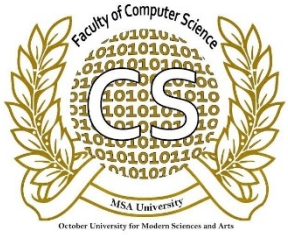


October University for Modern Science and Arts

Faculty of Computer Science

Graduation project Document

Semester 70 Spring 2020

Using dynamic vehicle sharing methodologies with pre scheduling techniques to enhance vehicles riding chances

Name: Mostafa Hesham Abdelgawad Mousa

ID: 170399

Supervised by: Dr. Moataz Samy

Abstract

Every day a university student or any employer face a problem to reach the university in the morning because it's the peak time for all of the country which traffic congestion and public transportation is at its highest, which also cause increases in the oil consumption which the world faces a bigger problem in the environment because of carbon emissions, and because public transportation are not covenant to the people and they use private cars instead and that's increases the traffic problem (W.-c. chien et al.)

So, the main idea for this project is to resolve the problem caused by traffic for the students and employers by making an application to help them communicate through and mostly to make the carpooling idea more efficient and convent by making the people pre-scheduling their trip Date and time and , creating or joining a groups to make It more efficient to find similar or known user that's are heading to the same destination in the same time and choosing between being a Driver or Passenger to make every one share his ride and cost with each other.

Table of contents

List of Figures

Chapter 1
Introduction

1 Introduction

1.1 Background

Rush hour congestion exists during every day specially at morning is affecting every big corporation or every person in daily life, delays caused by traffic can make people late for work, late for school, late for travelling schedules and staff members, that will affect any business lifecycle, also it is affecting economy and causing accidence, Also, one of the problems is the lateness caused to college student and staff, teachers …etc.

1.2 Motivation

The idea motivated from (Uber, Careem sawa ,etc.,) car sharing companies after analyzing its pros & cons and by the daily experience and problems that every college student or stuff or the car driver and the small organization face every day

- **Student perspective:** lateness causes many consequences to the student missing up lecture or an exam, causes problems with the doctors , and how the students suffer to reach their destination by very little sleeping hours and exhausting, high transportation cost for a student, Beside the fuel cost and economy problems that affect Egypt this days.
- **Car Drivers perspective**: The high number of Upper-Careem drivers(which do carpooling as part time job) that complains from going to many destination that they are not going to already, or the passenger changing his drop location, this two problems may cause a high cost and time for the car driver going back to his work/needed location from the beginning
- **Small companies/Groups:**
  1- companies that don't have private transportation to its facilities and need to help its employers reaching the company in easy way by making their own communication app so employers finding other employers that have a car going to the same facilities.
  2- any groups or friends that need a better communication tool to travel with each other to any destination if they are going to the same one.

1.3 Problem Definition

- Hundreds of thousands of students and workers that needs to reach their destination "university" at the same time using mostly the same way every day that causes congestions, every one person is using on car, instead of 3 or 4 in one car.
- Fuel high cost and Transportation problems for economy.
- Carpooling Drivers can be forced to go to any destination and lost his profit in the way back beside loosing time.

2 Project Descriptions

2.1 General Objectives

- Organize the communication between students or employers that's going to the same place
- Organize and schedule the user's trips between each other
- reduce the traffic jam during the rush-hour
- Increase cost efficiency for the car owners and passenger

2.2 Scope

- the project shall make the user that can be "captain or passenger "to choose a university destination to reach, and if the destination is the same for passenger and captain or in the way of the captain "MSA", the passenger can request a pickup from the captain.
- The system displays the users with close or similar time and destination and then choose to invite as a driver or request as a rider.
- the delivery cost appears to both captain and passenger according to the traveled distance.
- Captain can take up to 3 passengers in one ride.
- Any user that joined organization can view the user's carpooling Offers or Request that's posted by the users that have joined the same organization.

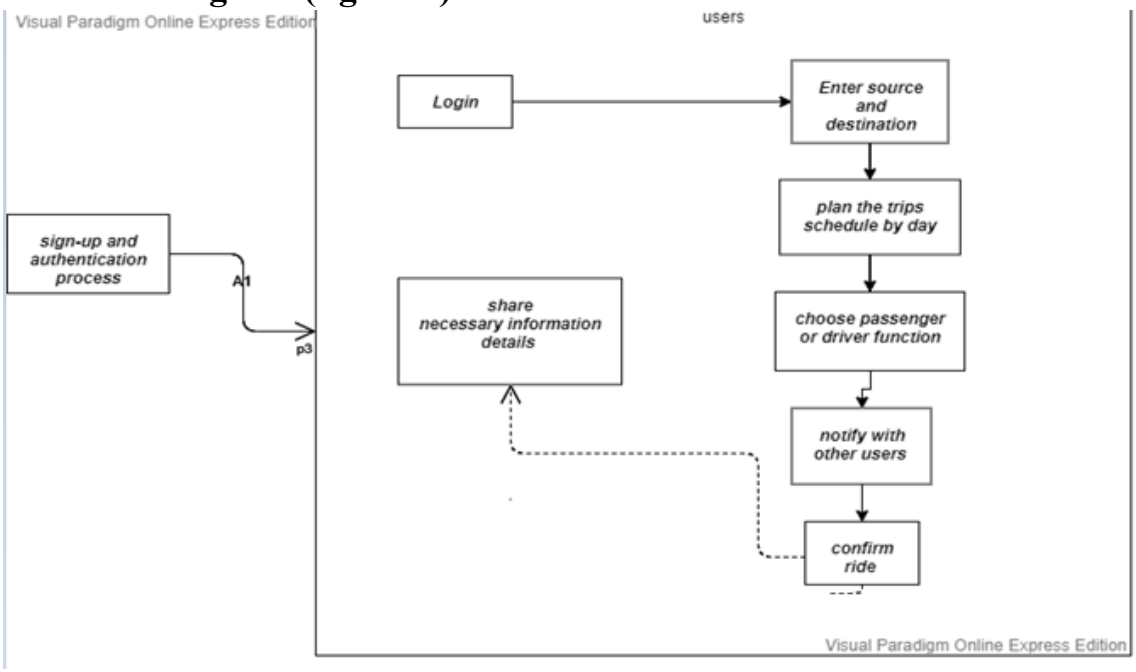
More features of the system

- Registration to an organization only done with (.edu) or facility account and Student-workID to ensure safety, and sharing thoughts with others academically users,
- Users can enter leaving home time and finish-work time for every day of the month then the system atomically will provide other user with the closest destination and moving time
- The most important that the user can make his Owen group(organization) that can be entered by other users if they have the same organization authorized email
- Every user has the ability to switch between driving mode or passenger mode

2.3 Project Overview

The system will be implemented as android mobile application

2.4 Block diagram (figure 1)



This figure 1 shows the functions parts and the connection between each part

3 Similar System Information

1. WEtransport: a context-based ride sharing platform
2. *DC2S*: a dynamic car sharing system
3. The car sharing problem by "Patrick Briest"
4. Playing with matches: vehicular mobility through analysis of trip similarity and matching
5. A Study of Uber-based Applications by "Ha Manh Tran , Sinh Van Nguyen"
6. **Smart car-pooling system** 2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC)

3.1 Similar System Description

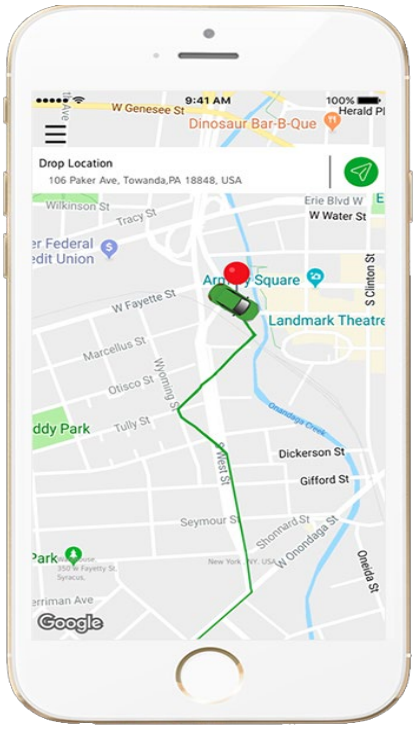
- SAWA car sharing system proposed by Careem-Uber company has been developed also to take advantage of the captains that is in the same place to ride with them but not the same going destination.
  The possible rides appear only if there are drivers in the same passenger location and going to same place, so it near impossible to find drivers when needed
- Uber bus which using a pre scheduled carpooling idea but like a public transport with only fixed destination and fixed times, and drivers and passenger not from the same university/organization.

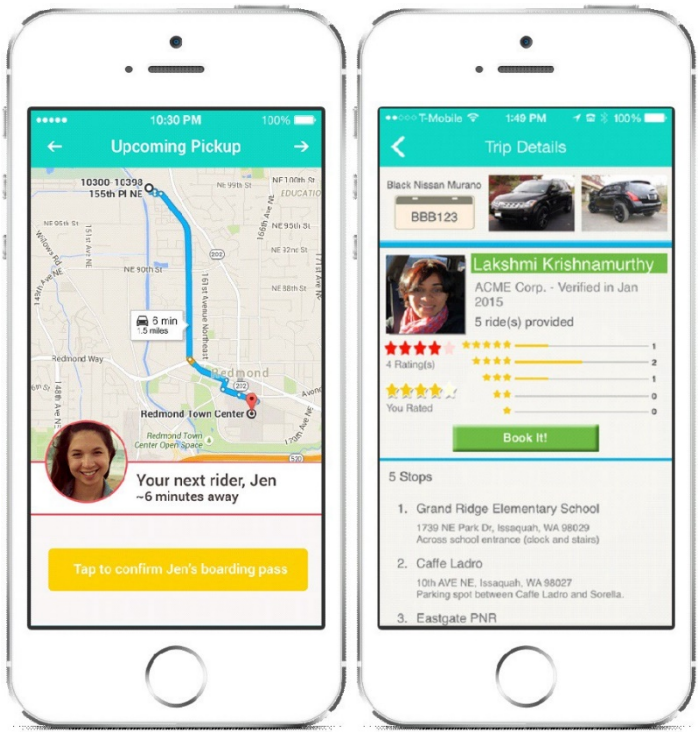
3.2 Comparison with Proposed Project "the difference between the system and other systems"

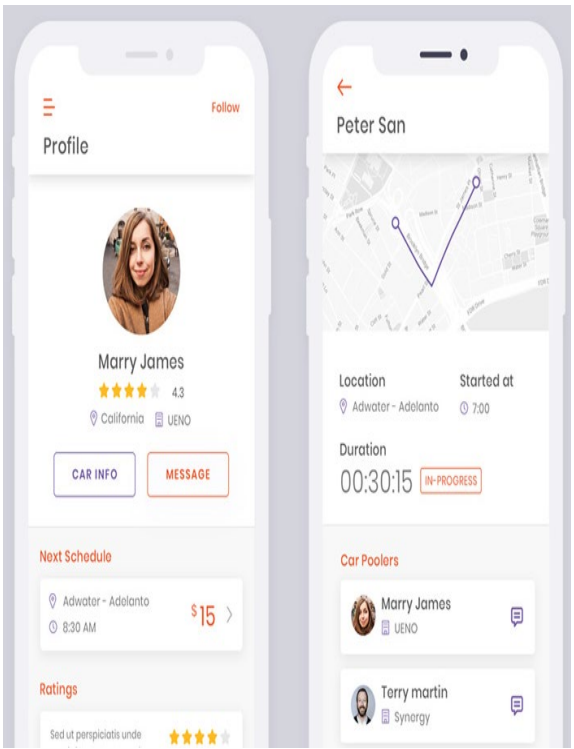
1- Our system is designed so the user can find the most matchmaking users.
2- The user can make an organization that can be joined by other users that have the same email authentication of the user (MSA.edu …. etc.) that upper and Careem doesn't allow
3- The system not designed for specific roots and not available for specific places like SAWA, its available in every location to active like google maps.
4- Uber or Careem doesn't have the functionality of creating private page between users that have the same organization or facility and creating private carpooling to be shared among them
5- The user can be the driver or passenger at any time unlike uber and Careem
6- The app allows the user schedule his ride at any time and with who, Uber and Careem only the user request the ride when he need it in a time, making in most cases only appear one driver to accept or refuse
7- uber and other carpooling system will force the captain to go in places they don't know and will cost them for driving back which not ideal for student or any one by statistics, this system if only they both need to reach the same destination
8- Uber-Careem doesn't show other users going to any destination, it only forces the driver to destination or not and force the rider not to share the ride with other users which is not time and cost efficient for students
9- The system allows the captain and passenger to choose the maximum number of passengers to ride with in the trip
10- Uber bus has only fixed time slots and fixed general locations which 's not accurate to the user needs

3.3 Screen shots from previous systems (figure 2)

(this figure 2 show the other systems interfaces which is close to what will be implemented)



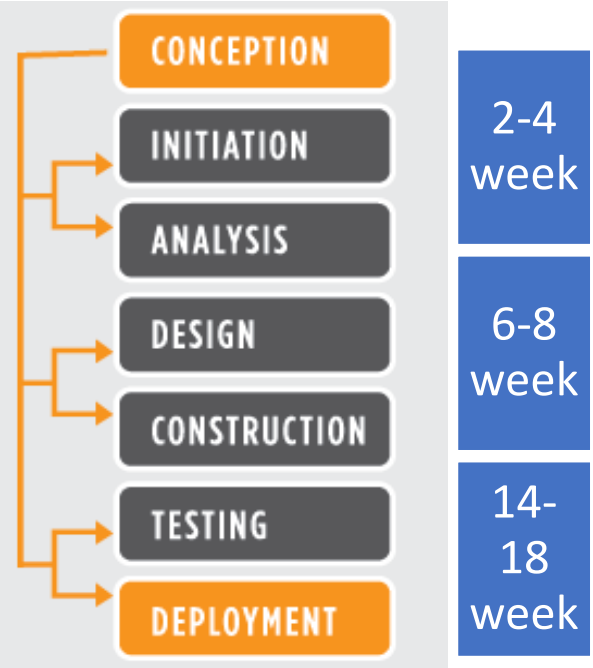




4 Project Management and Deliverable

4.1 Tasks and Time Plan

Agile method (figure 3)



(This figure 3 is to show the time plan to start and finish the project and what time it should take in each part)

4.2 Budget and Resource Costs

- Mobile phone 400-700 $
- Google Cloud console and maps 30-50 $

Chapter 2
Software Requirements Specifications

1.Introduction

1.1 Purpose of this document

The purpose of this document is to provide a detailed description about the software product requirements that's needed to effectively finish the dynamic vehicle sharing application to support the project management, this document will provide a general description of the project, a description of the functional and non-functional requirement , beside the application design and its interface, beside the scenarios and Use Case diagrams.

the documented is intended to be for both users and SE people.

1.2 Scope of this document

This requirement effort meant to describe the requirement to be effortlessly understandable by the development team and customers, the elicitation team include Mostafa Hesham beside dr Moataz Samy, and the MSA students as the users, the requirement elicitation process will spend between Jan 1,2020 to Jan 14,2020 to identify the functional ,non-functional, performance and interface, the actual time that's spent on the elicitation process will be less than 10 days.

1.3 Overview

The "dynamic vehicle sharing" application purpose is to organize and make it more simple for the users to carpool a vehicle that's is already going to the same drop destination beside the ability to be the car captain, make it more trusted and simpler to the user finding the ride by scheduling trips and find other matchmaking users, and every user will be notified with a reminders of the carpool that will take a place during the scheduled day ,beside the users can carpooling together by adding or finding an organization.

1.4 Business Context

The development of this software mobile application is sponsored by the faculty of computer science in modern science and art university, the product is concerned with the software Engineering and IEEE processes and techniques to produce well developed and designed real-world system to be used by large scale of users, and applying those processes and technique in many different actual development environments and effort,

2.General Description

2.1 Product Functions

The “dynamic vehicle sharing” app will allow the user to carpool and switch between car Captain or a rider and allowing the user finding the ride to schedule it by choosing the ride date and time plus the pickup/drop-off point as a passenger user, And as a captain that can also add the date and Arriver/Departure time and date, also source/destination location and maximum passenger to ride with.

beside the user can find and add a new organization or business facility and finding the matching users and teams, the application will also have profile for every user to view and edit.

2.2 Similar System Information

Most of the bellow systems also intendent to solve the traffic problem by carpooling from an application that the user communicates through with their own functional technique and mostly deal with a driver as an employer to the company like UBER, this application deal with the driver as a normal user that’s need to get benefit of driving every day to any destination not as an employer .

1) WEtransport: a context-based ride sharing platform
2) *DC2S*: a dynamic car sharing system
3) The car sharing problem by “Patrick Briest”
4) Playing with matches: vehicular mobility through analysis of trip similarity and matching
5) A Study of Uber-based Applications by “Ha Manh Tran , Sinh Van Nguyen”
6) **Smart car-pooling system** 2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC)

2.3 User Characteristics

the dynamic vehicle sharing application is intended to be used by any university users “students or staff” or any organization facility users’ “employers” that should have some experience using mobile application and basic knowledge of any mobile operating systems as will a having the authentication of university or organization email.

2.4 User Problem Statement

Despite there are many carpooling applications that's exists, there are no application that's designed specifically to organize and schedule trips between users in any university or organization beside giving the user the ability to be a car captain or passenger, and adding their personal organizations or facilities.

2.5 User Objectives

After the user sign up to the application, the application should automatically display an easy button to choose between if he want to be a driver or a passenger and choose between the carpooling trip requests (that's has been made by a passenger) that's viewed as a posts or offer a new one (as a driver) and choosing the date and time and choose the pickup and drop of from google map, also the user want the ability to add his Owen organization and view other car poolers post details , and need to have to search for a matches to his posts.

2.6 General Constraints

The application must be running on any version of the Android mobile system, the User's device must have an internet connection and a GPS Antenna so the application be fully function

3. Functional Requirements

Function Name	Choose between a Driver or Passenger
Function ID	RQ1
Description	the application will have a user choice between being a car driver who can offer a ride, or a passenger who can request a ride, the choice should be available for all the users that's only has registered and logged into the application. This is the first page of the application after the user login
Criticality	Extreme
Technical issues	the app must use an external Web RealTime database to keep a live track for all the users that have been registered in or not, the app should store the data even if the application is restarted or terminated by the user, the app must also be able to store all the user database in a structured list to obtain an easy maintains
Risks	medium risk – the implementation of the database should be easy, but there a dependency on an internet connection to keep the database updated and live to obtain the functionality of the app.
Dependencies	None

Function Name	Make an offer post as a Diver
Function ID	RQ2
Description	after choosing the driver function, the driver should have the ability to create a carpool offer by entering the source and the destination location and the arriver/departure time and the date of the trip, or choose to accept a carpool request that has been posted by a passenger user. • Arriver time: the time that the user wants to reach his destination at. • Departure time: the time that the user wants to leave his location at. • The driver offers should appear for the driver as a post that display all the trip info • The driver should be able to specify the maximum number of passengers for his offer
Criticality	Extreme
Technical issues	the data must be stored as a Realtime Web database to keep a live track for all the users offers and request for a trip, the app should store the data even if the application is restarted or terminated by the user

	• The user must be a membership to a University or an organization to offer a carpool
Risks	medium risk- implementation is simple but the data is only generated from the user's inputs and only can be retrieved from a web service real-time database.
Dependencies	RQ1, RQ4

Function Name	Make a request post as a Passenger
Function ID	RQ3
Description	after choosing the passenger function, the passenger should have the ability to create a carpool request by entering the pickup and the destination location and the arriver/departure time and the date of the trip, or choose to accept a carpool offer that has been posted by a driver user. • Arriver time: the time that the user wants to reach his destination at. • Departure time: the time that the user wants to be Picked Up at. • The passenger request should appear for the driver as a post that display all the trip info.
Criticality	Extreme
Technical issues	**Technical issues –** the data must be stored as a Realtime Web database to keep a live track for all the users offers and request for a trip, the app should store the data even if the application is restarted or terminated by the user • The user must be a membership to a University or an organization to offer a carpool
Risks	medium risk- implementation is simple but the data is only generated from the user's inputs and only can be retrieved from a web service real-time database.
Dependencies	RQ1, RQ4

Function Name	Create a New Organization/Facility Group
Function ID	RQ4
Description	any user should have the ability to create or add a new organization/University, creating it required to enter the organization's name, description, business or university mail, Website URL (optional), and the type
Criticality	Extreme
Technical issues	user registry and inputs are required to create or add the organization, the app should store the organization data even if the application is restarted or terminated by the user • The user must be a membership to a University or an organization to offer an organization carpool.
Risks	medium risk- implementation is simple but the data is only generated from the user's inputs and only can be retrieved from a web service real-time database.
Dependencies	None

Function Name	Join a new Organization/Facility Group
Function ID	RQ5
Description	The user who only have registered for the application should be able to join any organization or university that they have its valid email credentials (email address and password. uni.edu, orange.te… etc.)
Criticality	Extreme
Technical issues	the application will automatically accept the user to enter the organization only if the email is valid.
Risks	low Risk - internet connection is required to access and validate the data.
Dependencies	None

