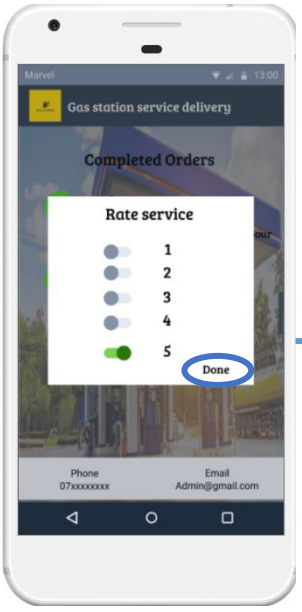


Figure (91): Rate dialog

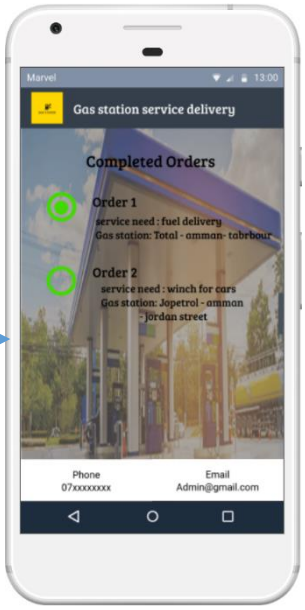


Figure (92): Ratting order

Invalid registration:

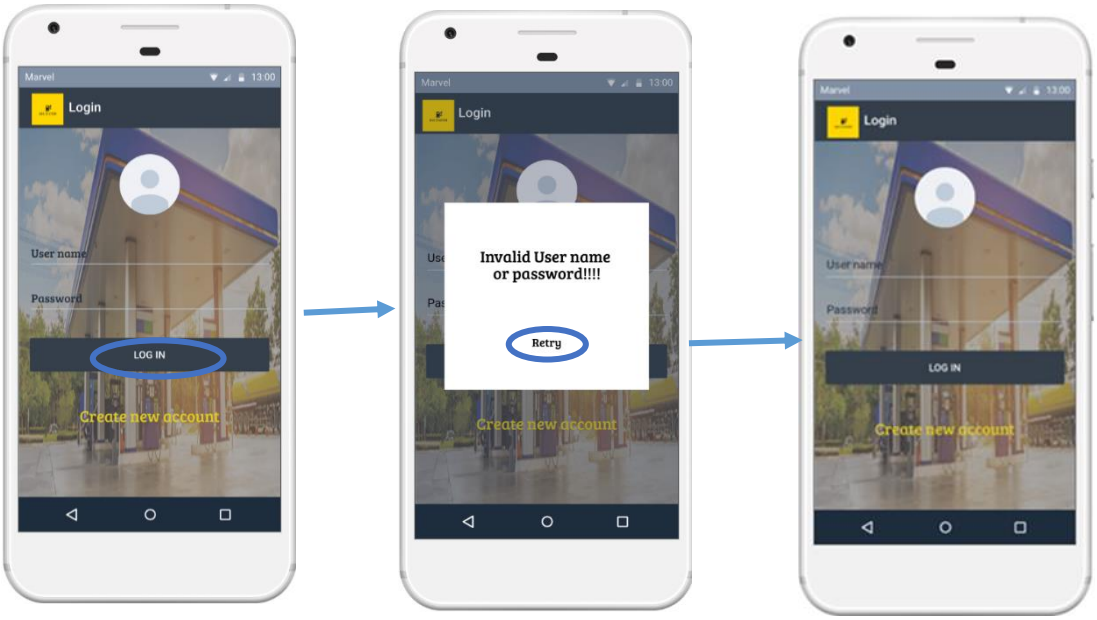


Figure (93): Login Figure (94): Invalid Registration Figure (95): Login

Admin:

Add gas station:

