[Settle Mobile Application]

SETTLE

WHERE YOU SETTLE IN!

Helwan
University

Faculty of computers and information
Information Systems Department

Supervised by:

DR. Manal Abdel-Kader

Team Members

Moaaz
Muhammad

Asmaa
Hamdy

Muhammad
Ashraf

Muhammad
Omar

Shirin
Mahmoud

Muhammad
Atef

ACKNOWLEDGEMENT

First and foremost, we would express our deepest gratitude and appreciation to our advisor **Dr. Manal Abdel-Kader** for her support, outstanding guidance and encouragement throughout our graduation project.
As well as **Engineer. Amr Essam** for his continuous support and help as he was always there for us providing more than the needed time and effort from him since the very beginning of our project.

We would also like to thank our families, especially our parents, for their encouragement, patience, and assistance over the years.
We are forever indebted to our parents, who have always kept us in their prayers and pushed us to Success.
Finally, for our faculty for providing the suitable environment that leaded us to represent the best image that computer science graduates of Helwan University are supposed to represent.

Table of Contents

List of Tables

List of Figures

Abstract:-

Main idea:-

This program solves a daily social problem in all societies, called the issue of the exploitation of brokers through blackmailing the customers by establishing an application for social communication that cares for this category of society which is not small

Technologies and process:

Structure:

Social media app with GPS and advanced location technologies.

The application is an application of social communication very normal, such as Facebook and Twitter, but solves the issue of brokers' exploitation by enabling the user to choose among the available estates appearing on the map around him.

Technologies

Firebase DB cloud service, Angular Ionic location plugin

We used firebase cloud services instead of web services to store data from users like images, location and info of posts

We used Angular Ionic location plugin to locate the estates and makes it easier to search among them on the map

The application shows the available estates for rent or sale on the map.

Process:

Register, login, add post, search, chat

- First: you should register by entering your info (name, email, address,…)
- Then you can login to the application by E-mail &password
- You can view posts in home page or visit your profile
- You can add post by taking or uploading an image to the app and set your location
- You can search for an estate through the GPS or the posts.
- You can chat with the estate owner.

Chapter 1: Introduction

In this chapter we're going to discuss and go deeper in the overview of the project and know more about its scope and limitations, and explain some terminologies we will find throughout the document.

1.1 Overview

It's a mobile application that enables the user to find estates (flats, shops .. etc) for rent or sale by entering the desired location and the application shows the nearest possible estates based on the category the user entered with the help of GPS.

The App has two users:-

- The first one:- the one who searches for an estate
- The second one:- the one who proposes his estate for sale or rent

3.5 Objectives

- The App shows the available estates for sale or rent
- The owner shows his estate details about it with no need to a broker
- The customer searches the GPS to find the desired estate in the location he chooses
- The App connects the customers to the estate owner with no need to a broker "which is the App goal"

3.5 Purpose

The App mainly tries to make a connection between the customer (the one who searches for an estate for rent or sale) and the estate owner with no need to brokers.

3.5 Scope

The approximate work involved to finish the project is divided into these five phases:-

1. <u>Planning:-</u>
   - Collecting data about the project and the lack that made us in a need to the App.
   - Making surveys in different ways (video, questionnaires, forms … etc).
   - Determining the functional and non-functional requirements.
   - Setting a Gantt chart for the project.
   - Determining the resources of the team

2. <u>Designing :-</u>
   - Determining the diagrams to be carried out within the project:-
     ERD Diagram
     Activity Diagram
     Class Diagram
     Use-case Diagram
     Sequence Diagram
     Context Diagram
     State Diagram