Function Name	Sign-up/ Register
Function ID	RQ6
Description	any user should be able to register and create an account by filling in the required information bellow • First and second name • User E-mail and password • Phone number
Criticality	Extreme
Technical issues	the user data only stored external as a firebase Realtime database
Risks	low Risk – the function required an internet connection to be successfully active
Dependencies	None

Function Name	Sign-in/ login to the application
Function ID	RQ7
Description	the user must be able to sign-in by entering his registered username and password to view and access all of the application's futures
Criticality	Extreme
Technical issues	the user data only stored external as a firebase Realtime database and must have EMAIL Authentication in firebase
Risks	low Risk – the function required an internet connection to be successfully active
Dependencies	RQ8

Function Name	Sign out / logout of the application
Function ID	RQ8
Description	the user must be able to log-out from the application by pressing a button that is displayed is the user profile page that will take him to the log-in page
Criticality	Very
Technical issues	the user must be already logged in to the application.
Risks	low Risk – the function is simple to implement and doesn't require an internet connection
Dependencies	None

Function Name	Displaying the carpooling posts from the same Facility/Organization group
Function ID	RQ9
Description	the application should display the others users posts that are belong to the same organization that's a user joined • The user can view the posts from the same organization that's created by other users • If the user has joined multiple organization, he should choose which organization he wants to create or view its posts
Criticality	Very
Technical issues	the user must be already logged in to the application.
Risks	the user data must be checked by the firebase to view the contents of the organization's posts.
Dependencies	RQ2, RQ3

Function Name	Deleting post
Function ID	RQ10
Description	the users should be able to delete a carpool post that he already created before • The function should be for both driver 's post or a passenger's post
Criticality	Normal
Technical issues	the user data must be checked by the firebase to remove it.
Risks	low Risk – the function required an internet connection to be successfully active
Dependencies	RQ2, RQ3

Function Name	browse and search for a specific carpool post
Function ID	RQ11
Description	the user should be able to browse and search posts for his joined organization as a driver or as passenger and search for a specific post by • Choose the kind of the post (driver's offer, Passenger's request) • The date and time of the trip • The source and destination
Criticality	Normal
Technical issues	The user must be joined to an organization first.
Risks	low Risk – the function required an internet connection to be successfully active
Dependencies	RQ2, RQ3

Function Name	view all of the user past and present carpooling activity
Function ID	RQ12
Description	the user should be able to view all of his past and current or planned carpooling activity, and should be able to view the details (time, date, location and destination) and the other users in the carpool. • The app should view the activity either if it happens as a driver or as a passenger.
Criticality	Normal
Technical issues	• The user must be already joined any carpooling activity to have the data to be viewed.
Risks	low Risk – the function required an internet connection to be successfully active and retrieve the data from the fire base
Dependencies	RQ2, RQ3

Function Name	User search and view his location and destination using updated google-map when entering a new carpool activity.
Function ID	RQ13
Description	when the user enters a new carpooling activity (offers or requests) as a driver or as a passenger, he/she should be able to search and choose the place/destination and location/pickup point of the carpool from an integrated visual google-map API. • The app should provide the google map for both Passenger and Driver. • The app should display the desired route between the source and destination for both Passenger and Driver. • The route should be displayed when the user enters to view any post.
Criticality	Extreme
Technical issues	The app should have an updated google-maps and google-places API keys so the map always be o the last updated version.
Risks	low Risk – the function required an internet connection to be successfully active.
Dependencies	RQ2, RQ3, RQ14

Function Name	cancel any multi step or process when creating a new post
Function ID	RQ14
Description	• when the user enters to make a new carpooling post (offers or requests) as a driver or as a passenger, he/she should be able to cancel the steps at any time or any step (enter time, destination, date …etc.)
Criticality	Very
Technical issues	None
Risks	low Risk – the function simple to implement
Dependencies	None

Function Name	Editing personal information and change default organization
Function ID	RQ15
Description	• when users enter the user info page, he can edit his name and phone number • if the user joined more than one organization, the user can choose his default organization to post in
Criticality	Very
Technical issues	None
Risks	medium Risk – the function needs to change and update the firebase data
Dependencies	RQ5

4. Interface Requirements

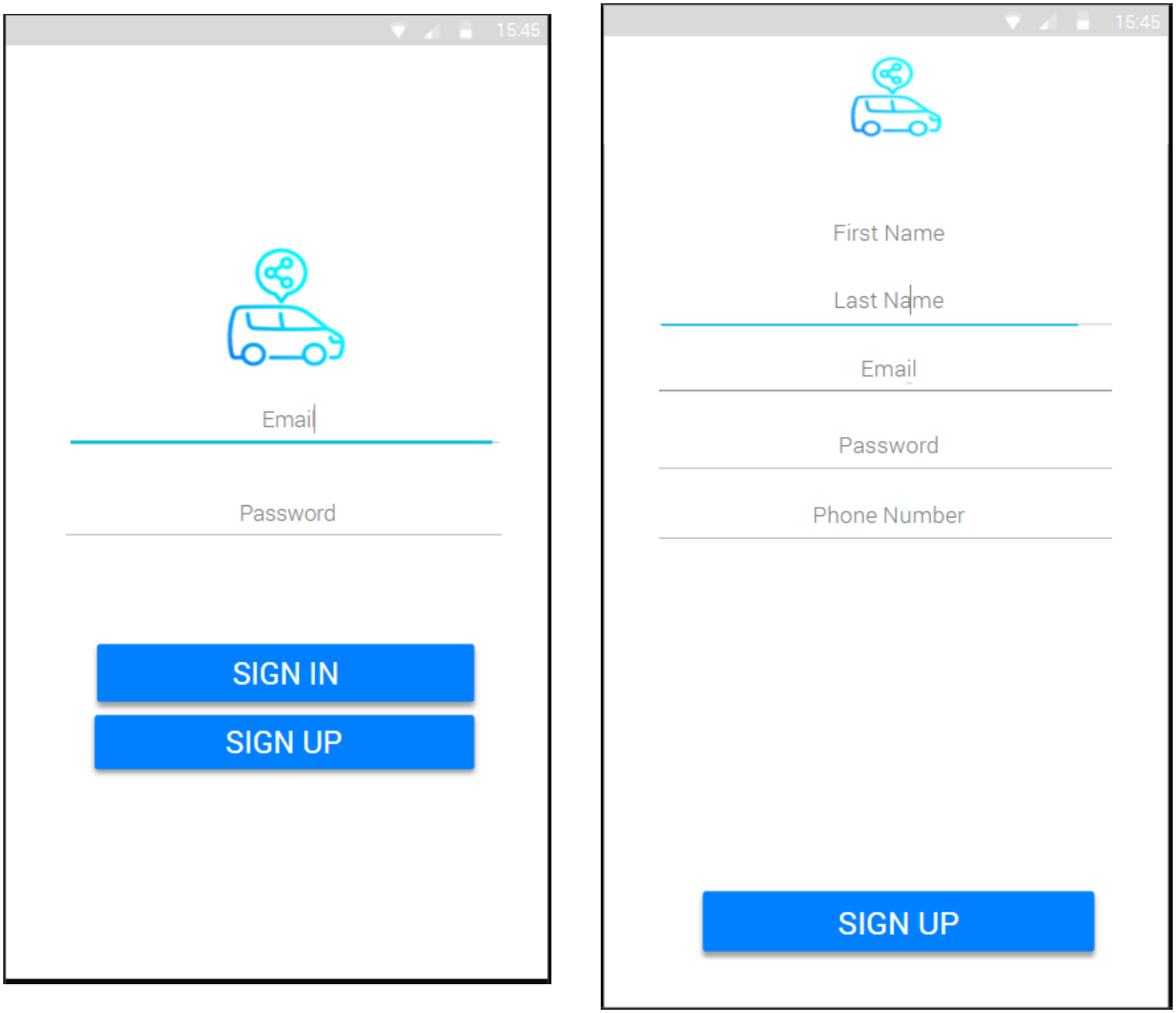
4.1 User Interfaces

The application interface should have (titles, Icons) on each page allowing the user to understand on which page they are on, the interface will have a page for choosing between a driver or passenger and by entering either of them it will show a number of steps with a help of (google-maps) to finish the process of making an offer or request, and simply join or create an organization and view all of its post, the user should also have an icon for displaying his profile and trips.

4.1.1 GUI (prototype)

(figure 4) This figure shows the prototype interface of the application

1.0 1.1



1.0 – the user enters here the user name and password to enter the application to entre home page

1.1- The user enters his user name and password /phone number …. etc. to register a new account

. 2.0

2.0- after the user login he can choose between the driver function or a passenger function

(Figure 4)

2.1 2.2

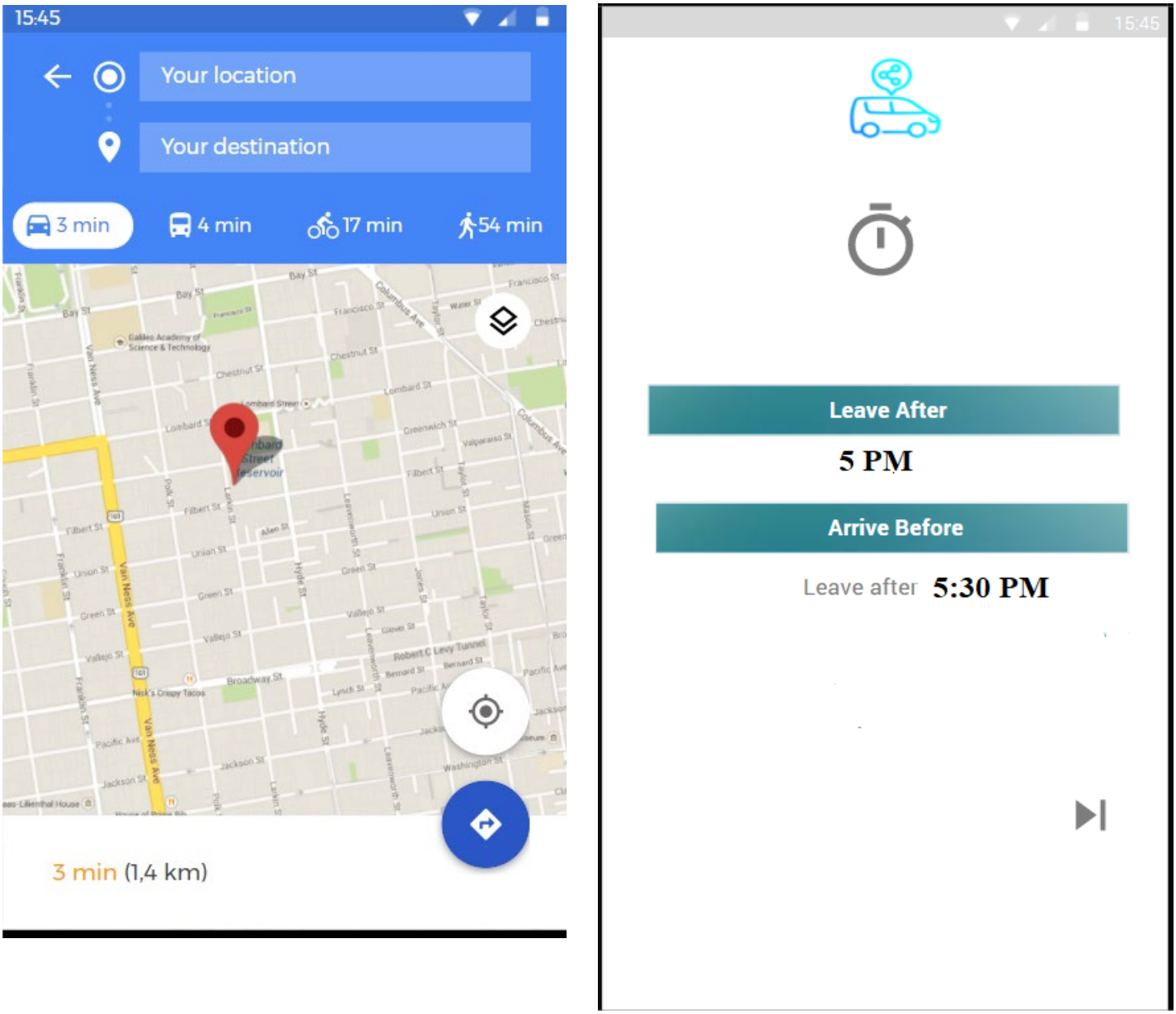
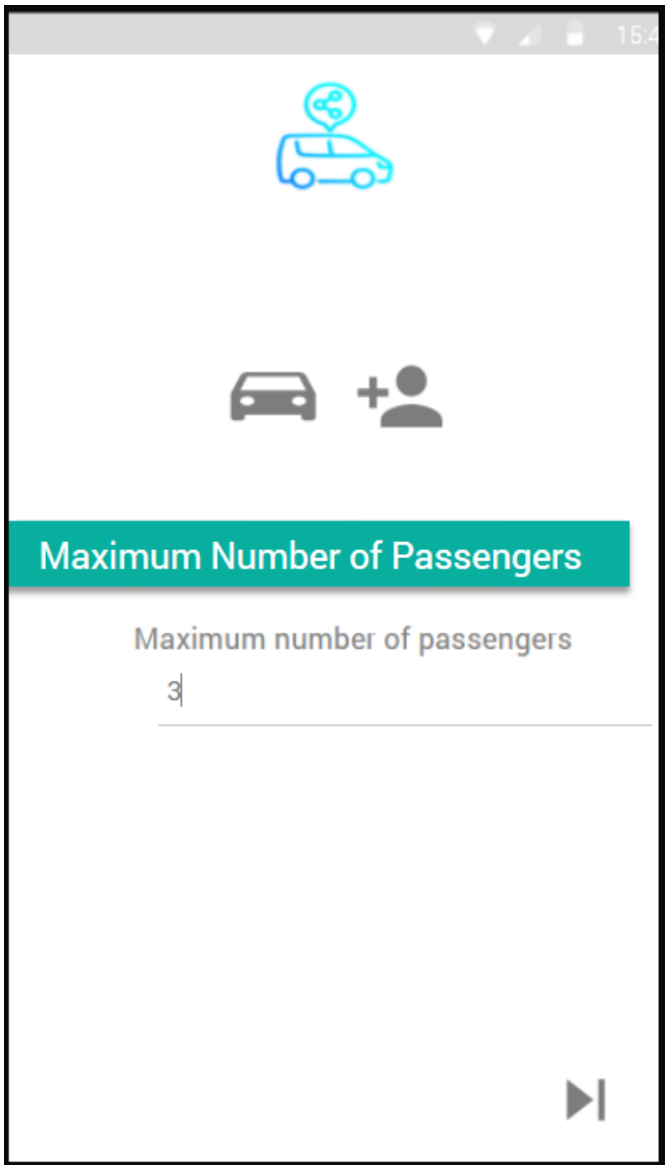


Figure 4

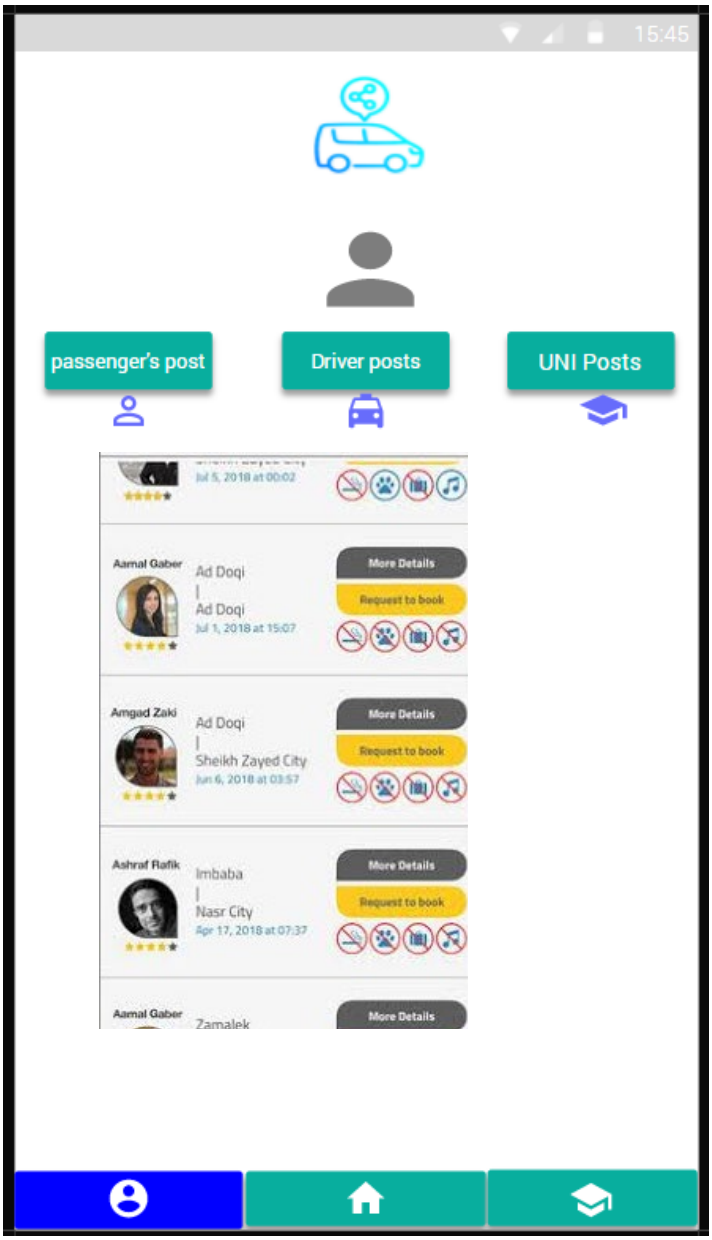
2.1- after the user choose driver or passenger, he goes to enter the location

2.2 – the user enters the leaving time and arriving time and the date of the carpooling

2.3



3.1

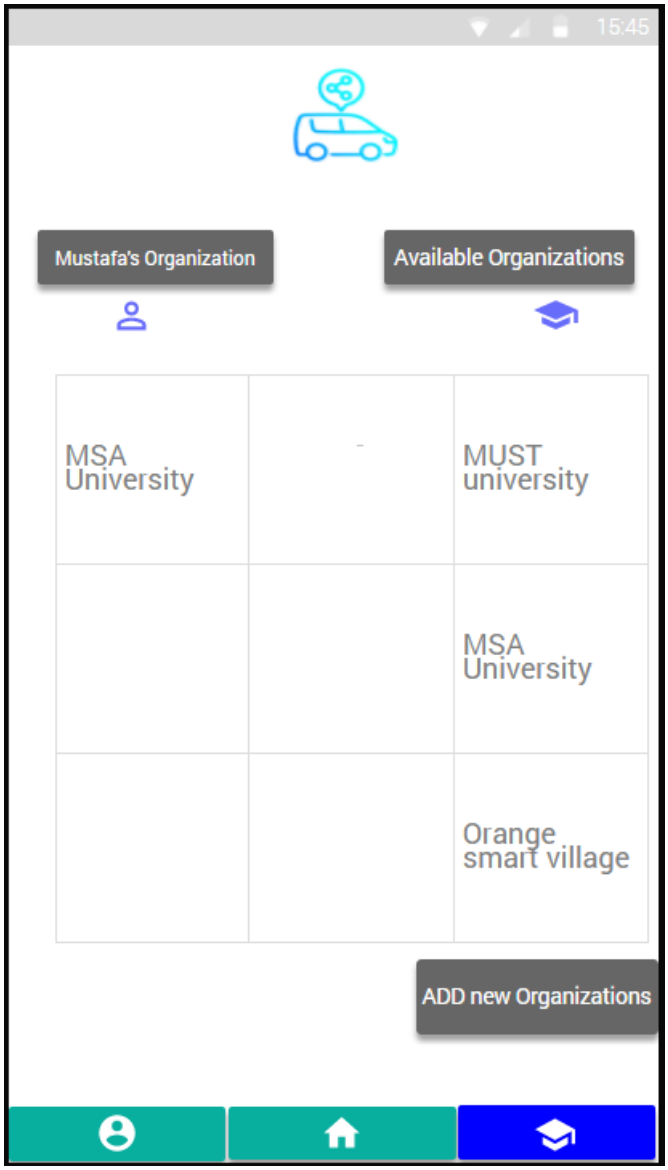


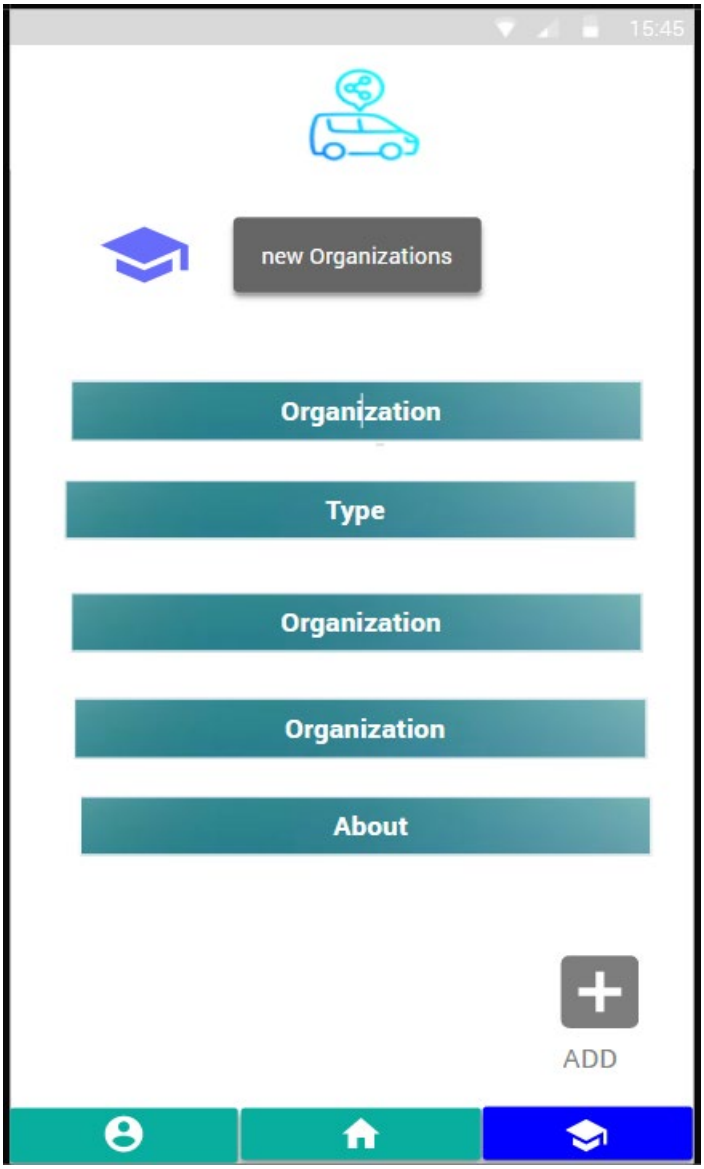
2.3 – if it's a driver function, the user can enter the maximum number of passengers to pickup

Hint: the user can make a post as an organization 'post only if he already joined one

3.1- entering profile page by choosing the profile icon below button will display the users is posts as a driver or passenger or the user's post in organization / the user can display the other user's posts

3.2 3.3





3.2 – entering the organization's interface by choosing the organization icon below button will display the list of the user 's already joined organizations and other facilities that the user can join

3.3- choosing (ADD new Organization) button will display another page that the user can create a new organization that other users can enter

4.2 Hardware Interfaces

This application is designed to run on Android smartphones.