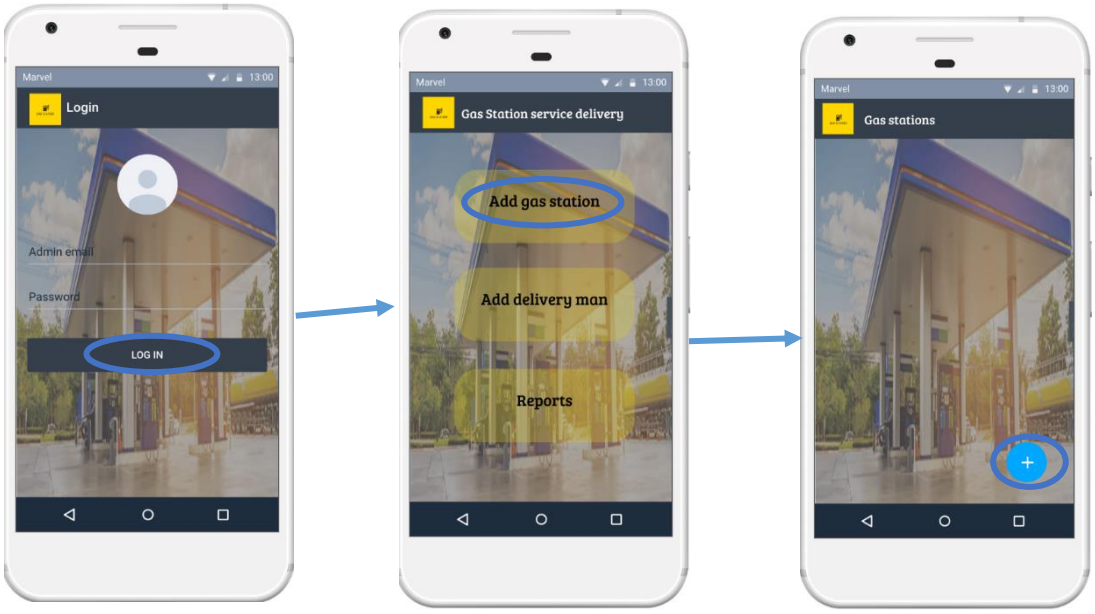


Figure (96): Login Figure (97): Main Page Figure (98): Gas stations

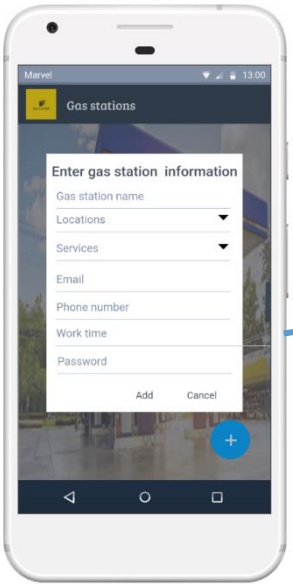


Figure (99): Enter gas station information

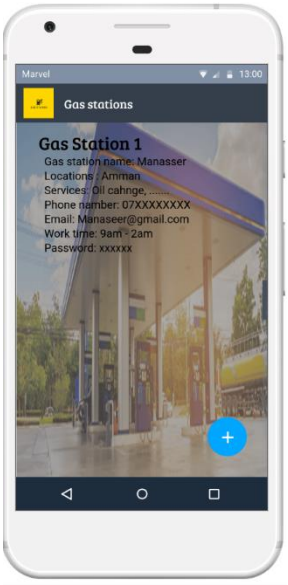


Figure (100): Gas station added

Add delivery man:

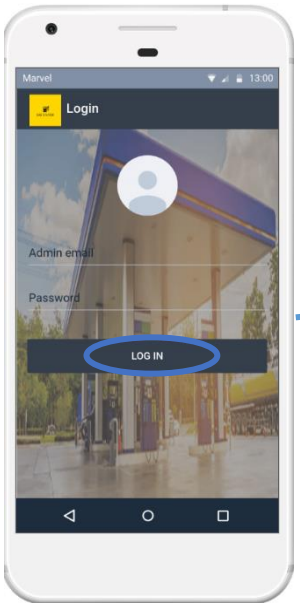


Figure (101): Login

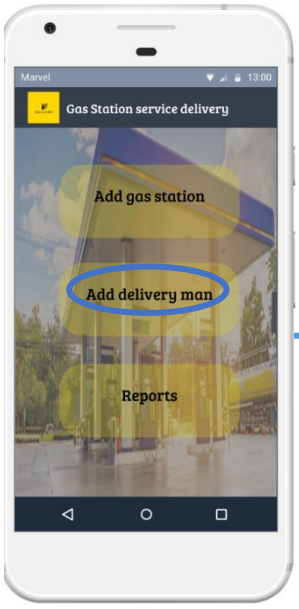


Figure (102): Main Page

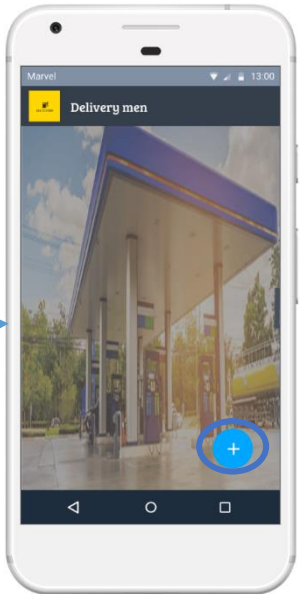
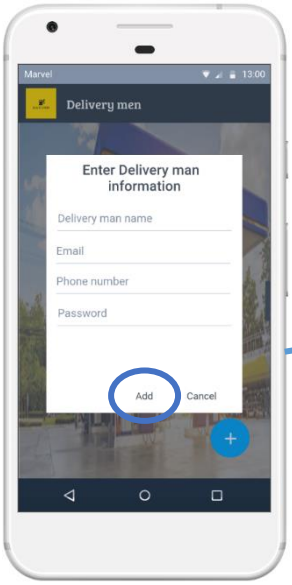