3. Coding

- The main supposed functions to be coded in this App are:-

  -Sign up

  -Login

  -Detect Location

  -Add Estate

  -Update Estate

  -Delete Estate

  -Block User

  -Comment

  -Rate

  -Feedback

  -Send a message

  -Search

  -Log out

  -Rent Estate

  -Buy Estate

4. Testing:-

- Functional Testing:-

- Unit testing
- Regression testing
- Integration testing

- Non-functional Testing:-

- Performance testing
- Street testing
- Security testing

5. Documentation:-

The documentation should mainly include these main chapters:-

- Introduction: includes an overview of the project and limitations
- Project Planning: includes the tools and technologies as well as tasks and timeline plan
- Project Requirements: includes the functional and non-functional requirements
- Project Design: includes the diagrams
- Project Implementation: includes user application and administrator system
- Project Testing: includes testing types

3.5 General Constraints

- Tasks division which can be not fair enough.
- Indiscipline “Human factor” like being late in delivering tasks or attending meetings
- Hesitation, especially when it’s related to taking a serious step or learning a new technique
- Time management
- Learning new technologies may take much time
- Underestimating the objectives that may not lead to realistic or achievable function

Chapter 2: Project Planning and Analysis

In this chapter we’re going to discuss and go deeper in how we plan the project and show the steps and the instructions that we’ve followed to plan the application.

2.1 Project Planning

2.1.1 Feasibility study

A Feasibility study is used to determine if a business or a specific project is achievable, so for determining the achievability of our project we'll go deeper in the following points:-

1- Market Analysis

- Our app exist our in technical market especially in the market of real estate.
- We can make money from our app using ADS.
- Our app has more than a competitive edge like:

  1- Settle mobile app make users rent or buy estate from homes by just few clicks.

  2- Settle mobile app uses GPS to rent or buy your flat and this point makes renting easier because it does not force users to know the places or street names which they need to rent or buy in.

  3- Settle mobile app does isn't in specific limited places.

  4- Settle mobile app contains the part of social media to make users communicate with each other and exchange experience.

  5. Settle mobile app has the part of ranking the result of the search according to your priorities.

2-Organizational Analysis:-

- The number of employees in our system :-
  We have 9 employees:
  = 2 Admins.
  = 5 Developers.
  = 2 Testers

- Payment for your employees?

  3,000 L.E for each Developer and tester.
  2,000 L.E for each Admin.

- The coverage we need to generate and support our sales:
  We upgrade the application by the developers.
  We make the application better from the rate and feedback from customers when they do the transactions.

3-Operational Analysis:-

- In our project we provide some rules and services to our employees such as:
    1. Environment suitable for work
    2. Commitment to the deadline for delivering tasks
    3. Good salaries
- We deliver our service by using social media, offline marketing and advertisements.
- we used Some equipment like:
    - Microphone: 100 LE
    - Camera: 150 LE
    - Purchasing programs like:
        1. Adobe Premiere
        2. Adobe Photoshop
        3. Obtaining a hosting server
- We plan to support and retain our customers by providing feedback and promotions offers.

2.1.2 Estimated Cost

A cost estimate is approximation of the cost of a program, project or operation .the cost estimate is the product of the cost estimating process and our estimated cost for this project comes as following:-

In our project we had a financial issue like:

1. Meeting once a week in a public workspace : (100 LE) (from august to January 100*15 including vacations and holidays = 1500 LE)