4.3 Communications Interfaces

the application will use the smartphone Is hardware networking and communication services that's controlled by the android system, the communication will be used for importing and exporting the user's data between the application and the database in Google's firebase services.

5. Performance Requirements

System:

The application will run on Android mobile Operating system from 6.0 to android 10.0 or later, the application will adjust to be running on any mobile or tablet screen sizes.

Response Time:

- The application shouldn't take more than 10 sec to be running on the smartphones with the minimum hardware specification on running android 6.0.
- The functions response time shouldn't take more than 0.8 sec depending on the hardware performance and networking speed to load the data.

Workload:

The application should support load of more than 5000 user.

6. Other non-functional attributes

6.1 Security

The users must be registered with their email address and phone number beside having their Organization 's email credentials to access most of the application 's functionality.

6.2 Reliability

- The user's device must have a good internet connection for the applications 's database to be accessed and have most of the application's function fully active.
- The app should be crashed in user's activities.

6.3 Portability

The application should be accessed from the user's device at any time or anywhere.

6.4 Reusability

The application should be reusable in any time of the year, and the user must be able to use any of its functionality many times and load all of his past data.

6.5 Application Affinity/Compatibility

the application must be compatible with any device (Mobile/Tablet) that's running in Android 6.0 or later.

6.6 usability

The application must be easy to use its functionality by any user and navigating between screens.

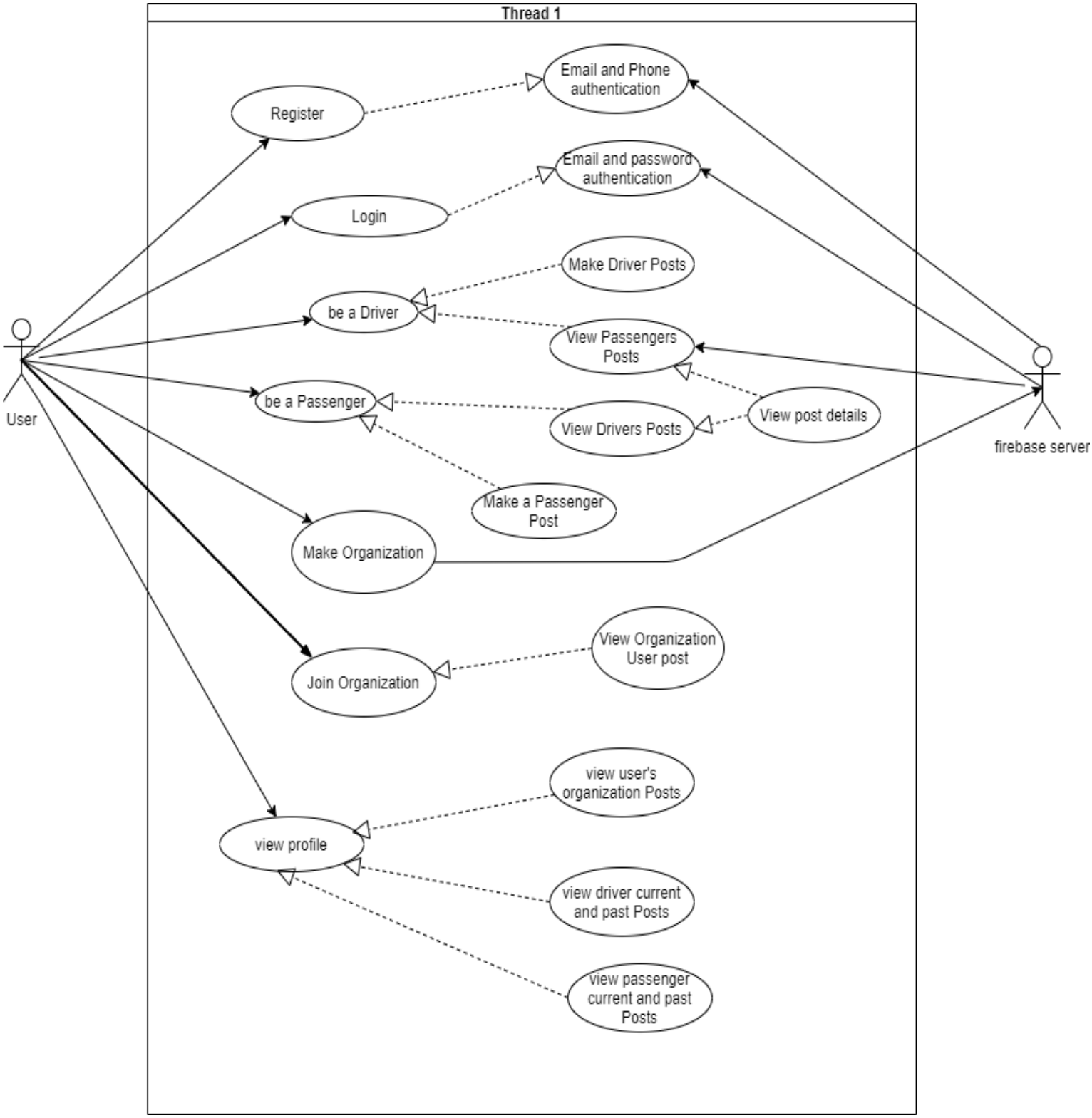
7 Operational Scenarios

1) When the user opens the application for the first time, the first page appear to him is the login and register page, the user register for the application with his email/password and phone number so the application authenticate his account ,then after saving the user information he goes to login page and enters his email and password to login.
2) After the user login with his newly registered account the application displays the home page which contain two main function that the user wants to be a passenger or a driver beside having sub function bellow like the profile and organization page that he can access.
3)
   - When the users enter the passenger page, he can see the offers from others users(drivers) if there is any and search for the posts that he preferred, the page have other main function that he can request a ride by pressing the new request button entering a new page to choose from/to location and destination then the pickup point then the Arriving and leaving time then the date of the trip and confirm to post the request
   - When the users enter the Driver page, he can see the requests from others users(passengers) if there is any, and search for the posts that he preferred, the page have other main function that he can offer a ride by pressing the new offer button entering a new page to choose from/to location and destination then the maximum number of passengers then the Arriving and leaving time then the date of the trip and confirm to post the offer
4) the user may create a new organization by entering its information (name , type, email website, Description) or joining an already exist one by entering its's email credentials (msa.edu.eg…..etc.) then he can view any of his organization's posts (any user in the same organization) and any offer or request he will post will appear as an organization post.
   - The user can only view or enter or post an organization's carpool posts when he joins the organization(university)
5) when the use enters an active carpool, he can view the road map and the expected time to reach the destination
6) the user can enter any post and view its details (location, destination, time, date driver or requester) and choose to join one or not
7) the user presses the profile icon to view his profile and see his past and current post as a driver or passenger and his organization's posts that he takes a part of it
8) the user may enter the logout button then he will go back the first login and registration page

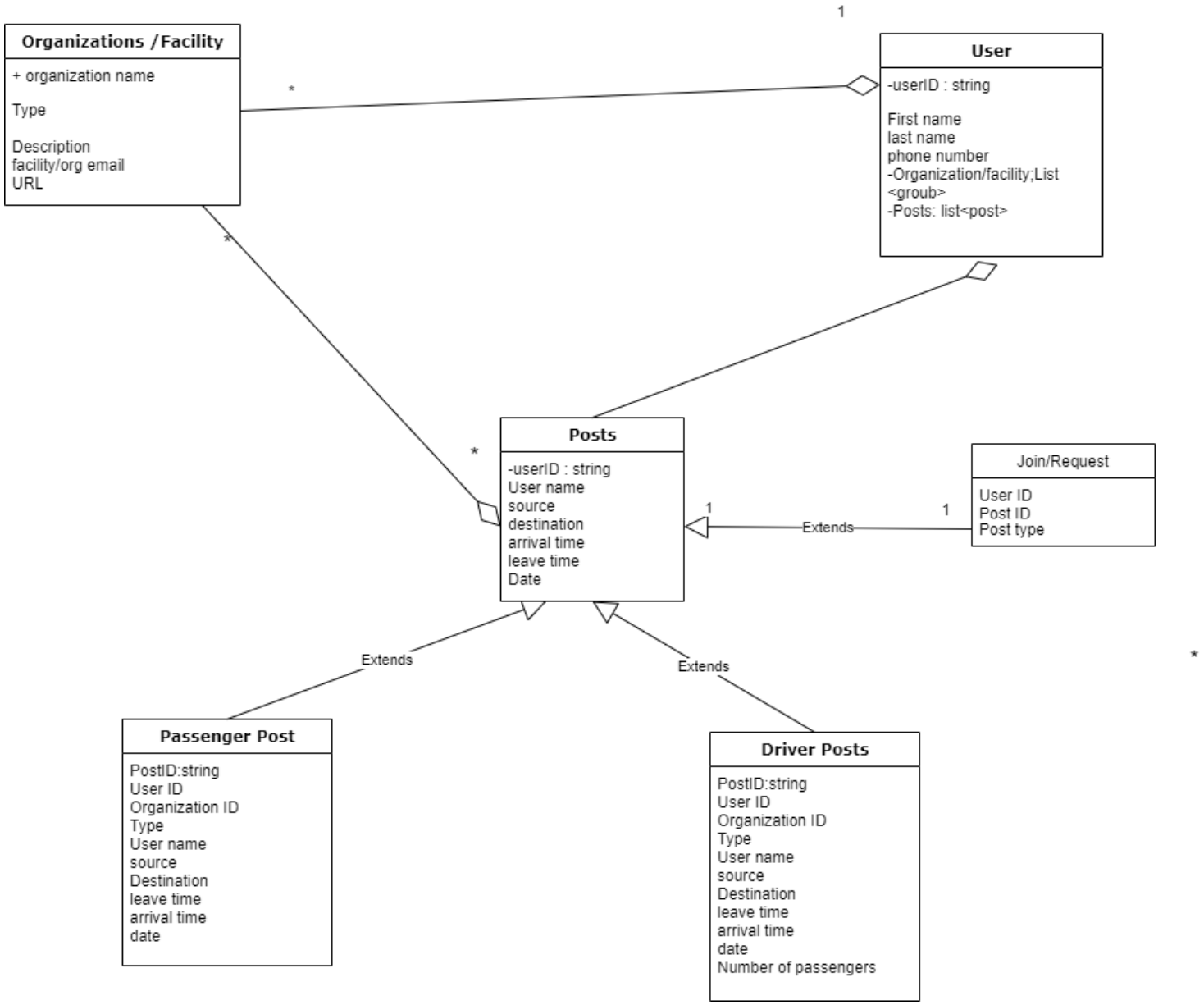
8 Application's Diagrams

8.1 Use case Diagram

(figure 5) this figure is for a use case that shows the functionality of the system



8.2 Class Diagram (figure 6) this class diagram shows the data object classes of the application and theire relationships



Chapter 3
Software Design Document

1. Introduction

1.1 Purpose

This software design document describes the design of the application and the components that's used to activate the features of the application that the user will interact with and the data architectures.

1.2 Scope

The goal of the application design is to provide a more functional way to enhance the carpooling experience for both Drivers and passengers, and view or create the user posts and organizations post in a sufficient way that's easy for the user to understand and use.

1.3 Overview

This document provides information about the design, components and data architecture of the application, to make it more understandable and easier to see the big picture for the users and any customers who wants to buy the application.

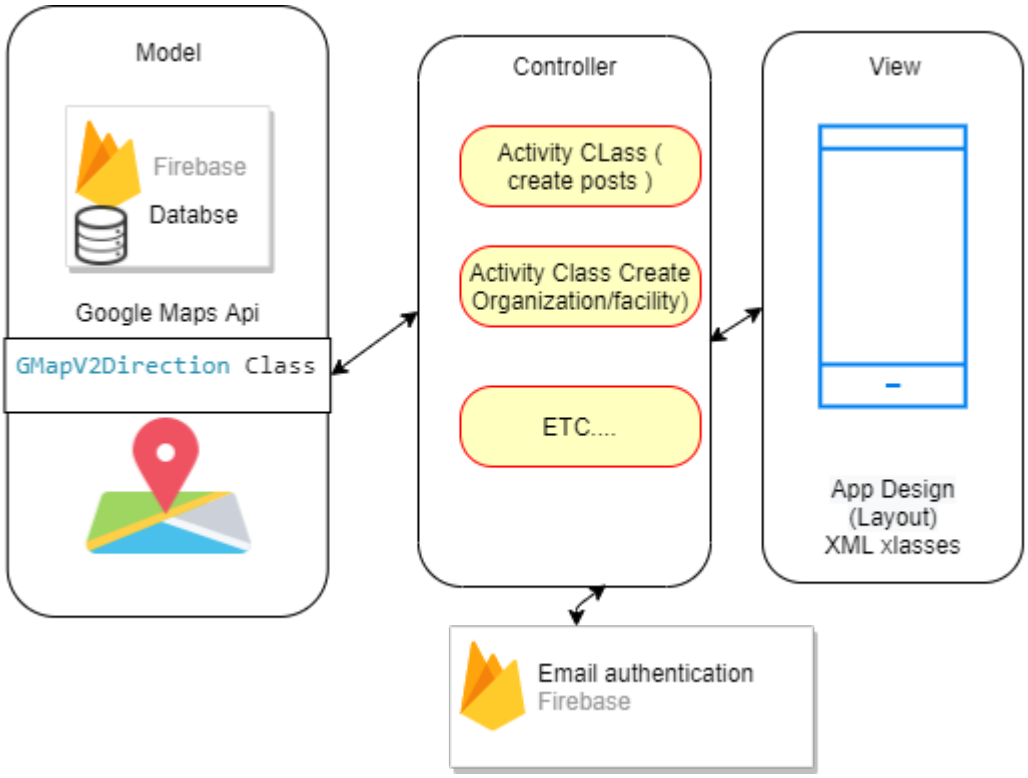
2 System Overview

This application will be designed to view all the users ride Requests/Offers posts that's the user can create and schedule it in a constructed way that the user can interact with to view more details or join it, or change or join a new organization that will displays another posts that's created by the users with the same organization that's has been chosen, the system will be designed to handle massive data from the user to be recycled and viewed.

3 System architecture

3.1 Architecture Design (Figure 7)

This figure shows the structure design of the mobile app which is MVC



The applications based on Android which will use MVC architecture design

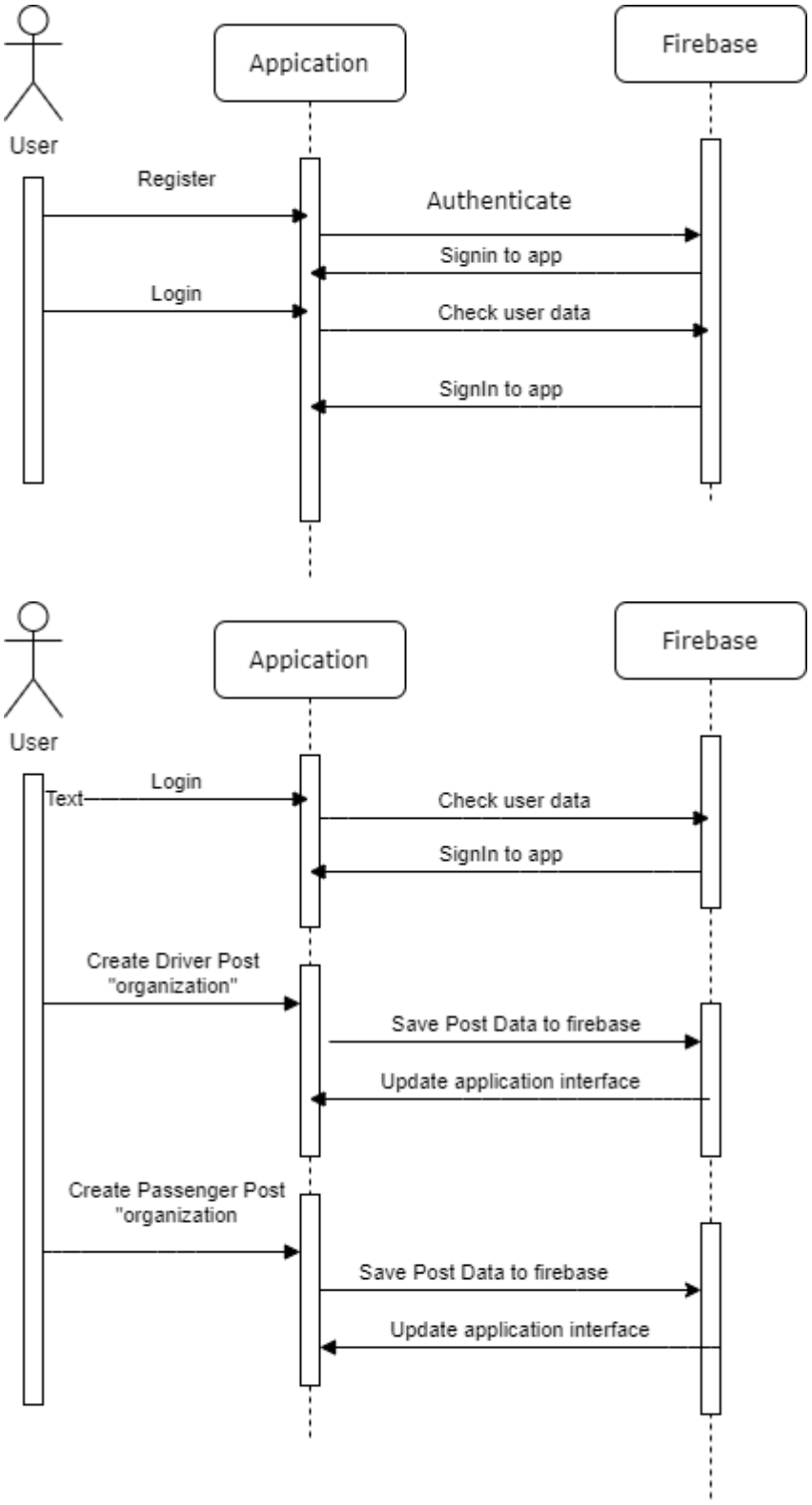
Model: which will have the data base classes (Data Objects) that's stored in Real time database on Firebase

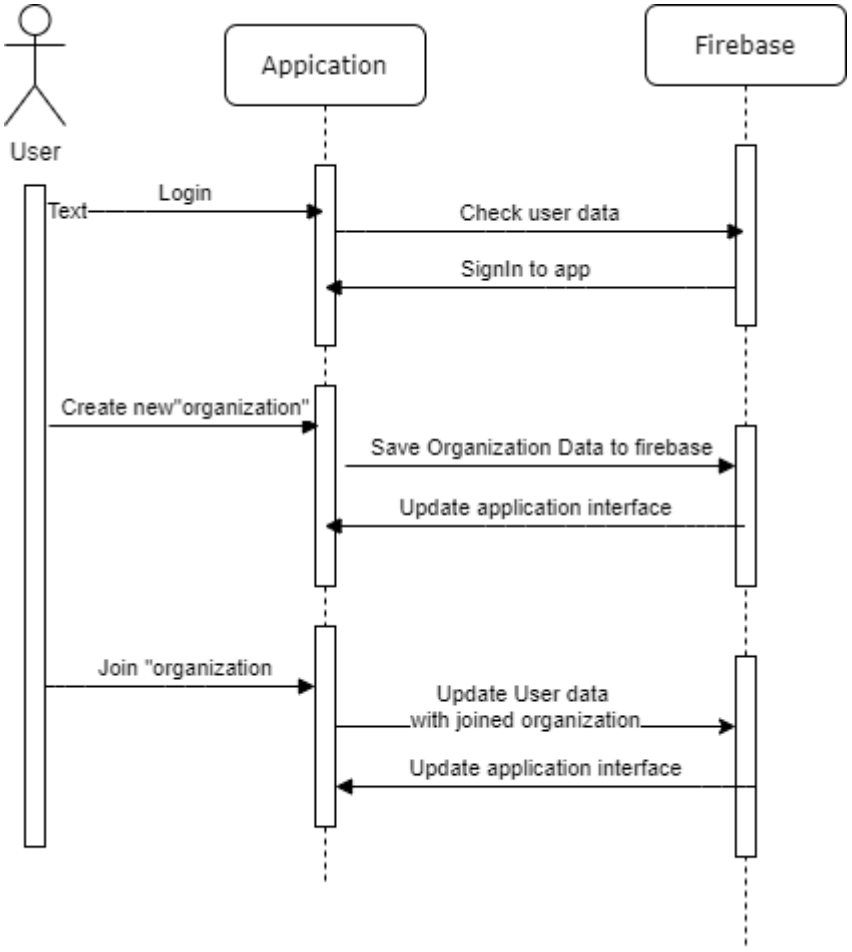
View: which will have the interface layout classes in XML

Controller: which will have the application function (Activity Classes)

3.2 Decomposition Description

(Figures 8) this show the sequence diagram of registering and creating driver/passenger post And join or create new organization/facility group





3.3 Design Rationale

The application architecture is selected based on the purpose of the system as it will be used by any type of users with using internet connectivity, so the application is designed to run as an android mobile application, which is best for MVC as the application architecture is separated from each other.