Figure (103): Delivery men



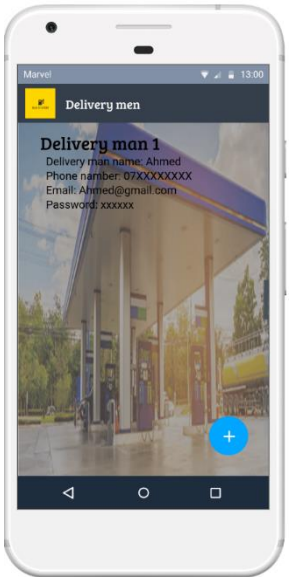
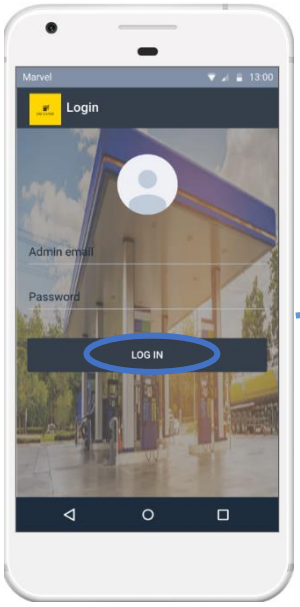


Figure (104): Enter delivery man information

Figure (105): Delivery men added

Reports:



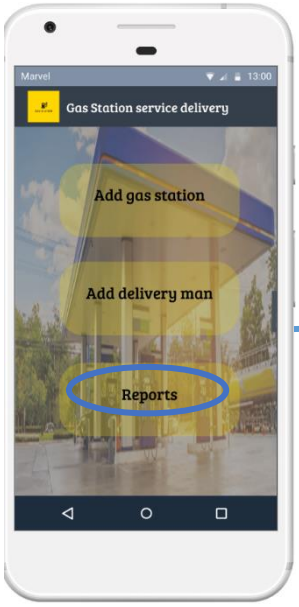


Figure (106): Login

Figure (107): Main Page

Figure (108): Reports

CAPTER 5: IMPLEMENTATION PLAN

5.1 Description of Implementation

Our purpose of our project is to enable gas station to provide and deliver service to clients in best and faster way in their location.

Each user should install app in their mobile to get benefit according to user type, then:

For the client, to get benefit from the service, they should enter all needed information whether during account creation (Username, Phone number, Email, Address, Password) or during login (Username, Password).

For the station, must also enter the information whether during account creation (Gas station name, Phone number, Email, Work time, Locations, Service provided, Password) or during login (Gas station name, Password), to be able to provide their services.

For delivery man, must also enter the information whether during account creation (Delivery man name, Phone number, Password) or during login (Delivery man name, Password), so that the station can add and send request for him.

To implementation this app, we use **Hardware** (Computer, laptop), **Software** (Visual studio code, Android studio, SQLite, Marvel, Lucid). **Programming language** (Dart, Flutter).

5.2 Programming language and technology

This section provides a list of programing languages, technologies, software, and databases required to support the implementation:

- **Programming language:**

1) Dart language:
   A programming language used for program mobile app and web app. It is an OOP language (contain objects (attribute, method)). It used to program operations (arithmetic, logical), conditional statement, loops, arrays.

2) Flutter framework:

   A software user interface **(UI)** written in dart language. It used to develop cross platform application (software application that are compatible with multiple mobile operation system).

- **Software:**

1. Visual studio code:

   We use it to code our program whether in dart language or flutter framework.
2. Android studio:

   We use it to find out how the app will appear in mobile, by create emulator to show how the app will appear in mobile.
3. SQLite:

   Is an opensource SQL database using in android applications.
4. Marvel:

   Online software we use it to design prototype for our application.
5. Lucid:

   Online software we use it create all diagrams.