2. Some resources and programs we used to make video survey and presentation like:

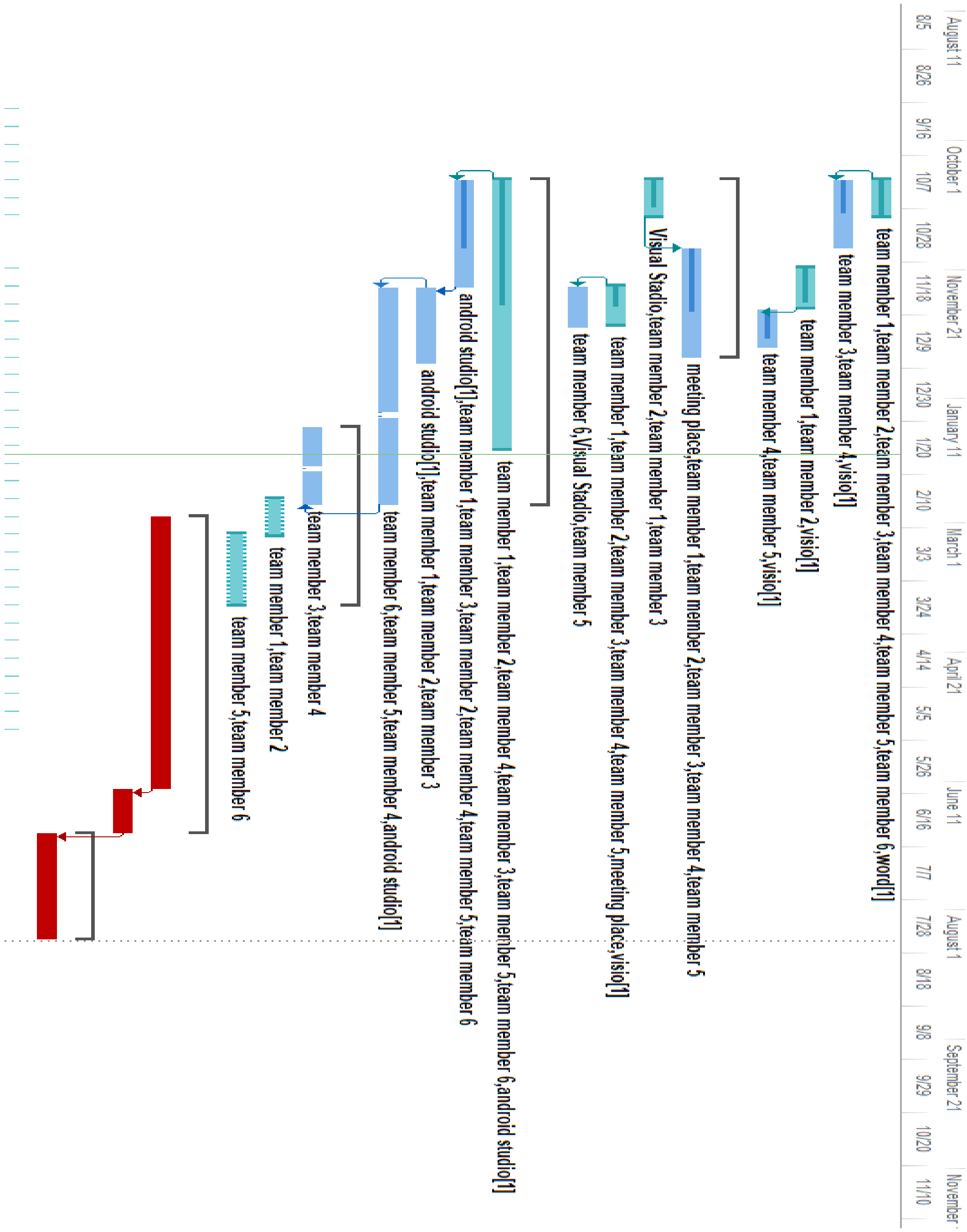
- Microphone: 100 LE
- Camera: 150 LE
- Making advertisements : 500 LE
- Purchasing programs like:
  1. Adobe Premiere
  2. Adobe Photoshop
  3. Obtaining a hosting server