4 Implementation

4.3System Development

The system was developed in 3 different phases on the android studio platform which support java language and flatter Framework and it's based on the MVC structure with is

- View which is the XML layout that's communicating with Main functional classes 9 controller
- Model which hold the data that's been transmitted and updated through fire base then to the controller classes
- Controller which is the heart of the application that's performing the main functions and communicating with the model and view classes

1- First phase is designing the layout of the system to understand how the functions and activity will work and finding the best design to complete the any process in the easiest way for the normal user.

   The design phase is beginning by making the layout of the system .XML files that's created in Android studio by drag and drop the icons and the button the user will use beside the text inputs

2- The second phase is by choosing and designing the Data Objects of the system that's will be transmitted though the firebase which will confine around 7 model classes begins with

- User information data object class which have the (user name, email, and phone number), beside the default organization that's will be chosen by him
- Driver and passenger + posts data objects classes which (user id, post id the post type, source and destination, arrival and leave time, and trip date)
- Comments data objects, which have (user id, user name, and the comments)
- Organization which have (name, type, description, URL and email)
- Source and Destination in posts data object class which will have the detailed information about post that's will be viewed by SDL_VIewholder class

The model classes are called and updated by the firebase database reference, and uses parcelable to store and transfer the complex type of data such as the map latling data, bundle which can hold any type of data to transfer data between the classes

```
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
```

3- The third phase is developed with the second phase in semi parallel process, and this phase contains the function classes (Activity) which is divided in 4 Main activities classes

- The classes which is responsible for creating and viewing and creating new post
- The classes which is responsible for creating and viewing and joining Facility/organization
- The classes which is responsible for the user information page and editing it
- The Fragments classes which is a small subclass from the main activity to complete it
- The view holder classes which is responsible for storing a Listview for posts and organizations to be viewed several pages

  The RecyclerView works with the viewholder so when scrolling the not visible posts that's disappear off screen when scrolling is recycled and reused with the post data that's appear to prevent the ram lacking issues and crashes of the application

4.2 Used APIs

1- DialogFragment Datepicker which is used to show a Date calendar in a GUI to choose from

2- DialogFragment TimePIckerDialog which is a clock GUI to choose a time from

3- Google maps API v2 which is used to place a marker on the location and destination show direction between them (latling)

4- Maps SDK for android, to view the world map

5- Places API to search for the placed (destination, and location)

6- Devlight.ntb.NavigationTabBar which is responsible for creating the action bar slider

7- Synnapps.varouselView that's responsible for creating floating page when creating the post

5 Data Design

5.1 Data Description

The data will be stored in Firebase Realtime Database that will be updated continuously with every user interaction, and will be called by database Reference to be displayed on the application , the system view component will be implemented as a "Recycler view" component so the application can handle a massive data that's coming from the users to be displayed.

5.2 Data Dictionary

Table User

Column Name	Type
User_name	String
First_name	String
Last_name	*String*
phone	*String*

Table Comments

Column Name	Type
User_id	String
User_name	String
Comment	*String*

Table Posts Driver /Passenger

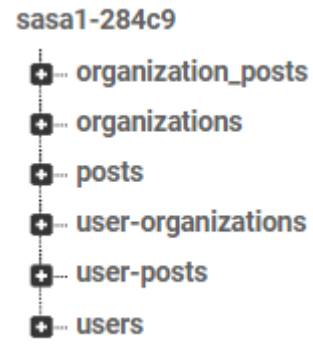
Column Name	Type
User_id	String
Username	String
source	string
Destination	String
Num of passengers	Int
Leave time	Int
Arrive time	Int
Trip date	int

Table Facility/Organization

Column Name	Type
Name	String
Type	String
Description	string
Email	String
URL	string

5.3 Database View

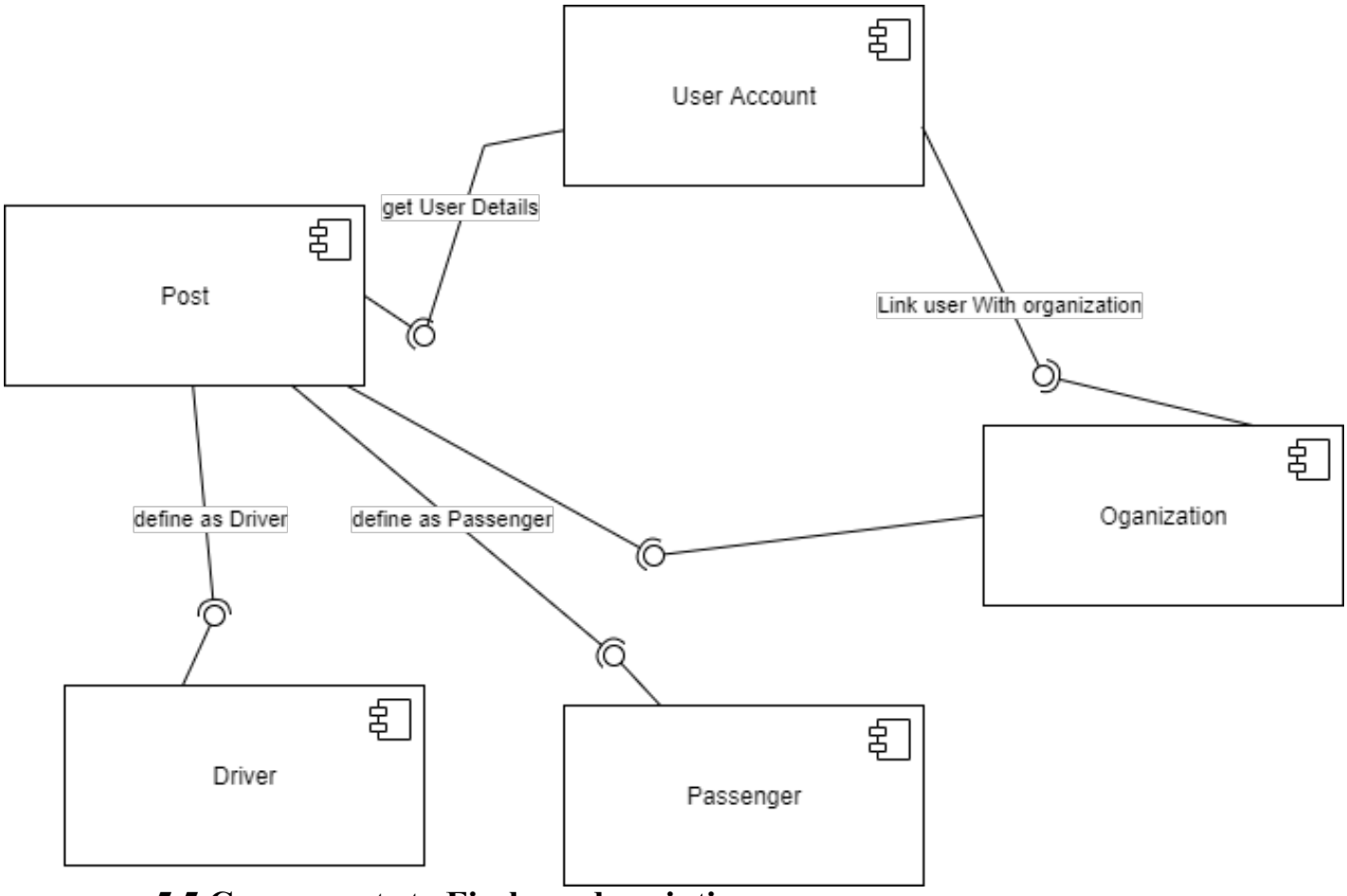
(Figures 9) this figure shows the architecture of the data base design in firebase




user-organizations
4yK4IUC3cDbIqtClwpsQM0YAIJP2
64m7FNB14SXGEnhX8ciSPX26dzz2
HK5ZTcMarcgqlfm6wskNR6SwM6a2
a872OSK165SIGK5drGbQfeP5TnY2
user-posts
4yK4IUC3cDbIqtClwpsQM0YAIJP2
64m7FNB14SXGEnhX8ciSPX26dzz2
HK5ZTcMarcgqlfm6wskNR6SwM6a2
users
4yK4IUC3cDbIqtClwpsQM0YAIJP2
64m7FNB14SXGEnhX8ciSPX26dzz2
First_name: "Mostafa
Last_name: "Hesham
Phone: "\"\"01276710187
User_email: "mustafahisham1998@gmail.cc
default_organization: "-MB5WzrkBZBswMWXL01
phoneNumber: "\"\"01276710187
HK5ZTcMarcgqlfm6wskNR6SwM6a2
a872OSK165SIGK5drGbQfeP5TnY2

5.4 Component Design

(figure 10) this show the component design of the application and its communication



5.5 Components to Firebase description

IF user signs up to the application

Firebase authenticate the user data and save it

If the user login to the application

The user can now create a new post as a driver or as a passenger

The post data is saved in firebase and then called when needed to be displayed in the application

The user can join or create organization

The firebase updates the organization data and the users that's are related to it

If the user is in organization

Then view the posts of each of the organizations he joined separately

If the user changed his default organization

The firebase changes the default organization

The user now can make a post for his default organization

If the user clicked on one of the created posts

The firebase sends post data to be displayed to the user

The application displays the post detailed info

If the user deleted a post

Firebase delete the post data from the organization section, but it is still saved in user post as a history

If the users enter Driver page

The firebase gets the passenger posts data

The passenger posts displayed

If the users enter Passenger page

The firebase gets the Driver posts data

The Driver posts displayed

If the user chooses to join post

The firebase messages send post data and user data through messages to third party messaging app by using both users phone number

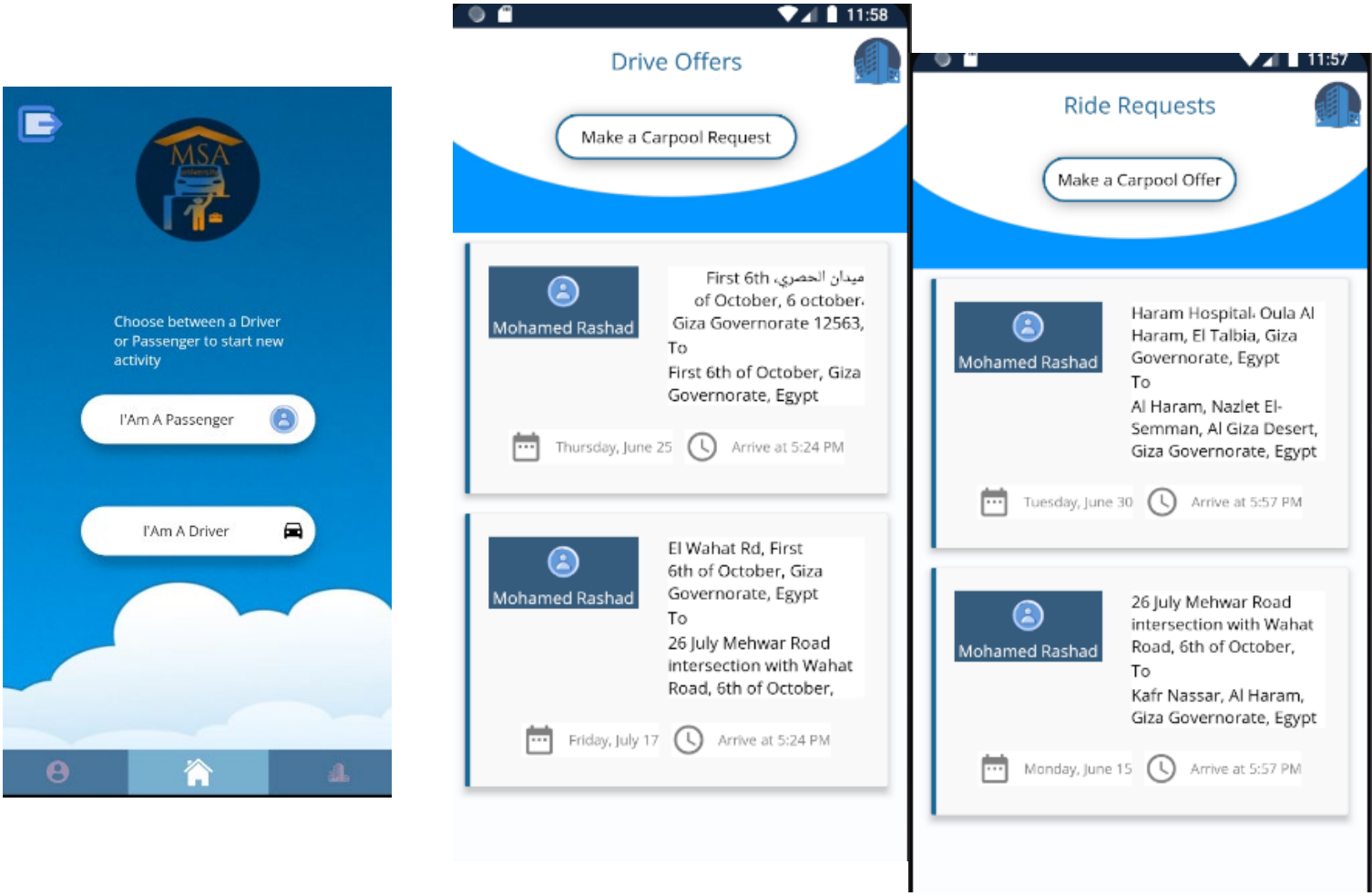
6 Human Interface Design

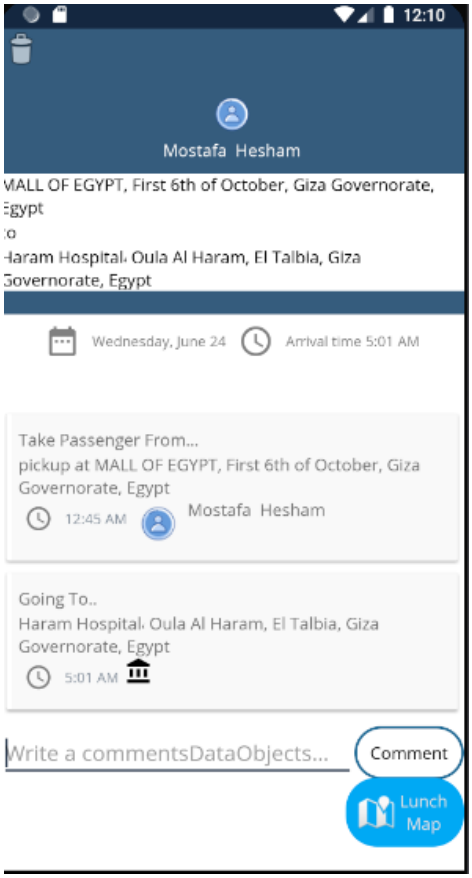
6.1 Overview of User Interface

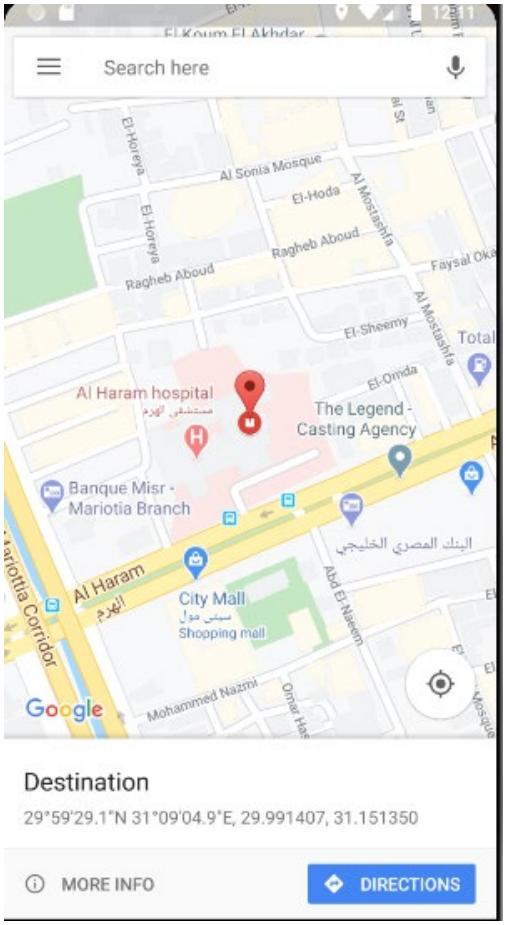
When user sign in with authentication email / Gmail/ Facebook …etc. the user will have access to the applications functions, the user could create a passenger or a driver post beside having sub function bellow like the profile and organization page that he can access. The users can scroll the passengers post in driver page or scrolled drivers offers posts in passenger page, and choose which post will be matching his needs, the user should join an organization from the listed and scrolling in it to find his needs or create new one with new authenticated email pattern to have a more specified view posts with his friends or company to be shared with him .