CHAPTER 6: TESTING PLAN

6.1 Black-box

Project Name:		Gas station system						
Module Name:		Login				www.SoftwareTestingMaterial.com		
TEST CASE ID	TEST SCENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	RESULT	CONDITION	RESULT	(PASS/
TC_LOGIN_001	Verify the login of account	Enter valid User Name and valid Password	1. Need avalid Account to do login	1. Enter User Name 2. Enter Password 3. Click "Login" button	Successful login	Home page is shown		
TC_LOGIN_001	Verify the login of account	Enter invalid User Name or invalid Password	1. Need a valid Account to do login	1. Enter User Name 2. Enter Password 3. Click "Login" button	A message "Invalid user name or password" is shown			
TC_LOGIN_001	Verify the login of account	Enter invalid User Name and invalid Password	1. Need a valid Account to do login	1. Enter User Name 2. Enter Password 3. Click "Login" button	A message "Invalid user name or password" is shown			

Sample Test Case

Figure (109): Login test case

Project Name:		Gas station system						
Module Name:		Vrification user			www.SoftwareTestingMaterial.com			
TEST CASE ID	TEST SCENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS (PASS/ FAIL)
TC_VARIFICATION_003	User varification	Enter valid varification code	1. Need a valid code to create account	1. Enter code 2. click"send to admin"	Successful signup	Home page is shown		
TC_VARIFICATION_003	User varification	Enter invalid varification code	1. Need a valid code to create account	1.Enter code 2.click"send to admin"	A massage "Incorrect code!" is shown			

Sample Test Case

Figure (110): Verification user text case

Project Name:	Gas station system							
Module Name:	Payment by cridit							
TEST CASE ID	TEST SCENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS (PASS/ FAIL)
TC_Payment _002	Verify the information of visa	Enter valid User Name,valid card number,valid cvv code and valid expired date	1. Need a valid visa	1. Enter User Name 2. Enter card number 3. Enter cvv code 4.Enter expierd date 5.click"done"	Successful payment	make order screen is shown		
TC_payment_002	Verify the information of visa	Enter invalid User Name,invalid card number,invalid cvv code and invalid expired date	1. Need a valid visa	1. Enter User Name 2. Enter card number 3. Enter cvv code 4.Enter expierd date 5.click"done"	A message "entered information is incorrect, please check!" is shown			
TC_payment_002	Verify the information of visa	Enter invalid User Name /invalid card number/invalid cvv code or invalid expired date	1. Need a valid visa	1. Enter User Name 2. Enter card number 3. Enter cvv code 4.Enter expierd date 5.click"done"	A message "entered information is incorrect, please check!" is shown			

Figure (111): Payment by card test case

White Box:

Case	Actor	Input	Expected Output	Actual Output	State
Login	Gas Station	Email: total1@gmail.com Password: t@123456	Login done	Login done	Correct
Send Order to delivery man	Gas Station	Order information, Name of delivery man	Order successfully sends to delivery man	Order successfully sends to delivery man	Correct

Update information	Delivery man	New information	Information updated	Information updated	Correct
Received order	Delivery man	Select order	Order Information dialog	Order Information dialog	Correct
Make order	Client	Station name, Needed service	Order sends to select gas station with select service	Order sends to select gas station with select service	Correct
Map	Client	Current location	Appear station on the map	Appear station on the map	Correct
Add Gas station	Admin	Gas station information	Station adds with entered information	Station adds with entered information	Correct
Add delivery man	Admin	Delivery man information	Delivery man adds with entered information	Delivery man adds with entered information	Correct

CHAPTER 7: CONCLUSION AND RESULTS

7.1 Summary of accomplished project

In this semester we have worked on several things general overview of our project, requirement analysis, list the existing systems that provide service like the subject of our project, use case diagram and the system architecture (class diagram, deployment diagram, sequence diagram).

Also, we design a prototype using Marvel software, describe implementation plan we follow it in this project and describe a testing plan using black box testing. Additionally, we learned a dart language and flutter framework to use it in developing.

7.2 Future Work

1. We will develop and maintain our application to make it better and more professional Implements the online payment within the application.
2. List more products to be delivered.
3. Add more interface languages.
4. Enhancement on the UI designs.

REFERENCES

1. https://play.google.com/store/search?q=manaseer+stations&c=apps
2. https://play.google.com/store/search?q=gazdel&c=apps
3. https://lucid.app/users/login#/login
4. https://marvelapp.com/projects/my
5. https://code.visualstudio.com/docs/?dv=win
6. https://developer.android.com/studio
7. https://www.tutorialspoint.com/android/android_sqlite_database.htm
8. https://en.wikipedia.org/wiki/Non-functional_requirement