4. We spent a lot of time and effort to get the best services and functions to be included in our app

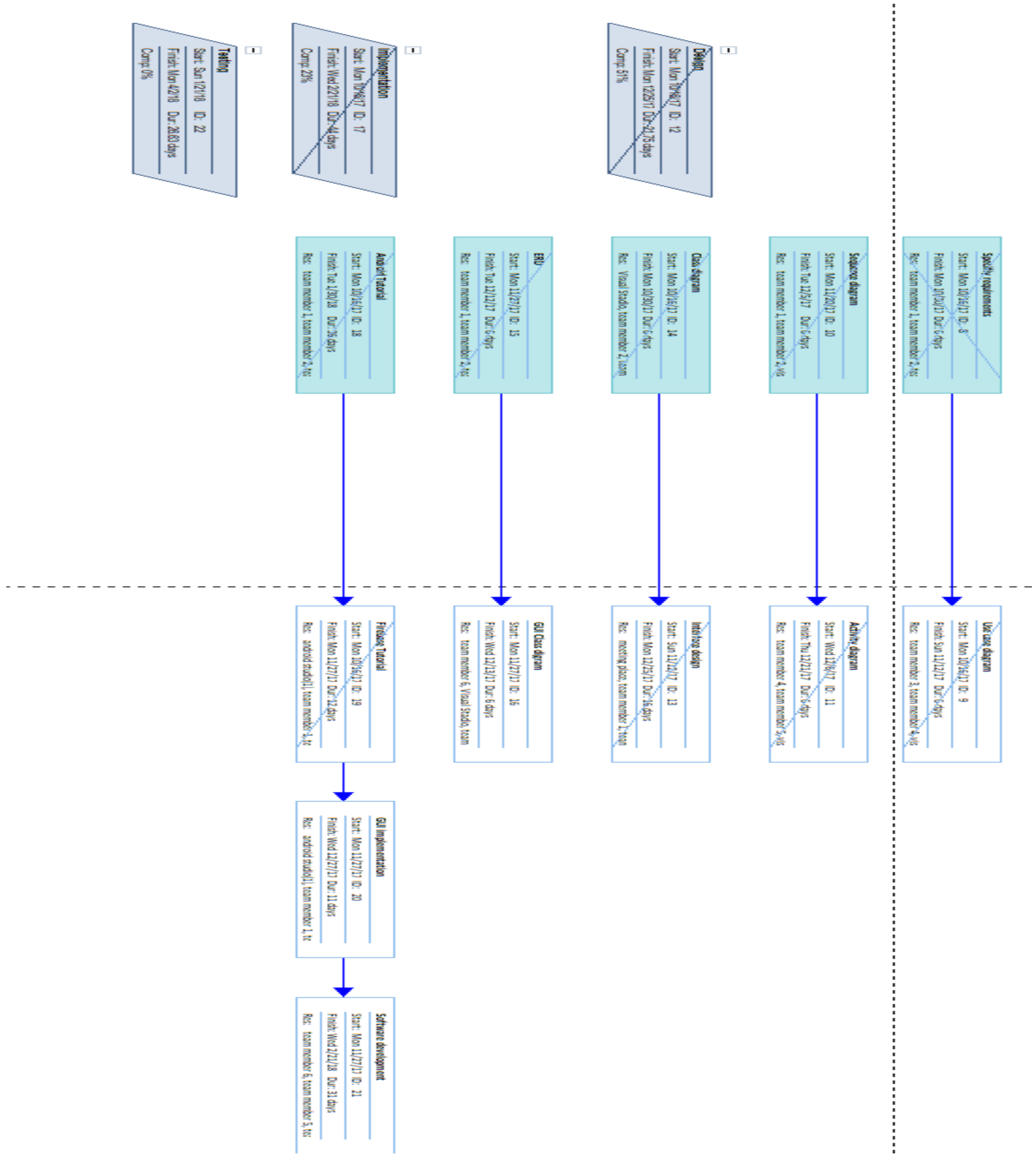
5. Salaries paid to employees

Ex: admin: 2000 LE
Tester: 2500 LE
Developers: 3000 LE

2.1.3 Gantt Chart



NETWORK DIAGRAM


Testing
Start: Sun 12/1/18 ID: 22
Finish: Mon 4/2/18 Dur: 26.63 days
Comp: 0%
Implementation
Start: Mon 10/16/17 ID: 17
Finish: Wed 2/21/18 Dur: 44 days
Comp: 22%
Design
Start: Mon 10/16/17 ID: 12
Finish: Mon 12/25/17 Dur: 21.75 days
Comp: 61%
Android Tutorial
Start: Mon 10/16/17 ID: 18
Finish: Tue 1/30/18 Dur: 36 days
Res: team member 1, team member 2, res
ERD
Start: Mon 11/27/17 ID: 15
Finish: Tue 12/12/17 Dur: 6 days
Res: team member 1, team member 2, res
Class diagram
Start: Mon 10/16/17 ID: 14
Finish: Mon 10/30/17 Dur: 6 days
Res: Visual Studio, team member 2, team
Sequence diagram
Start: Mon 12/20/17 ID: 10
Finish: Tue 12/5/17 Dur: 6 days
Res: team member 1, team member 5, res
Specify requirements
Start: Mon 10/16/17 ID: 8
Finish: Mon 10/30/17 Dur: 6 days
Res: team member 1, team member 2, res
Firebase Tutorial
Start: Mon 10/16/17 ID: 19
Finish: Mon 11/27/17 Dur: 12 days
Res: android studio[1], team member 4, tr
GUI Class diagram
Start: Mon 11/27/17 ID: 16
Finish: Wed 12/13/17 Dur: 6 days
Res: team member 6, Visual Studio, team