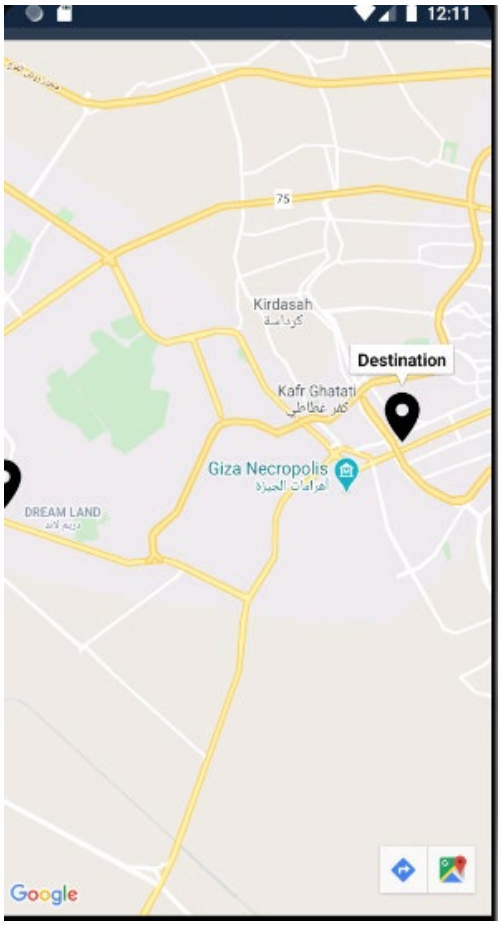
6.2 Screen Images of the application (figure 11)

This shows all of the application interfaces that the user will interact with

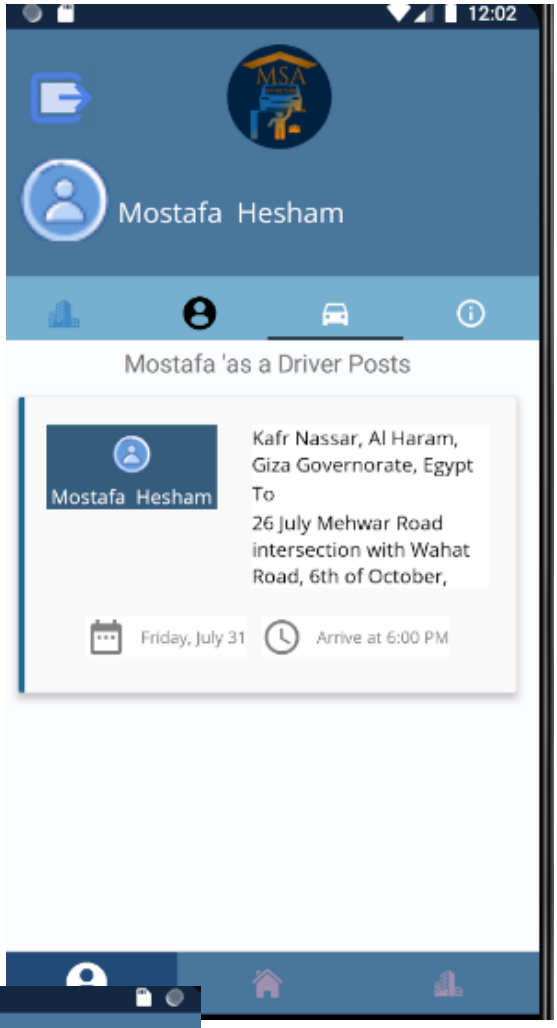


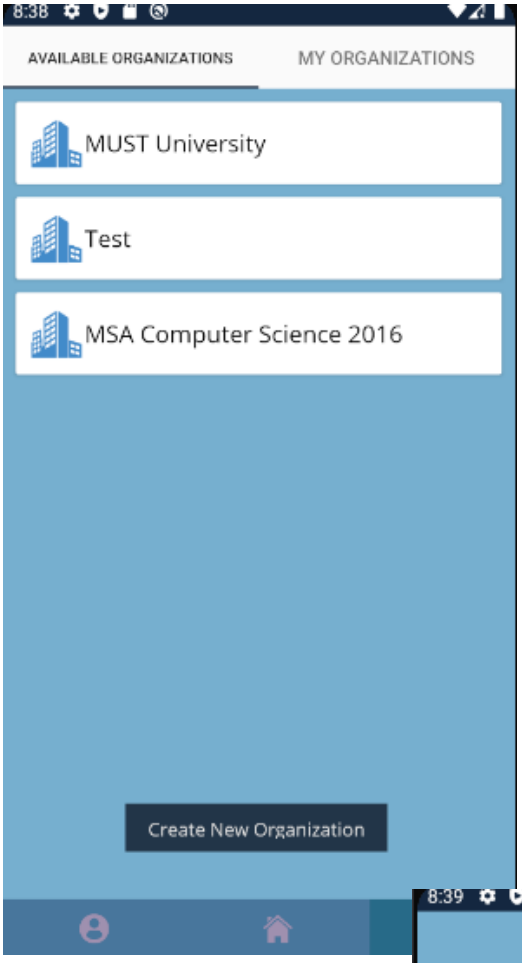

12:10
Mostafa Hesham
MALL OF EGYPT, First 6th of October, Giza Governorate, Egypt
:o
Haram Hospital، Oula Al Haram, El Talbia, Giza Governorate, Egypt
Wednesday, June 24
Arrival time 5:01 AM
Take Passenger From...
pickup at MALL OF EGYPT, First 6th of October, Giza Governorate, Egypt
12:45 AM
Mostafa Hesham
Going To..
Haram Hospital، Oula Al Haram, El Talbia, Giza Governorate, Egypt
5:01 AM
Write a commentsDataObjects...
Comment
Lunch Map

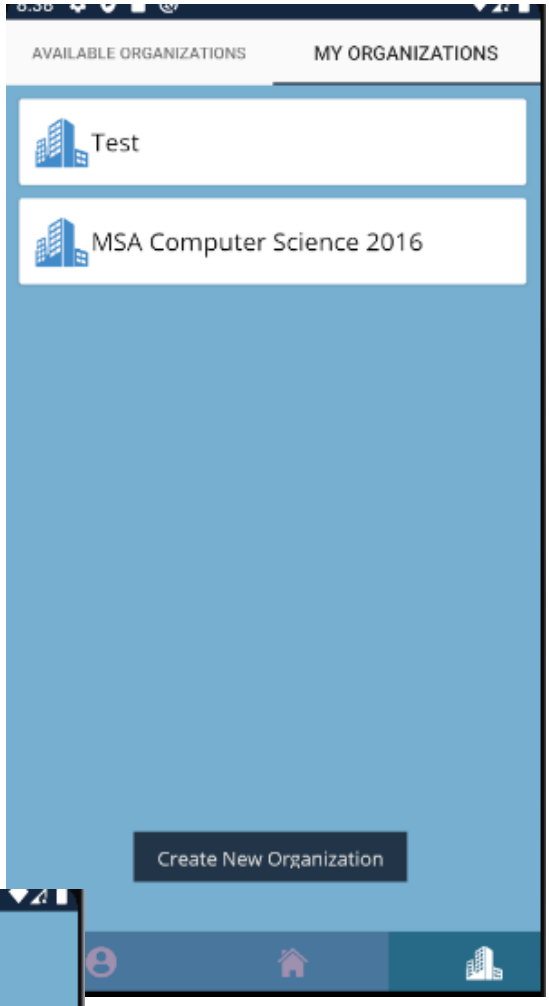

Search here
Al Haram hospital
The Legend - Casting Agency
Banque Misr - Mariotia Branch
Al Haram
City Mall
Shopping mall
Google
Destination
29°59'29.1"N 31°09'04.9"E, 29.991407, 31.151350
MORE INFO
DIRECTIONS

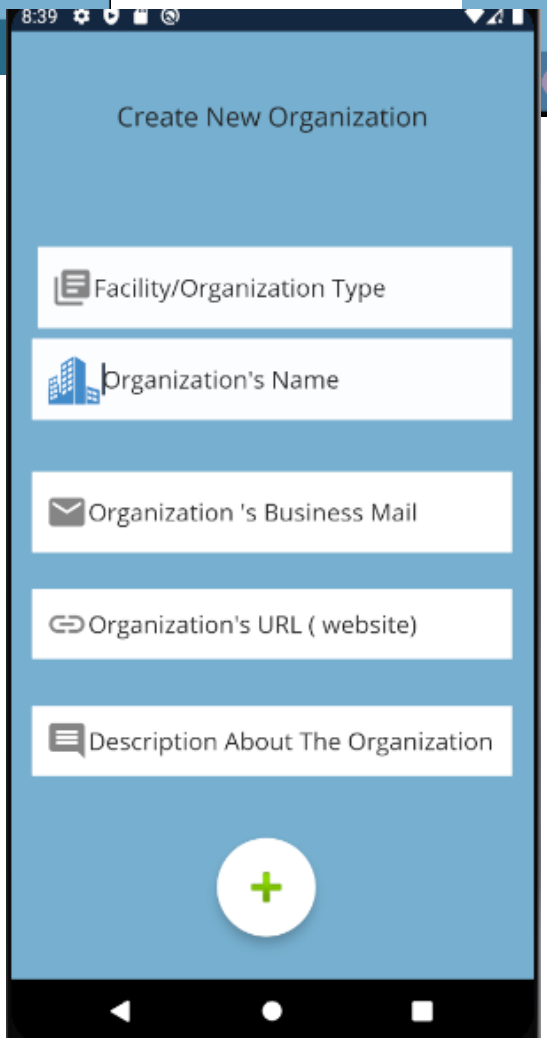

12:11
75
Kirdasah
Destination
Kafr Ghatati
Giza Necropolis
DREAM LAND
Google

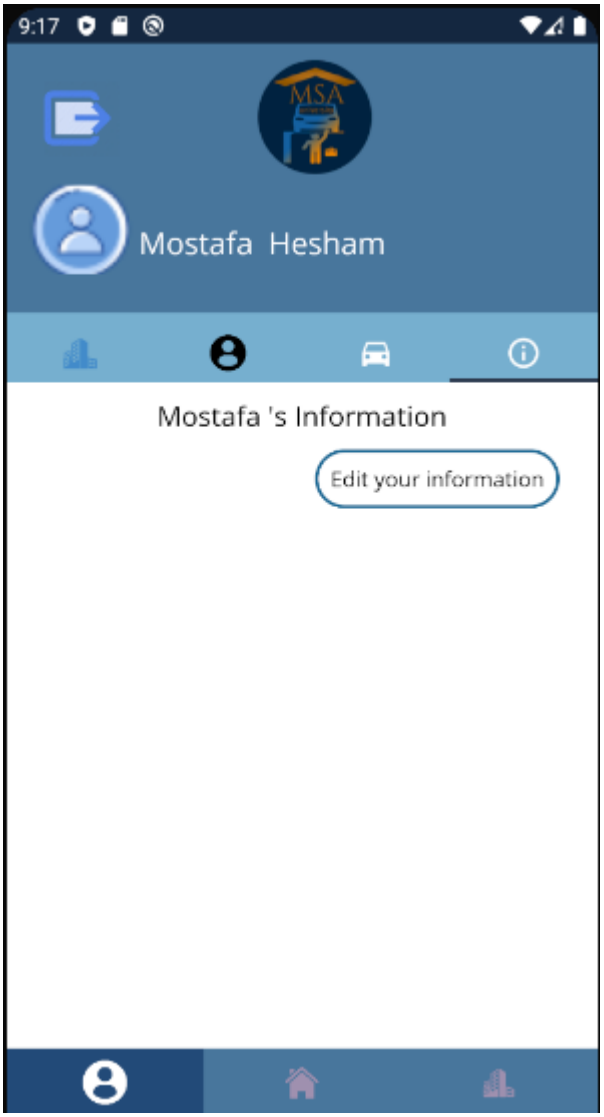

12:05
Mostafa Hesham
Mostafa 's Organizations
Test
MSA Computer Science 2016

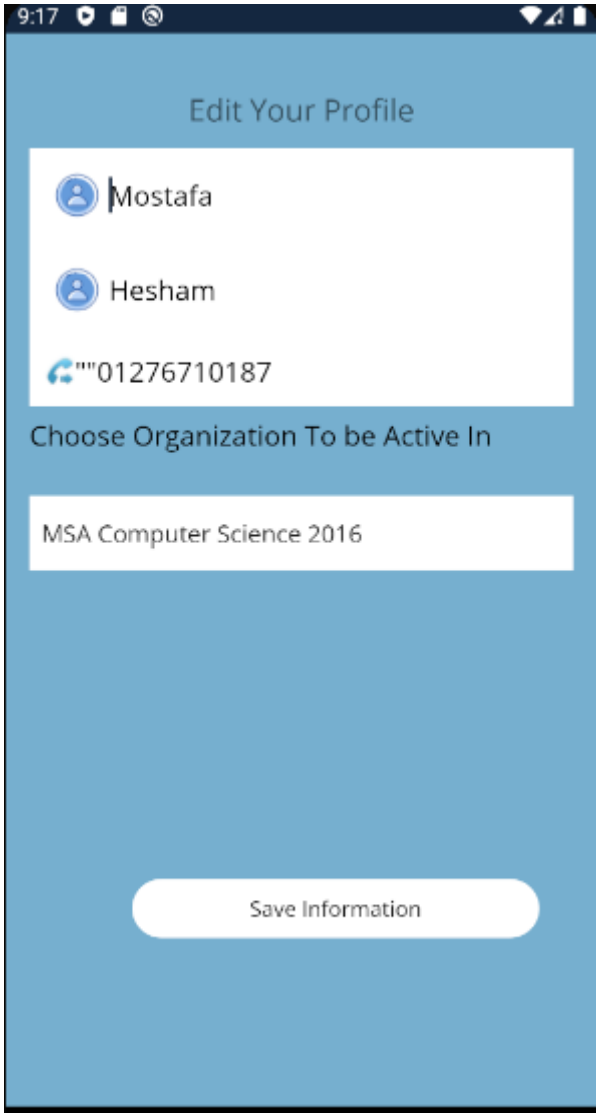

12:02
Mostafa Hesham
Mostafa 'as a Driver Posts
Mostafa Hesham
Kafr Nassar, Al Haram, Giza Governorate, Egypt
To
26 July Mehwar Road intersection with Wahat Road, 6th of October,
Friday, July 31
Arrive at 6:00 PM

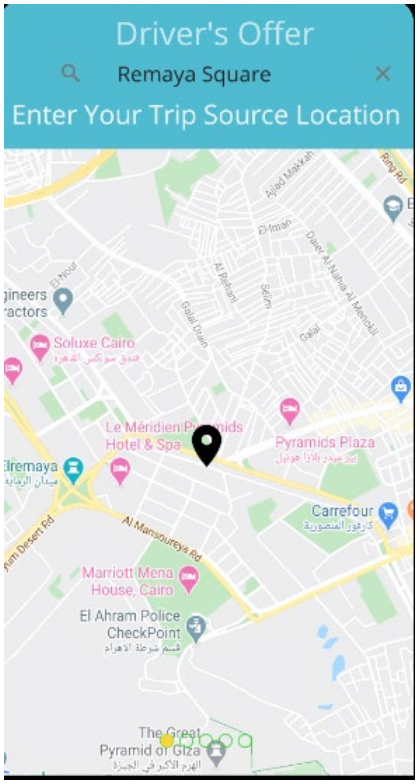

8:38
AVAILABLE ORGANIZATIONS
MY ORGANIZATIONS
MUST University
Test
MSA Computer Science 2016
Create New Organization

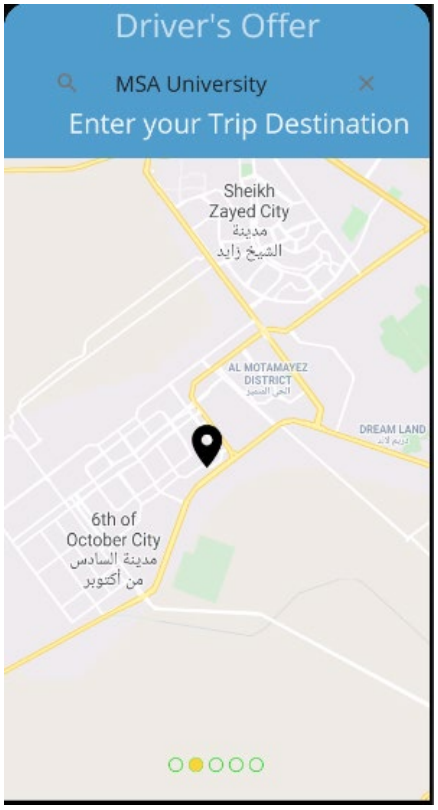

AVAILABLE ORGANIZATIONS
MY ORGANIZATIONS
Test
MSA Computer Science 2016
Create New Organization

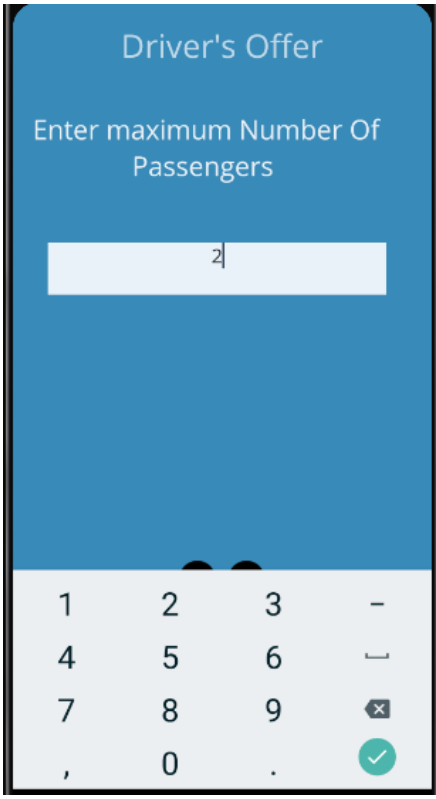

8:39
Create New Organization
Facility/Organization Type
Organization's Name
Organization 's Business Mail
Organization's URL (website)
Description About The Organization

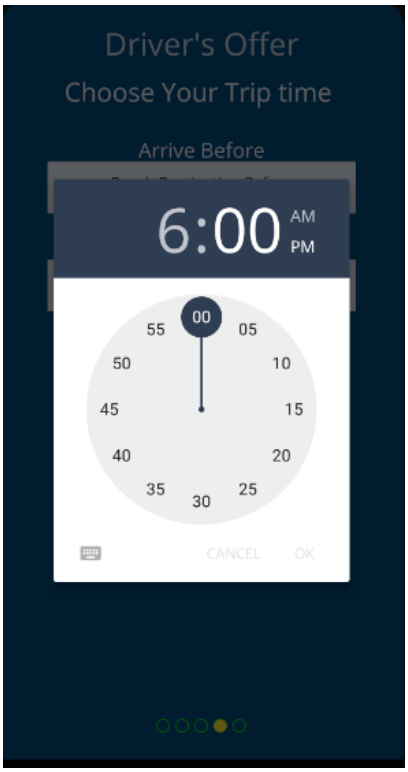

9:17
Mostafa Hesham
Mostafa 's Information
Edit your information

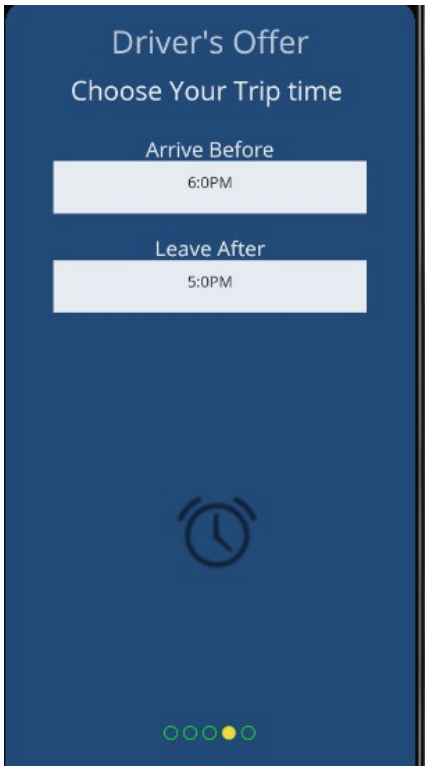

9:17
Edit Your Profile
Mostafa
Hesham
""01276710187
Choose Organization To be Active In
MSA Computer Science 2016
Save Information

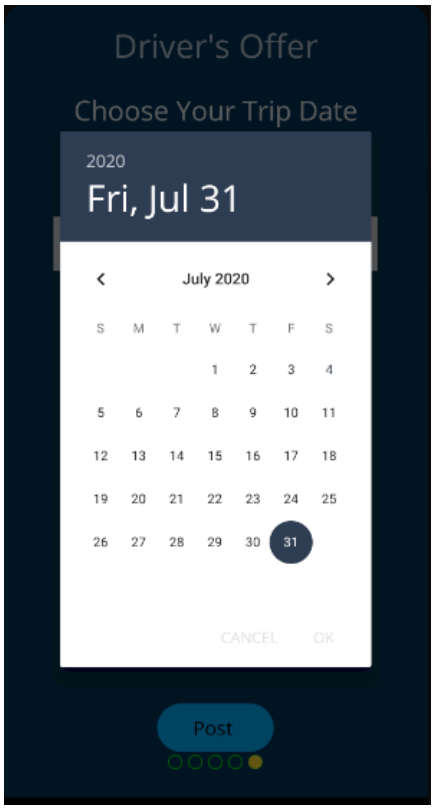

Driver's Offer
Remaya Square
Enter Your Trip Source Location
Soluxe Cairo
Le Méridien Pyramids Hotel & Spa
Pyramids Plaza
Carrefour
Marriott Mena House, Cairo
El Ahram Police CheckPoint
The Great Pyramid of Giza


Driver's Offer
MSA University
Enter your Trip Destination
Sheikh Zayed City
مدينة الشيخ زايد
AL MOTAMAYEZ DISTRICT
DREAM LAND
6th of October City
مدينة السادس من أكتوبر


Driver's Offer
Enter maximum Number Of Passengers
2
1 2 3 -
4 5 6
7 8 9
, 0 .