Interface design
Start: Sun 11/12/17 ID: 13
Finish: Mon 12/25/17 Dur: 16 days
Res: mocking plus, team member 2, team
Activity diagram
Start: Wed 12/6/17 ID: 11
Finish: Thu 12/21/17 Dur: 6 days
Res: team member 4, team member 5, res
Use case diagram
Start: Mon 10/16/17 ID: 9
Finish: Sun 11/12/17 Dur: 6 days
Res: team member 3, team member 4, res
GUI implementation
Start: Mon 11/27/17 ID: 20
Finish: Wed 12/27/17 Dur: 11 days
Res: android studio[1], team member 1, te
Software development
Start: Mon 11/27/17 ID: 21
Finish: Wed 2/21/18 Dur: 31 days
Res: team member 6, team member 5, res

App logo	App name	GPS	Used in any place	Has part of social media	Specific in estates
	عقارات مطروح	F	F	F	T
	سمسارمصر	F	T	T	T
	السوق المفتوح	F	T	T	F
	الوسيط	F	T	T	F
	OLX	F	T	T	F

SETTLE WHERE YOU SETTLE IN!	Settle	T	T	T	T

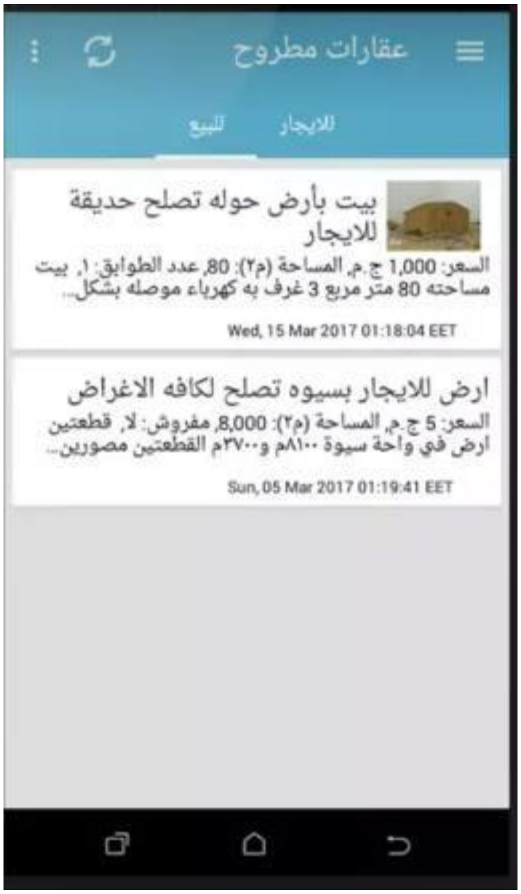
2.2 Analysis and limitations of existing system

عقارات مطروح

Description

-Features

This app used to help users to buy and rent flats easily , this app categorizes flats to two categories the first one is to buy and the second is to rent all this to make usability in the app and this app contains only homes , flats and lands .



بيت بأرض حوله تصلح حديقة للايجار

السعر: 1,000 ج.م, المساحة (م٢): 80, عدد الطوابق: ١, بيت مساحته 80 متر مربع 3 غرف به كهرباء موصله بشكل..

Wed, 15 Mar 2017 01:18:04 EET

ارض للايجار بسيوه تصلح لكافه الاغراض

السعر: 5 ج.م, المساحة (م٢): 8,000, مفروش: لا, قطعتين ارض في واحة سيوة ٨١٠٠م و٣٧٠٠م القطعتين مصورين..

Sun, 05 Mar 2017 01:19:41 EET

-Shortfall

1. This app does not use GPS to rent or buy a flat and this point is not good because it forces you to know all places that you want to rent in.

2. This app only used to rent or buy flats from Matrouh.

3. This app does not contain the part of social media to make users communicate with each other and transfer experience.

(2)

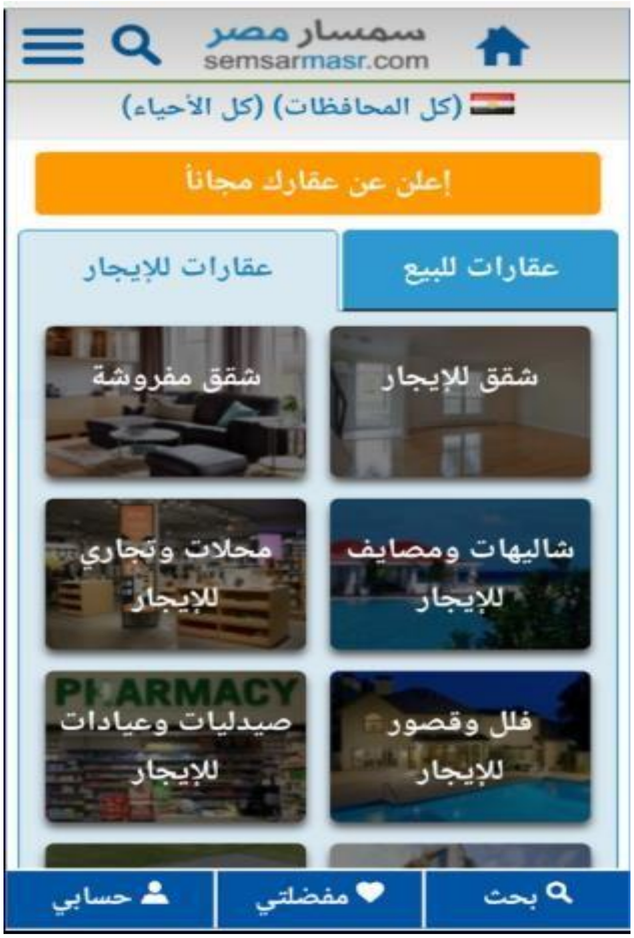
سمسار مرص

Description

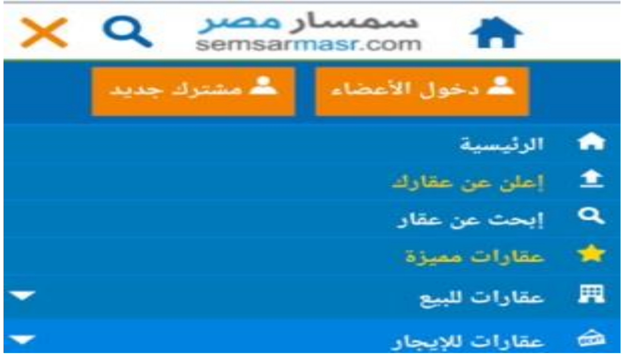
-Features

1. This app is used to help users to buy or rent flats easily this app categorizes flats to two categories the first one is to buy and the second