Driver's Offer
Choose Your Trip time
Arrive Before
6:00 AM PM
00 05 10 15 20 25 30 35 40 45 50 55
CANCEL OK


Driver's Offer
Choose Your Trip time
Arrive Before
6:0PM
Leave After
5:0PM


Driver's Offer
Choose Your Trip Date
2020
Fri, Jul 31
July 2020
S M T W T F S
1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
CANCEL OK
Post

Chapter 4
Evaluation and Testing

1.0 Testing

1.1 Test Cases

Test case name ID	T1
Tested Requirements	RQ 6
Test scenario	Signing up to the application
Test case description	The users enter his name and email and password and phone number to register to the application
Pre-requisites	The user device must have an internet connection
Test input	First Name: Mostafa Last Name: Hesham Email: mustafahisham1998@gmail.com Password: 123456 Phone: 01221515452
Execution steps	1- Open the app 2- Press signup button 3- Type in Mostafa in first name input field 4- Type in Hesham in Last name input field 5- Type in mustafahisham1998@gmail.com in email input field 6- Type in 123456 in password input field 7- Type in 01221515452 in phone input field 8- Click Signup button
Expected behavior	The application will direct user to the home page. the user data will be saved in firebase
Future Test	The GUI and Firebase will be tested in a System testing scenario
Actual results	The application opens the home page with the user inputs and save data to the firebase
Status	tested

Test case name ID	T2
Tested Requirements	RQ 6
Test scenario	Signing up to the application without writing the first name
Test case description	The users enter his name and email and password and phone number to register to the application
Pre-requisites	The user device must have an internet connection
Test input	First Name: Last Name: Hesham Email: mustafahisham1998@gmail.com Password: 123456 Phone: 01221515452
Execution steps	1- Open the app 2- Press signup button 3- Type in Hesham in Last name input field 4- Type in mustafahisham1998@gmail.com in email input field 5- Type in 123456 in password input field 6- Type in 01221515452 in phone input field 7- Click Signup button
Expected behavior	The application should show an error message with a required field “First Name“
Future Test	The GUI and Firebase will be tested in a System testing scenario
Actual results	The application shows an error message with a required field “First Name”
Status	tested

Test case name ID	T3
Tested Requirements	RQ 7
Test scenario	Signing In to the application
Test case description	The users enter his email and password to sign in to the application
Pre-requisites	The user device must have an internet connection
Test input	Email: mustafahisham1998@gmail.com Password: 123456
Execution steps	1- Open the app 2- Type in mustafahisham1998@gmail.com in email input field 3- Type in 123456 in password input field 4- Click SignIn button
Expected behavior	The application should direct user to the home page.
Future Test	The GUI and Firebase will be tested in a System testing scenario
Actual results	The application opens the home page with the user Data
Status	tested

Test case name ID	T4
Tested Requirements	RQ 7
Test scenario	Signing In to the application
Test case description	The users enter wrong email or password to sign in to the application
Pre-requisites	The user device must have an internet connection
Test input	Email: mustafahishamhhghg@gmail.com Password: 100000
Execution steps	1- Open the app 2- Type in mustafahishamhhghg@gmail.com in email input field 3- Type in 100000 in password input field 4- Click Sign In button
Expected behavior	The application should not direct user to the home page and should display an error message.
Future Test	The GUI and Firebase will be tested in a System testing scenario
Actual results	The applications show an error message “wrong email or password” in the same sign in page
Status	tested

Test case name ID	T5
Tested Requirements	RQ 1
Test scenario	Display home page
Test case description	When the user enters his email and password cardinals and signing the system will display the home page which have nab bar and a Driver / Passenger button to start activity
Pre-requisites	-The user device must have an internet connection - user should have signed up to the application
Test input	Enters any valid user name and password
Execution steps	1- Open the app 2- Enters Login credentials 3- Click Sign In button
Expected behavior	The application should direct the user to the home page which will displays a Driver/Passenger button and navigation bar
Future Test	none
Actual results	The application directed the user to the home page and displayed a Driver/Passenger button and navigation bar, with the user information
Status	tested

Test case name ID	T6
Tested Requirements	RQ 2
Test scenario	Make a Driver offer Post
Test case description	When users enters the home page and choose " I am a driver button "then click the "make a carpool offer" , he can be able to make a driver post by entering the post data
Pre-requisites	-The user device must have an internet connection - user should have signed in to the application
Test input	- MSA as destination - remaya square as a Location - enter 2 as a number of passengers - trip time, 6 PM "arrive before" - trip time, Leave source after 5 Pm - trip date 31/7/2020
Execution steps	1- Open the app 2- Click "I am a Driver" button 3- Click on "Make a Carpool Offer" button 4- Write Remaya square in source field 5- Write MSA university in destination filed 6- Enters 2 in passengers' field 7- Enters 6 pm in arrive before filed 8- Enters 5 pm in leave source filed 9- Enters trip date 31/7/2020 10- Click on post button
Expected behavior	The application should direct the user back to the Driver page, the post should be added to fire base then displays in passenger page
Future Test	none
Actual results	The application goes back to the Driver page. And the actual post Is displayed correctly in the passenger page
Status	tested

Test case name ID	T7
Tested Requirements	RQ 3
Test scenario	Make a Passenger request Post
Test case description	When users enter the home page and choose “I am a Passenger button “then click the “make a carpool Request”, he can be able to make a Passenger post by entering the post data
Pre-requisites	-The user device must have an internet connection - user should have signed in to the application
Test input	- MSA as destination - Remaya square as a Location - enter pickup location - trip time, 6 PM “arrive before” - trip time, Leave source after 5 Pm - trip date 31/7/2020
Execution steps	1- Open the app 2- Click “I am a Passenger” button 3- Click on “Make a Passenger request” button 4- Write Remaya square in source field 5- Write MSA university in destination filed 6- Enters any pick up location between source and destination 7- Enters 6 pm in arrive before filed 8- Enters 5 pm in leave source filed 9- Enters trip date 31/7/2020 10- Click on post button
Expected behavior	The application should direct the user back to the Passenger page, the post should be added to fire base then displays in Driver page
Future Test	none
Actual results	The application goes back to the Passenger page. And the actual post Is displayed correctly in the Driver page
Status	tested

Test case name ID	T8
Tested Requirements	RQ 4
Test case name ID	T8
Tested Requirements	RQ 4
Test scenario	Create a new Facility/Organization group
Test case description	When users enter the home page and choose “Organization in the action bar “then click the “Create new organization”, he can be able to make an Organization group post by entering the organization data
Pre-requisites	-The user device must have an internet connection - user should have signed in to the application
Test input	- type: university - name: MSA computer science 2016 - Mail: CS2016@msa.edu.eg - URL: www.msa.com - Description: this group is for MSA 2016 Students
Execution steps	1- Open the app 2- Click on “organization button in action bar” button 3- Click on “Create new organization” button 4- Write university in type field 5- Write “MSA computer science 2016” in name field 6- Write cs2016@msa.edu.eg in email field 7- Write www.msa.com in URL field 8- Write “this group is for MSA 2016 Students” 9- Click on “+” button
Expected behavior	The application should direct the user back to the Organization main page, the Organization group should be added to fire base then displays in “Available Organization” section
Future Test	none
Actual results	The application goes back to the Organization main page and the Organization group is displayed in “Available Organization” section successfully
Status	tested

Test case name ID	T9
Tested Requirements	RQ 5
Test scenario	The user joins an organization/Facility group
Test case description	When users enter the home page and choose “Organization in the action bar “then click on any organization he wants to join in “Available Organization” section he can display the organization details and enter the email credential to join it
Pre-requisites	-The user device must have an internet connection - user should have signed in to the application
Test input	-Organization email: cs2016@msa.edu.eg
Execution steps	1- Open the app 2- Click on “organization button in action bar” button 3- Click on “Available organization” 4- Click on desired organization 5- Write “Organization email: cs2016@msa.edu.eg 6- Click “join the organization” button
Expected behavior	The application should direct the user back to the Organizations main page, the user should be added to the Organization in fire base then displays in “My Organizations” section
Future Test	none
Actual results	The application directs the user back to the Organizations main page and the joined organization is successfully displayed in “MY Organizations” section
Status	tested

Test case name ID	T10
Tested Requirements	RQ 8
Test scenario	The user logs out of the application
Test case description	When user want to sign out of the application, he clicks on the logout image button
Pre-requisites	-The user device must have an internet connection - user should have signed in to the application
Test input	none
Execution steps	1- Open the app 2- Click on “log-out image button” button in home page or user information page
Expected behavior	The application should direct the user back to the signing in page
Future Test	none
Actual results	The application successfully logout the users and displays the sign in page
Status	tested

Test case name ID	T11
Tested Requirements	RQ 9
Test scenario	The user displays the chosen organization posts only
Test case description	Enter the Driver or passenger page, he can choose to View specific organizations posts by click on one of the joined organizations
Pre-requisites	-The user device must have an internet connection - user should have signed in to the application
Test input	None
Execution steps	1- Open the app 2- Click “I am a driver” button or “I am a passenger button” 3- Click on organization image button 4- Click on MSA computer science 2016 group
Expected behavior	The application should display the chosen organization posts
Future Test	None
Actual results	The application successfully displays only the chosen organization posts
Status	tested

Test case name ID	T12
Tested Requirements	RQ 12
Test scenario	Deleting created post
Test case description	The enters enter his Owen created post and click on recycler image button to delete it
Pre-requisites	-The user device must have an internet connection - user should have signed in to the application
Test input	None
Execution steps	1- Open the app 2- Click “I am a driver” button or “I am a passenger button” 3- Click on Owen created post from user 4- Click in recycler image --- 1-Open the app 2- Click “user page” button in the action bar 3- Click on driver or passenger button and choose any post, click on 4- Click in recycler image
Expected behavior	The post should be deleted from the interface and firebase
Future Test	None
Actual results	The post is successfully deleted in both interface and firebase data
Status	tested

Test case name ID	T13
Tested Requirements	RQ 11,10
Test scenario	Browse posts
Test case description	The user enters the driver or passenger page to display the passenger's requests in driver page, or display drivers offers in passenger page
Pre-requisites	-The user device must have an internet connection - user should have signed in to the application
Test input	None
Execution steps	1- Open the app 2- Click "I am a driver" button 3- Click on organization RQ 10 (not required) 1- Open the app 2- Click "I am a Passenger" button 3- Click on organization RQ 10 (not required)
Expected behavior	-The application should display passengers' posts in the driver page - The application should display Drivers posts in the Passenger page
Future Test	None
Actual results	-The application displays passengers' posts in the driver page successfully - The application displays Drivers posts in the Passenger page successfully
Status	tested

Test case name ID	T14
Tested Requirements	RQ 12
Test scenario	View all of the past created posts
Test case description	The user enters his information page to view all of his created drivers and passengers ‘s posts + view any of the post details
Pre-requisites	-The user device must have an internet connection - user should have signed in to the application
Test input	None
Execution steps	4- Open the app 5- Click user information” button in the action bar 6- Click on Driver or passenger slider button 7- User click on any of the posts
Expected behavior	-The application should display all of the past passengers’ posts in the user passenger section - The application should display all of the past Drivers posts in the user Driver section - application display post details
Future Test	None
Actual results	-The application successfully displays all of the past user passengers’ posts in the user Passenger section - The application successfully displays all of the past user Drivers posts in the user Driver section - the application successfully displays the post details (time, date, location and destination on active map)
Status	tested

Test case name ID	T15
Tested Requirements	RQ 13
Test scenario	View entered location and destination on map
Test case description	The user enters his desired go to destination, or enters his go from location That's will be directly displayed the location on the active map
Pre-requisites	-The user device must have an internet connection - user should have signed in to the application
Test input	1- -location "Remaya square" 2- Destination "MSA university"
Execution steps	1- Open the app 2- Click on "I am the Driver" or "I am the passenger" button 3- Enter "Remaya Square" in location 4- Enter "MSA university" in destination 1- Enter any post, click on map button to view route details
Expected behavior	-The application should display an active map in the location page, and view the entered location place in map -The application should display an active map in the Destination page, and view the entered Destination place in map
Future Test	None
Actual results	1-The application successfully displays an active map in the location page, and view the entered location place in map 2-The application successfully displays an active map in the Destination page, and view the entered Destination place in map The application successfully displays the desired route from the location and the destination in the map for the chosen post
Status	tested

Test case name ID	T16
Tested Requirements	RQ 14
Test scenario	Cancel any multi step
Test case description	The user enters to create a new post and before click on post button he canceled the process
Pre-requisites	-The user device must have an internet connection - user should have signed in to the application
Test input	1 -location “Remaya square” 2- Destination “MSA university” 3- 5 (num of passengers) 4- Time 6 pm arrival 5- Time 7 pm leave 6- Date 31/7/2020
Execution steps	1 Open the app 2 Click on “I am the Driver” or “I am the passenger” button 3- Click on create driver offer or passenger request 4- Proceed the post inputs 5- Press back button to cancel the process
Expected behavior	The application should go back normally to driver or passenger page
Future Test	None
Actual results	The application successfully goes back to the driver or passenger page without saving any of the inputs
Status	tested

Test case name ID	T17
Tested Requirements	RQ 15
Test scenario	Edit user info and default organization
Test case description	When the user enters user info page, he can edit his personal information beside choose his default organization to post in.
Pre-requisites	-The user device must have an internet connection - user should have signed in to the application - the user must have joined an organization
Test input	1- First name: Mostafa 2- Last name: Mousa 3- Phone: 0101212354 4- Organization: Test
Execution steps	1- Open the app 2- Click on “user page” in action-bar 3- Click on “user info” section 4- Click on “edit user information” 5- Proceed the post inputs 6- Click on “save information”
Expected behavior	The user information should be updated correctly, and default organization changes
Future Test	None
Actual results	The application successfully updates user information, and change the default organization
Status	tested

2.0 User acceptance Test

The test was done on 3 users from MSA university

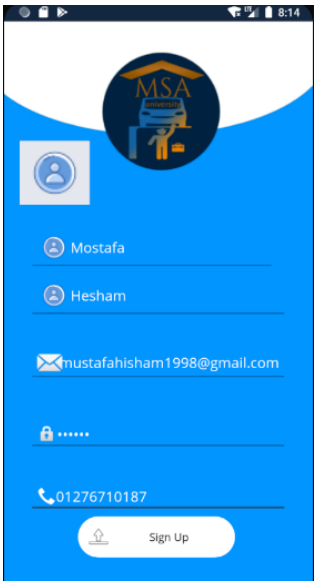
Number	Requirement	Acceptance requirement	Critical Yes/No	Test Result	Comments
1	RQ6	the layout of the system must be simple	No	Accepted	The layout was simple and the colors were soft on eyes
2	RQ1	the home page must be simple and don't required any practice to understand	Yes	Accepted	None
3	RQ2/RQ3	When creating a new post, the user should visualize the location and destination he is going to and the creating posts steps should be in individual page to minimize complicity	Yes	Accepted	The layout was very understandable to create a post in an easy way
4	RQ4/RQ5	Creating a new organization should have more information to distinguish organization from other	Yes	Accepted	It is very easy to create a new organization, but it needs a pic to be easy to recognize organization/facility from other without searching for the name Joining the organization is very simple, but in need to told me that's needed the same organization email structure
5	RQ9	Go from an organization to another should not contain many steps	Yes	Accepted	Very simple and recognizable
6	RQ11	None	Yes	Accepted	Works smoothly but in weak devices the app is heave in this section
7	RQ8	None	Yes	Accepted	Need a confirmation message when deleting
8	RQ13	View source or destination route should be in google maps app	Yes	Accepted	The app communicates with google maps app smoothly

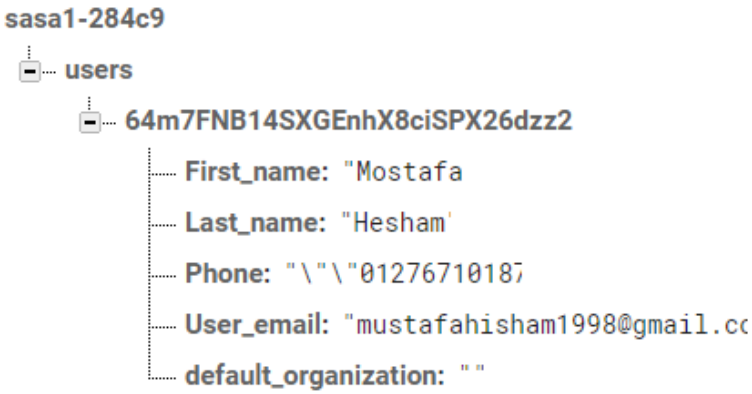
4.0 System Evaluation

(figure 11) these figures sow the outputs of the application

1- Sign up activity (Regiater_Activity class) **RQ 6_ T1**

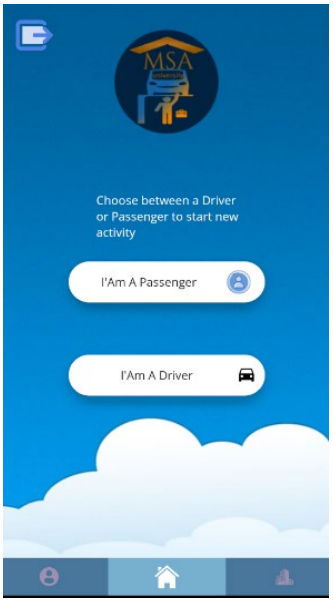
- the user enters the information to register and press “Sign UP” button (performing steps like in test case T1)
- The Fire base quarry successes and updated with the entered user data





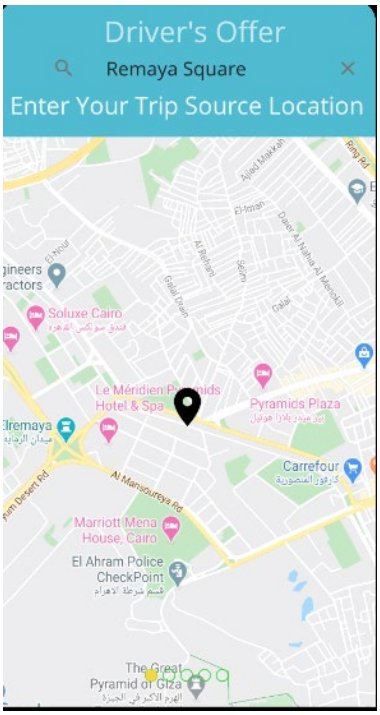
2- Display home Page (MainActivity Class) RQ 1_T5

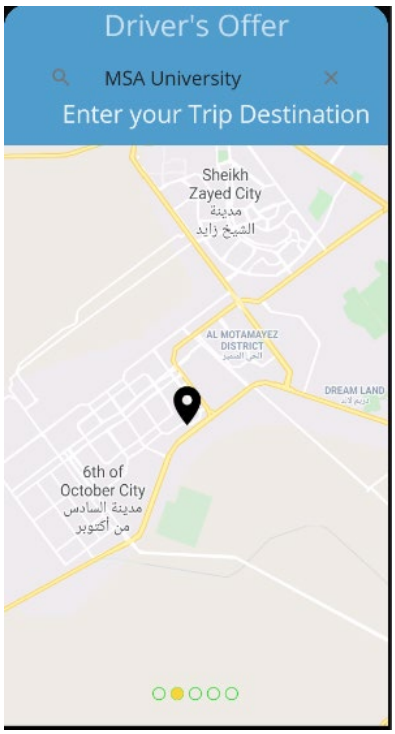
- The system displays the home page after user signup or sing in

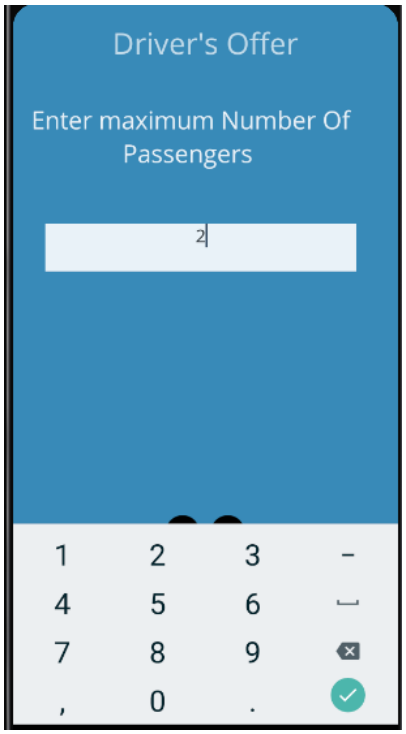


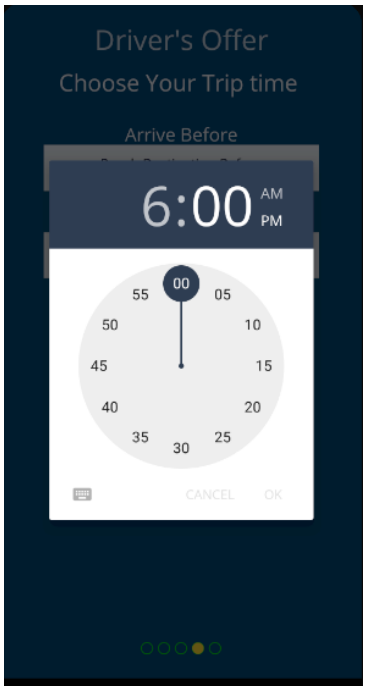
3- Make a Driver offer Post (Create_Post_Activity class) RQ2_T6 / RQ13

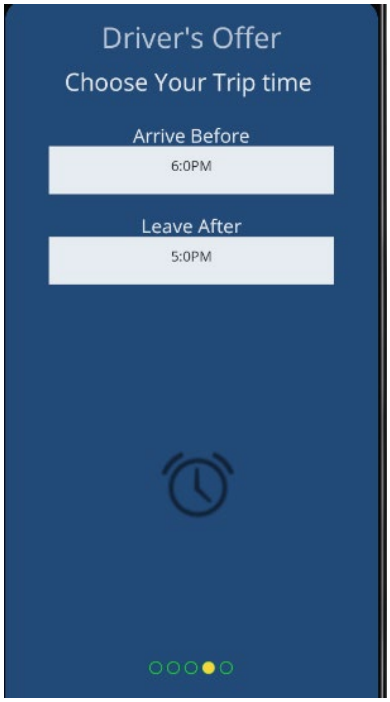
- the user performs steps like in Test case T6
- the application performs the Activity successfully that's displays correct result in the user inter face as expected
- The Fire base quarry successes and updated with the entered Post data for the user and organization

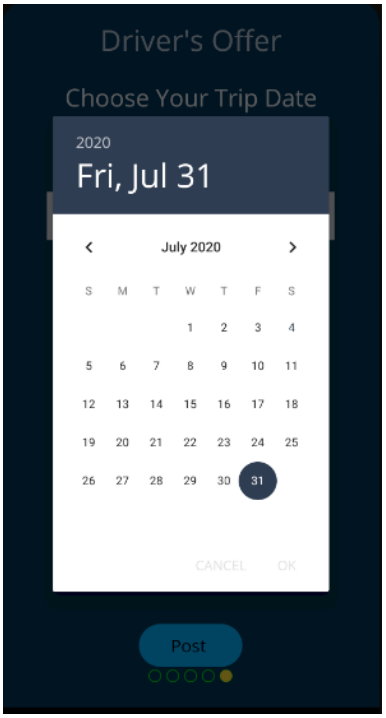


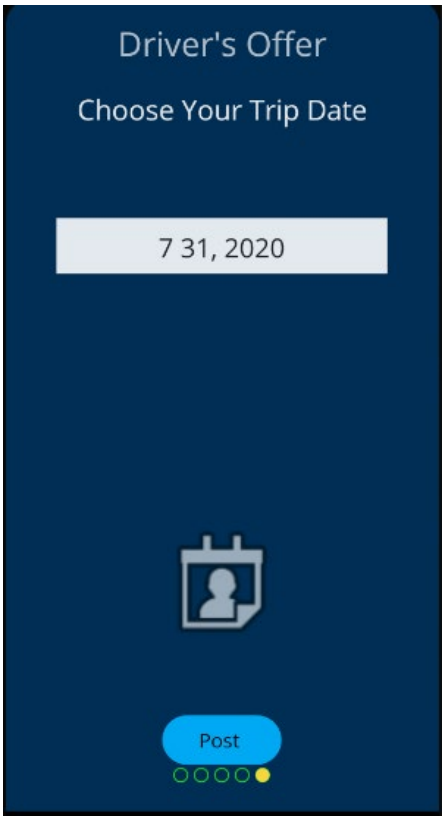


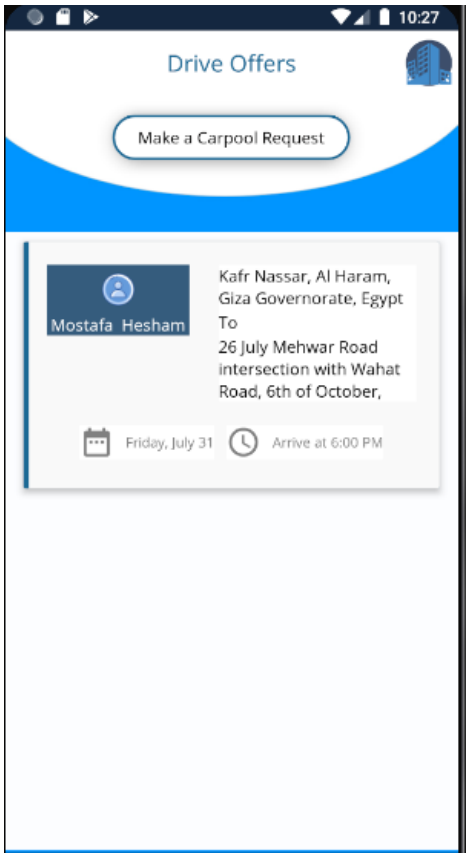






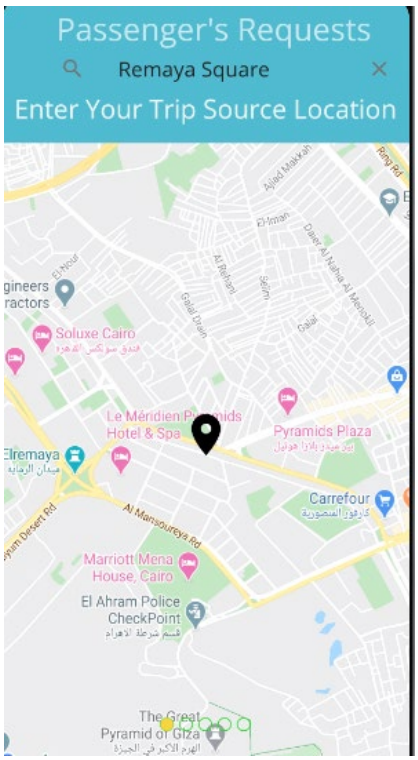


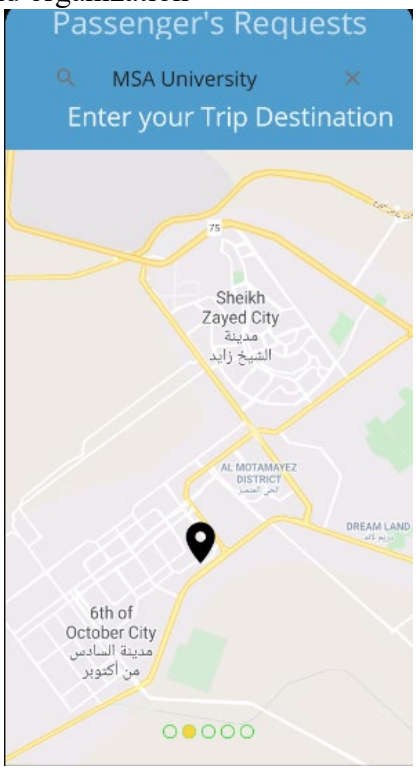


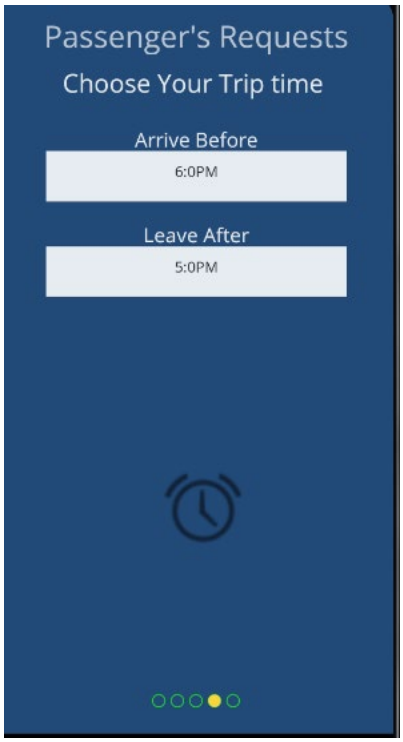


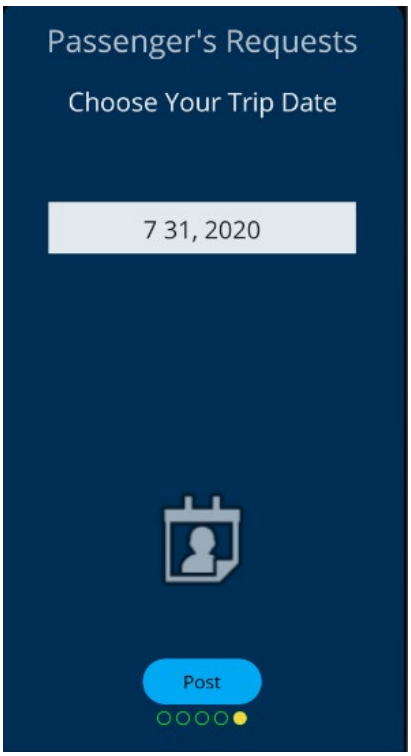
4- Make a Passenger Request Post (Create_Post_Activity class) RQ3_T7 /RQ13

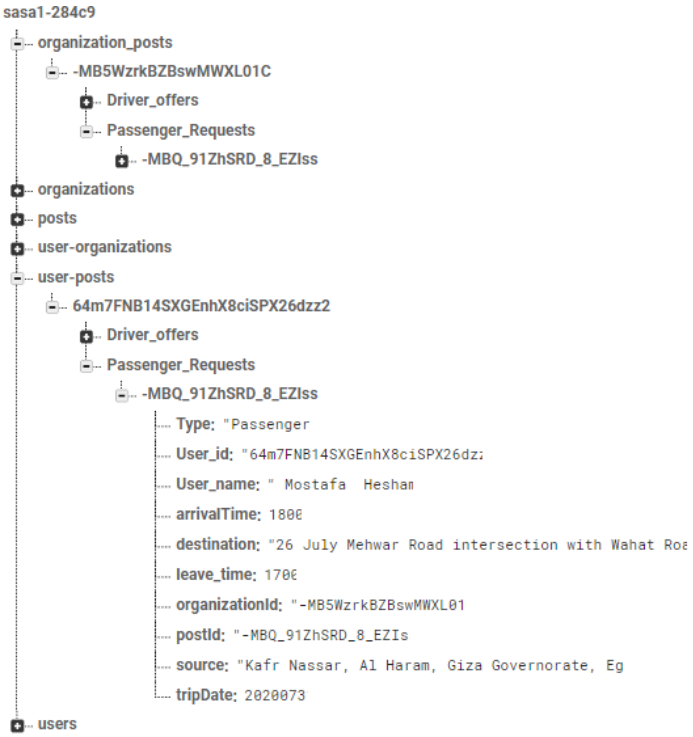
- the user performs steps like in Test case T7
- the application performs the Activity successfully that's displays correct result in the user inter face as expected
- The Fire base quarry successes and updated with the entered Post data for the user and organization





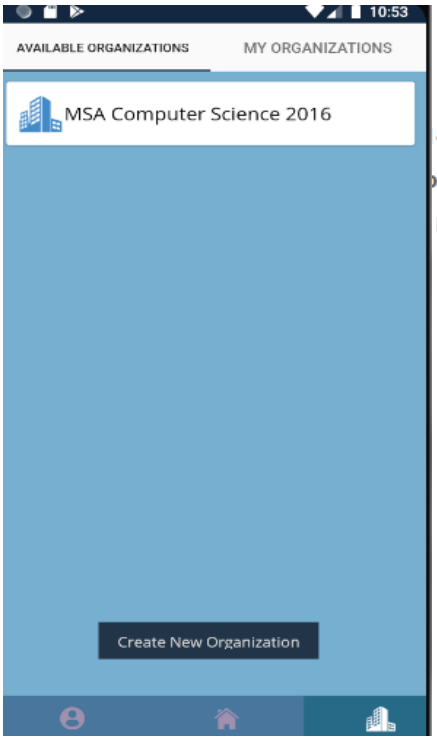


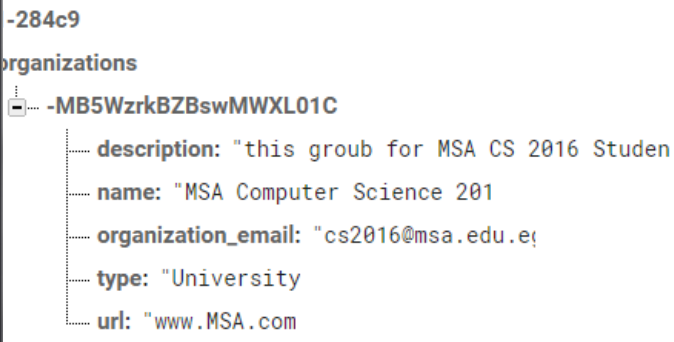


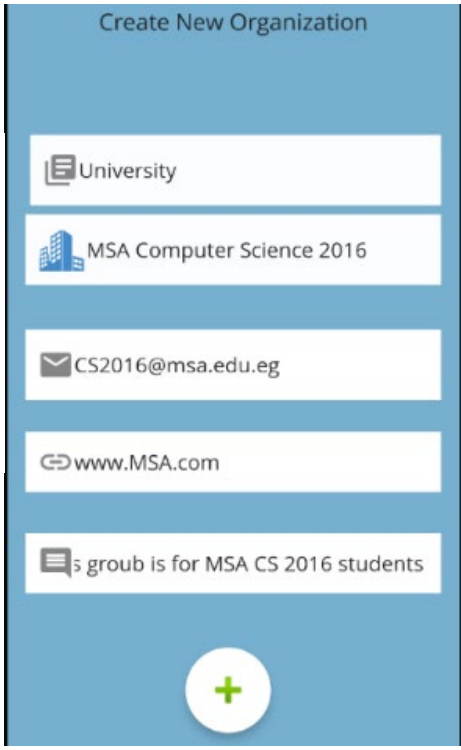


5- Create a new Facility/Organization group (Create_new_organization_Activity class) RQ4_T8

- The user performs steps like in Test case T8
- the user enters organization the Press Create button "+ "
- The Fire base quarry successes and updated with the entered Organization data
- User interface updated with the created organization

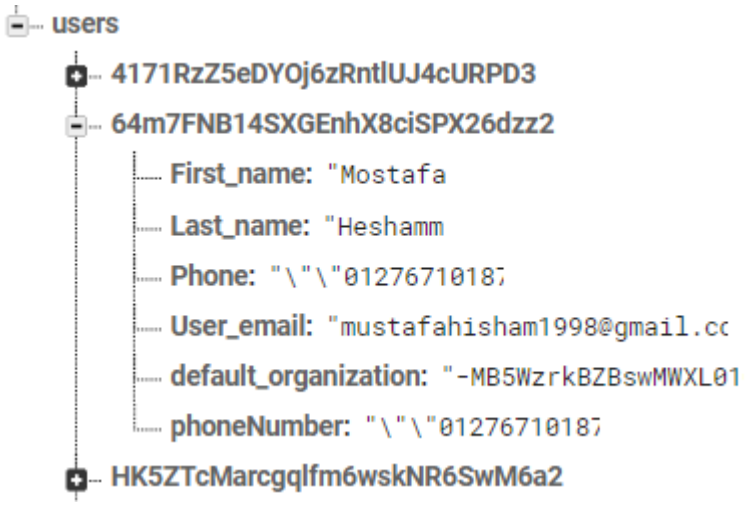


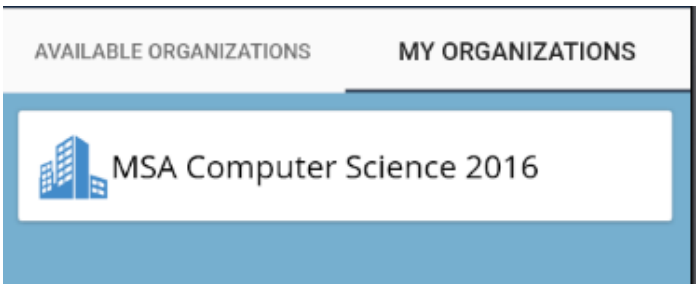




6- user joins an organization/Facility (join_organization_Activity class) RQ5_T9

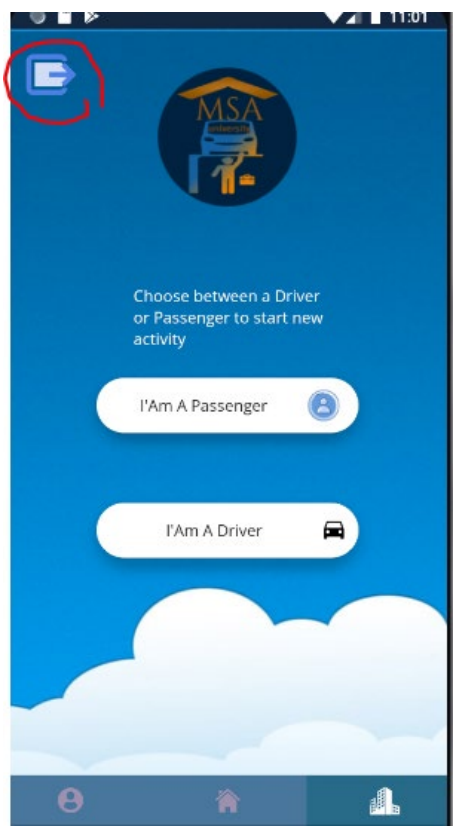
- the user enters the organization join page by click on its name
- the users enter organization email then press “join” button
- The Fire base quarry successes and update the user with the joined organization as a default
- User interface updated with the created organization (My Organization)

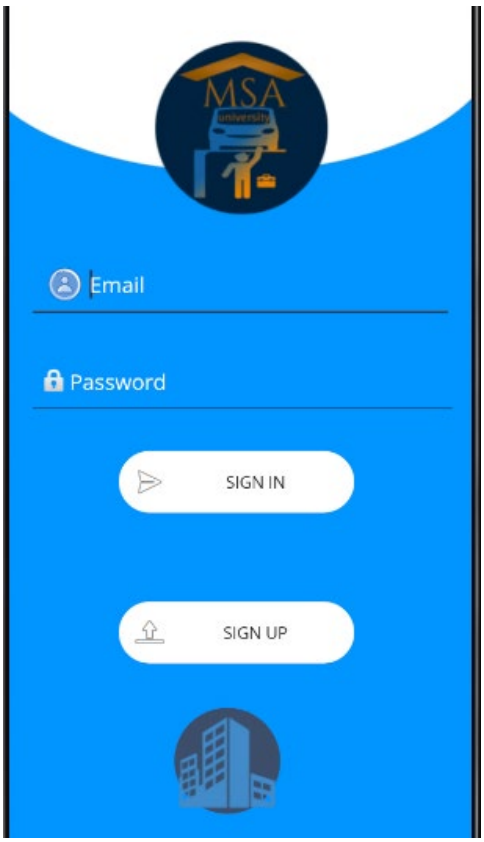




7- user logs out of the application RQ8_T10

- the user performs steps like in test case T10
- the system successfully logout the user out of the main page and goes directly to the sign in page





8- user displays the chosen organization posts only (onClick_load_spinner Function in passenger_ Driver_main_Activity) RQ9_T11

- the user performs steps in the testcase T11
- the system displays the joined user organization successfully and update the interface with only the organization posts as chosen

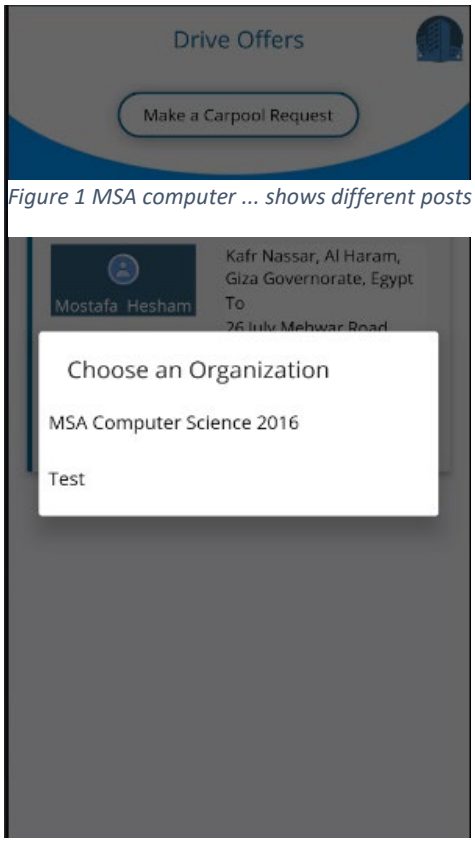
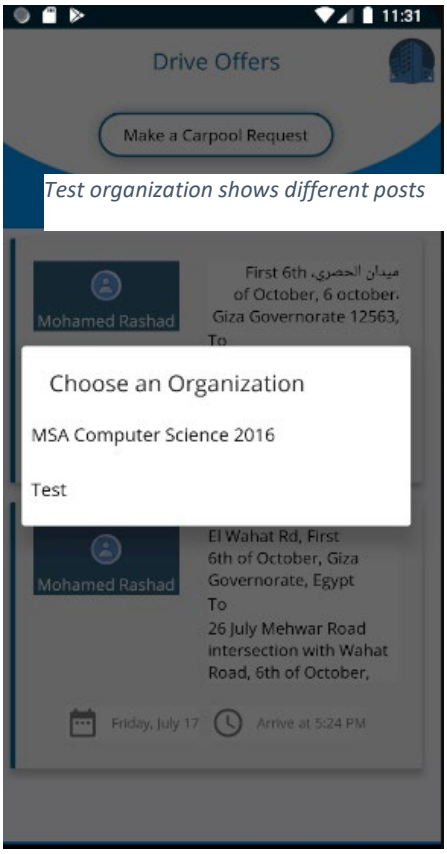