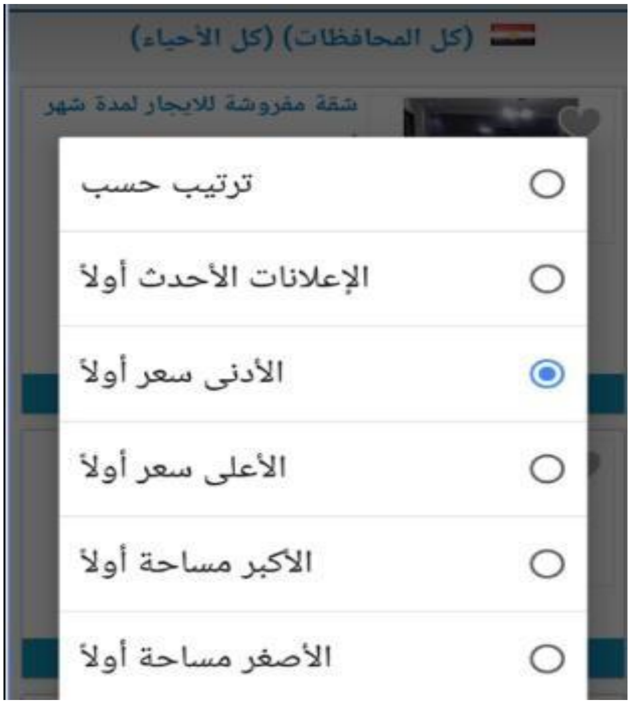
is to rent all this to make usability in the app , this app contains only homes , flats , lands and markets.



2. This app has the part of social media to make users communicate with each other and transfer experience.



3. This app has the part of ranking the result of the search according to your Priorities.



ترتيب حسب ○

الإعلانات الأحدث أولا ○

الأدنى سعر أولا ●

الأعلى سعر أولا ○

الاكبر مساحة أولا ○

الأصغر مساحة أولا ○

-Shortfall

1. This app does not use GPS to rent or buy a flat and this point is not good because it forces you to know all places that you want to rent in.

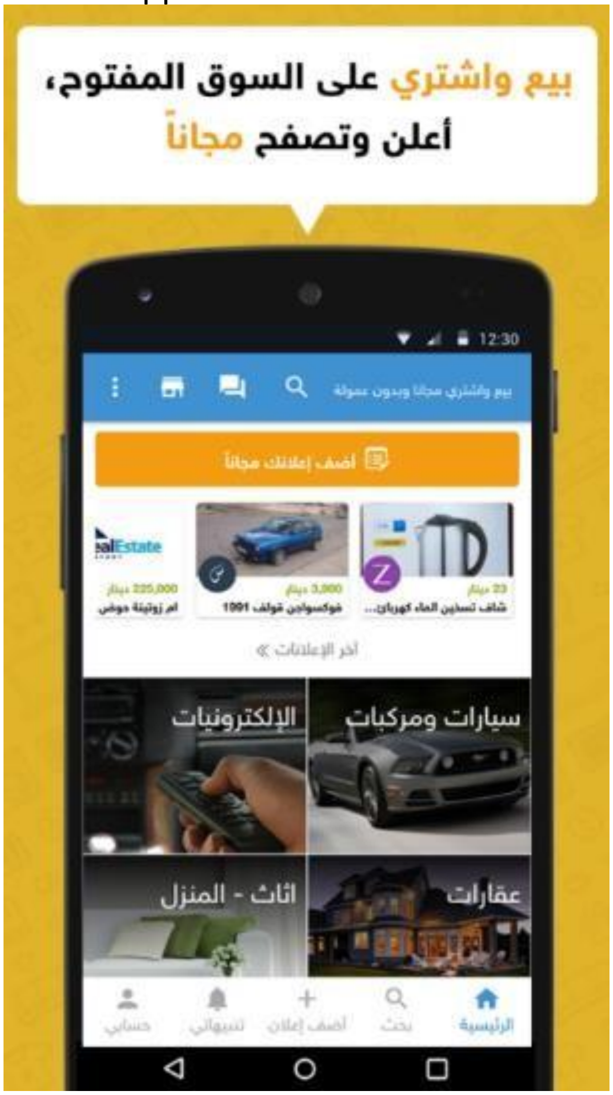
(3)