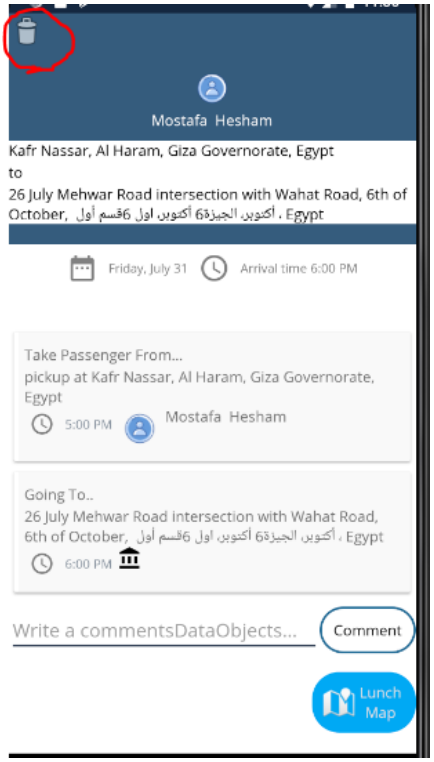
Figure 1 MSA computer ... shows different posts



Test organization shows different posts

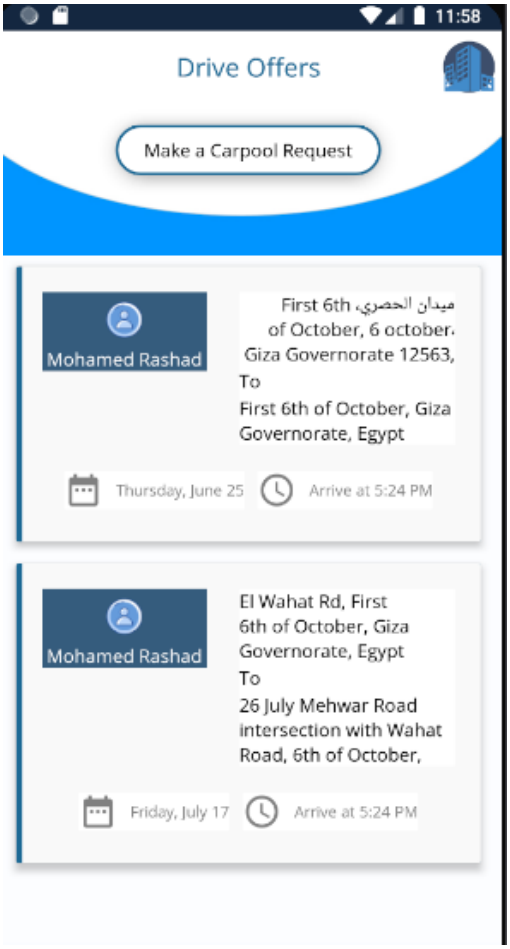
9- Deleting created Post RQ10_T12

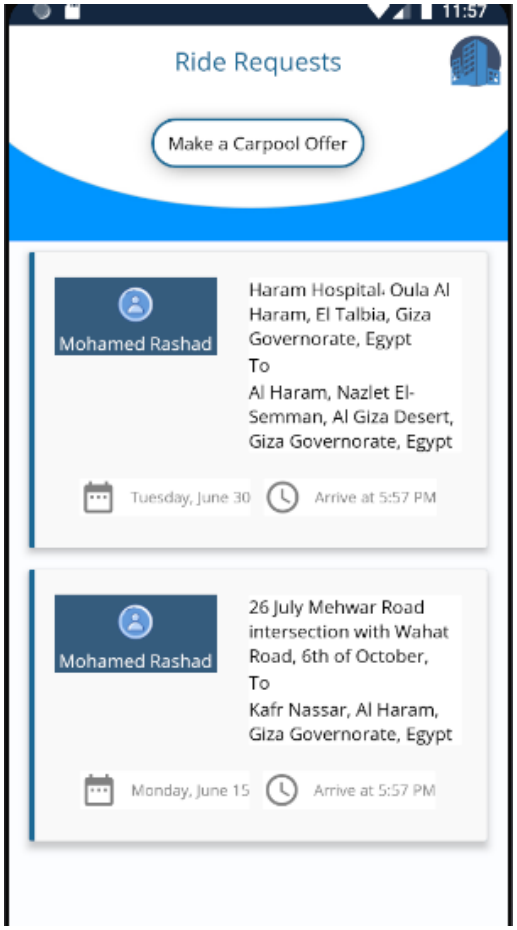
- The user performs the steps as showed in test case T12
- After open the post from tow of the mentioned process in T12 the post is deleted successfully form the interface
- Post data deleted successfully from fire base with the Id (MBQVh4w5c3H_zGo_t2g) as showed before in the previous test and still in the user_post data to be saved for the RQ 12



10- Browse posts RQ10,11 T13

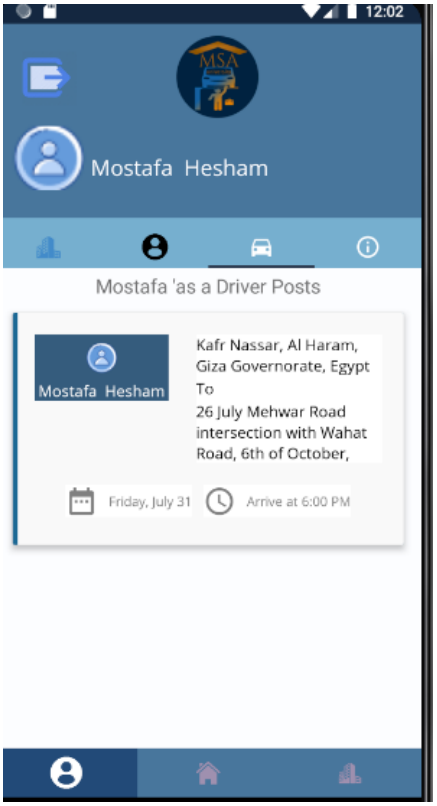
- The user performs the steps as shown in test case T13
- The system successfully displays the passenger and driver posts





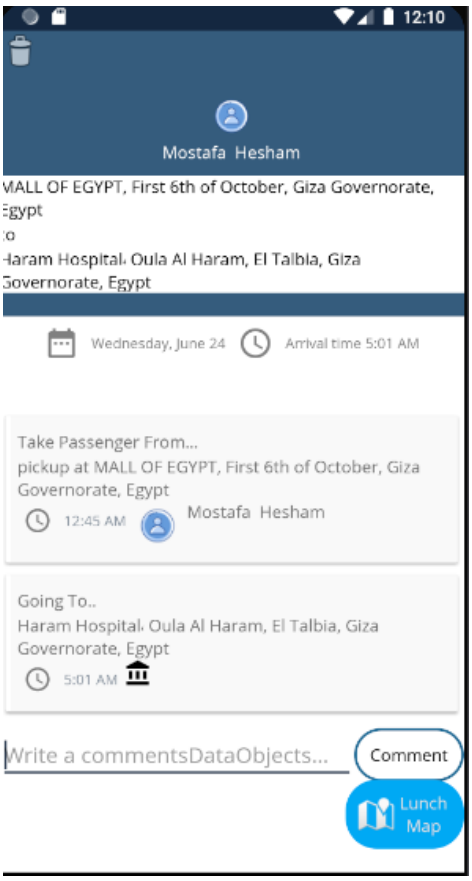
11- View all of the past created posts User_Page_Activity class RQ12_ T14

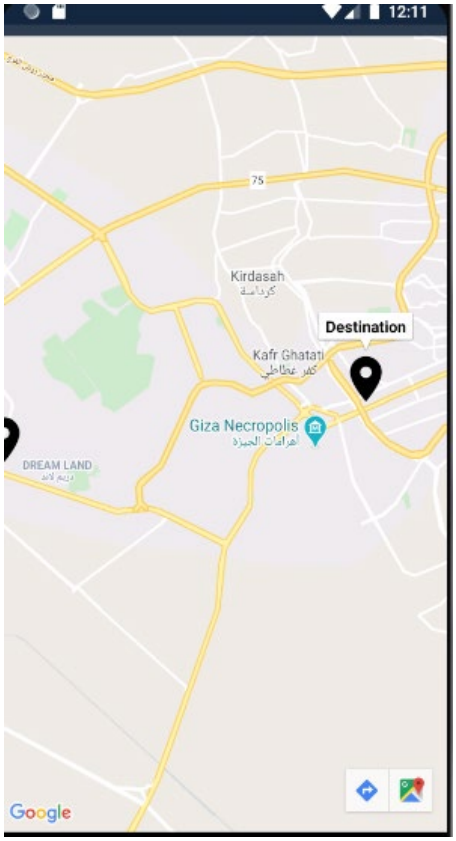
- The user performs the steps as shown in test case T14
- The system successfully displays all of the user posts and organization in user information page

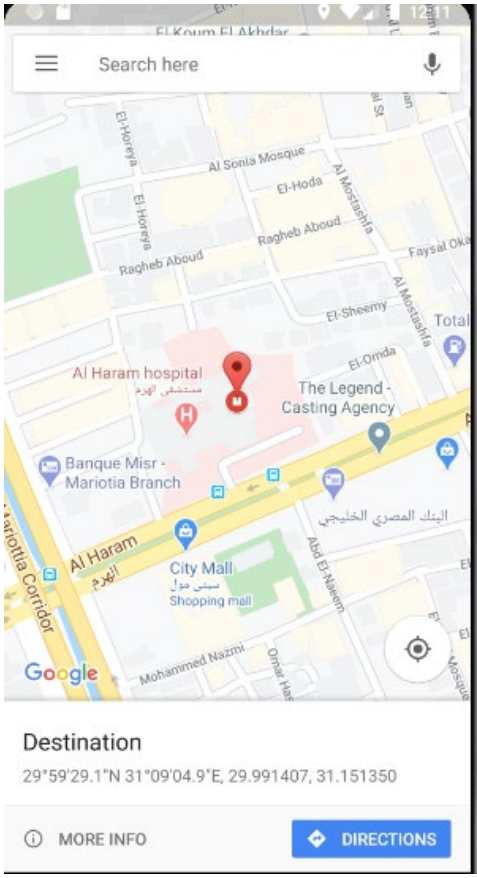


12- View entered location and destination on map Post_information_Activity class RQ13_ T15

- The user performs the steps as shown in test case T15
- The system successfully displays the active location on map when creating post or show the route on a pre created post

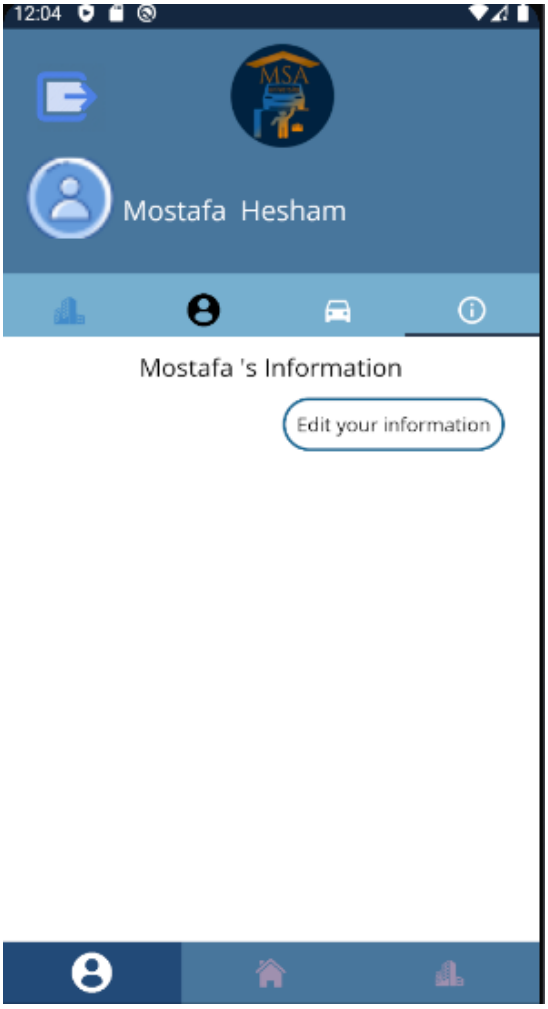


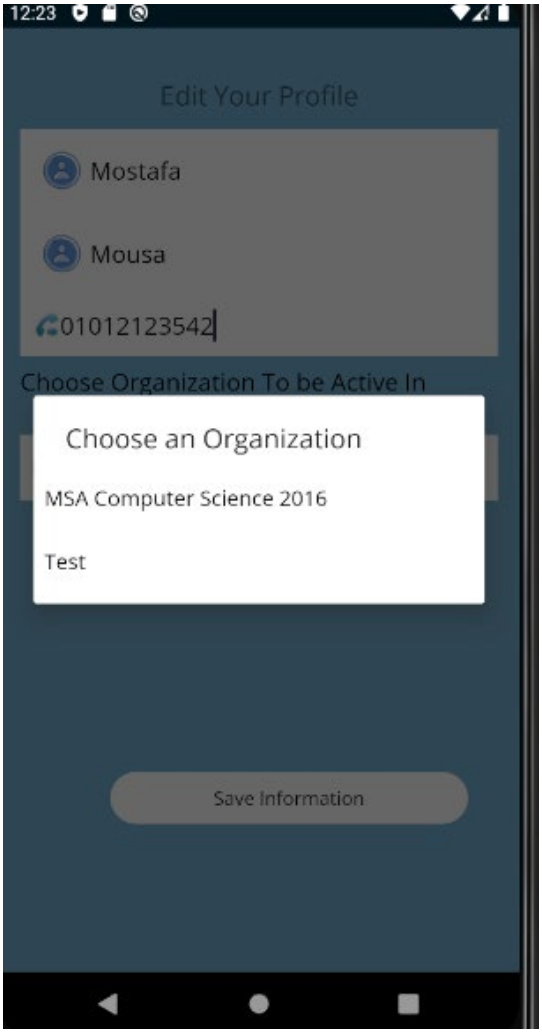


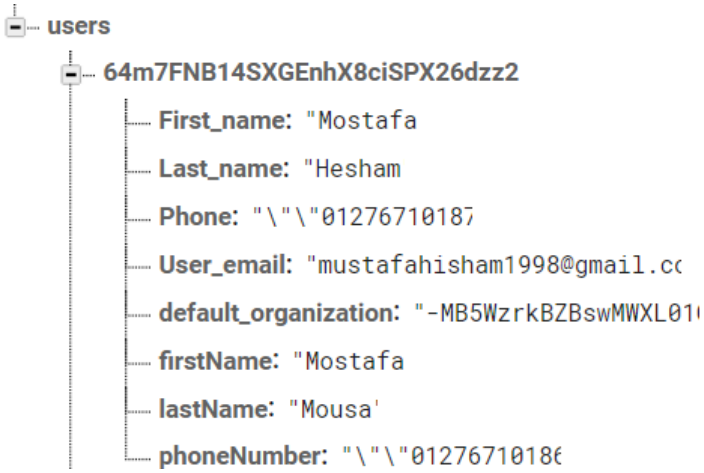


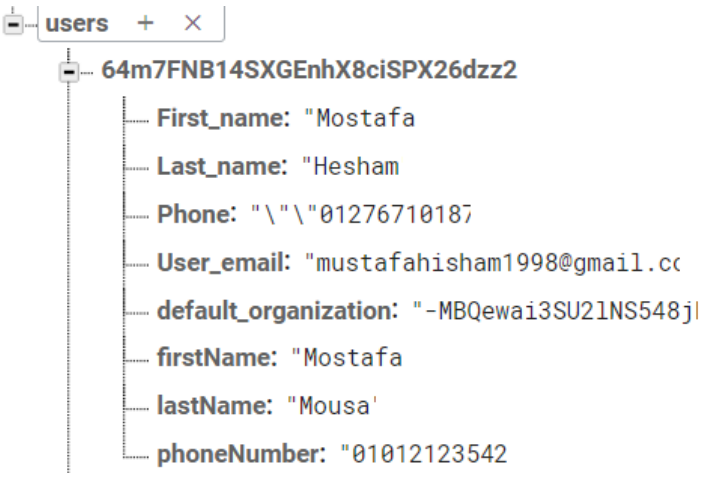
14- Edit user info and default organization Edit_user_information_Activity class RQ15_ T17

- The user performs the steps as shown in test case T17
- The system successfully displays the user with the new data
- Firebase update user information successfully









Chapter 5
Conclusions and future work

6.1 Conclusion

This Mobile application is developed to help the members of any (Facility, Organization, University, school…etc) to communicate with each other in the most efficient and organized way to reach their desired place in the right time and least cost and most comforts. The application allows users to make a carpooling activity but in more efficient way for the drivers and passengers beside lowering environmental pollution. The application allows the user to pre schedule their trip and sharing it with other users that's are from the same Organization or group they belong to. The application is most efficient for those users than Uber, Careem, Sawa because those applications are like Taxi Activity concept but only uses live tracking system, the cost is high like a private taxi for most users, and as a Driver it's a concept of delivering someone to his desired place not to the Driver's already "going to" place which is not ideal to a University student or any employer whose going to work.

The application works by allowing the user to join group which is an facility/Organization group which they belong to by authenticating the user with his organization mail, the user can join more than one organization and choose between them when posting or viewing posts, the user then can make a carpooling post by choosing his location and destination beside Date and time of the carpooling and share his post with all of other organization users, the user can choose his carpool ride type that's between a driver offer which allows the user to be the driver for the carpooling, or a passenger request post which the user is a passenger with others in the carpooling, the application allows the user to communicate with each other through third party chatting app by their phone numbers, or through firebase notification messages.

After finishing designing and implementing the application functions and features, a several tests was done to ensure the usability and functionality of the application in every main activity, White-Box testing was the way to go because it's the most extensive way in terms of collecting the user feedback and acceptance in terms of application design and functionality beside getting the firebase Database view and outputs when user enters an activity or input, beside checking how well the.

The application has proven to give an important functionality to the user that's takes a part in some facility or organization that was missed in other carpooling app, and find new ways to enhance the user's experience whose like the idea of carpooling as not only as passenger but also as Driver to be effective for him in terms of cost and time.

6.2 technical issues:

- The application crashes if not entering the pickup/point and move to next step in passenger request post.
- The application is heavy and slow in the devices with weak CPU (tested on emulator)

6.3 Future work

Many improvements can be added to application to make it a more of a complete system.

- Adding a chat room for the organization and private chat room between users rather than using third party app
- Adding live tracking system for the users in active carpool
- Implementing online payment to the application
- Adding rating/voting system on users.
- adding editing function to the posts

7 References

References

1. Burris, M. and J. Winn. 2006. Slugging in Houston— Casual Carpool Passenger Characteristics. Journal of Public Transportation 9 (5): 23–40.

2. C. Oman, "General Census of Population Housing & Establishments", *Nation Centre for Statistics & Information 2010*, [online] Available: http://www.ncsi.gov.omINCSI_website/documents/Census_2010.pdf.

3. Ciari, F. (2012). Why do people carpool? Results from a Swiss survey. Retrieved from http://www.strc.ch/conferences/2012/Ciari.pdf

4. Correia, G., Viegas, J.M. (2006) 'Carpooling clubs: solution for the affiliation problem in traditional/dynamic ridesharing systems', Proceedings of the 11th Meeting of the EURO Working Group on Transportation, Bari, Italy,September.

5. EPA, "Transportation Benefits: Carpooling,"http://www.epa.gov (cited May 2006)

6. Estimating the Energy Consumption Impact of Casual Carpooling (2011, January), Retrieved September 19, 2017.

7. Handy, S., L. Weston, and P. Mokhtarian. 2005. Driving by Choice or Necessity? Transportation Research Part A, Vol. 39, No. 3., pp. 183-203.

8. Horowitz, A., Sheth, J. Ridesharing to work: an attitudinal analysis. (1978).

9. B.-A. Hartman, D. Keren, A. A. Dbai, E. Cohen, L. Knapen, D. Janssens et al., "Theory and practice in large carpooling problems," Procedia Computer Science, vol. 32, pp. 339–347, 2014.

10. LBSN '10 Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks Pages 51-59 https://dl.acm.org/citation.cfm?id=1867710

11. M. Islam, A. Al Hadhrami, "Increased Motorization and Road Traffic Accidents in Oman", *Journal of Emerging Tre nds in Economics and Management Sciences (JETEMS)*, vol. 3, no. 6, pp. 907-914, 2012.

12. Maniezzo V., Carbonaro A., Hildmann H. (2004). An ants heuristic for the long-term carpooling problem. In: Onwubolu G, Babu BV (Eds.).

13. Proceeding SPAA '11 Proceedings of the twenty-third annual ACM symposium on Parallelism in algorithms and architectures Pages 167-176 San Jose, California, USA — June 04 - 06, 2011 https://dl.acm.org/citation.cfm?id=1989518

14. Shaheen, S., A. Cohen, I. Zohdy, and B. Kock. 2016. Smartphone Applications to Influence Travel Choices: Practices and Policies. U.S. Department of Transportation. www.ops.fhwa.dot.gov/publications/fhwahop16023/fhwahop16023.pdf

15. Shaheen, S., N. Chan, and T. Gaynor. 2016. Casual Carpooling in the San Francisco Bay Area: Understanding Characteristics, Behaviors, and Motivations. Transport Policy 51: pp. 165-173.
16. Teodorovic D., Dell'Orco M. (2008). Mitigating Traffic Congestion: Solving the Ride-Matching Problem by Bee Colony Optimization. *Transportation Planning and Technology*, Vol. 31, No. 2, pp. 135-152.