السوق المفتوح

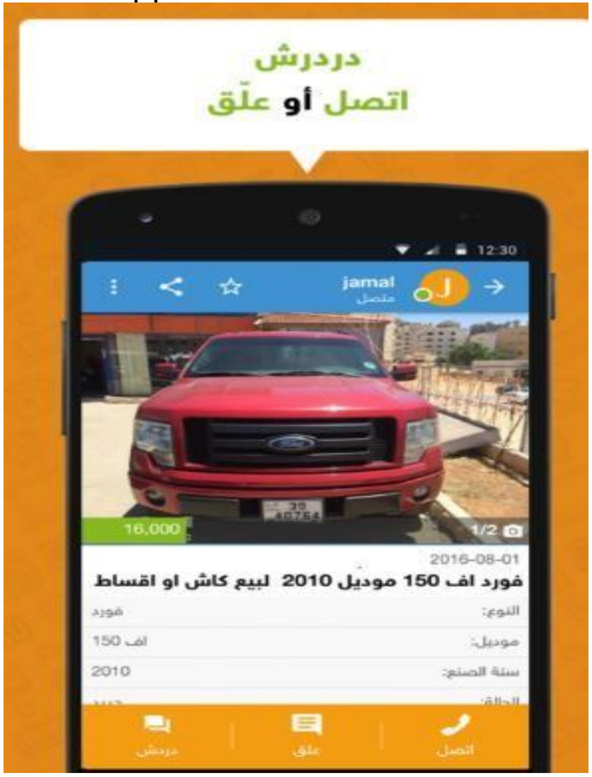
Description

-Features

1. This app is used to help users to buy or rent what you need not only flats but also cars, furniture and Electronic Devices.



2. This app has the part of social media to make users communicate with each other and transfer experience.



2016-08-01

فورد اف 150 موديل 2010 لبيع كاش او اقساط

النوع: فورد

موديل: اف 150

سنة الصنع: 2010

-Shortfall

1. This app doesn't use GPS to rent or buy or sell.

2. This app is not specific in buy or sell flats and homes only.

(5)

OLX

Description

-Features

1. This app is used to help users to buy or sell what they need.

2. This app do not used in specific places.

3. This app contains of the part of social media to make users communicate with each other and transfer experience.

-Shortfall

1. This app is not specific in buy or sell flats and homes only.

2. This app doesn't use GPS to buy or sell.

Our project

Description

- It is a mobile application that enables the user to find flats to rent or buy by entering the desired location and the application will show the nearest possible flats based on the category you entered with the help of GPS this application has 2 users
- The first one:- is the one who wants to buy or rent the flat.
- The second one:- is the one who wants to propose his flat or shop for sale or rent .

-Features

1. Our app uses GPS to rent or buy your flat and this point make rent easily because it do not force users know the places or street names which you need to rent or buy in.
2. Our app do not used in specific place.
3. Our app contain the part of social media to make users communicate with each other and transfer experience.

4. Our app has the part of ranking the result of the search according to your priorities.

2.3 Need for the new system

This program solves a daily social problem in all societies, called the issue of the exploitation of brokers through blackmailing the customers by establishing an application for social communication that cares for this category of society which is not small as we found that the society is in a dire need to this because of:-

- The continuous exploitation of real estate brokers.
- The time, effort and money spent on the traditional ways of searching for an estate.
- The world we live in has become more connected so, it's a must to enroll the technology in the field of real estate
- The unavailability of having a variety of options when it comes to searching for estates
- The limited resources that can be used for finding the suitable choice.

2.4 Analysis of the new system

2.4.1 User Requirements

Mandatory requirements:

1- The user searches for app to rent or buy his estate easily only by few clicks at home without need to broker so we will do that in our app.

2-The user not only needs to rent or buy estates but also selling his estate and sharing it with people to buy it, so we will make two categories the first to buy and the second to sell.

Desirable requirements:

1- The user needs to use GPS to make the process easier so we will provide this feature in our app.

2- The user uses the app with him at any place and any time so our app should be available in any place and any time.

Optional requirement:

1- The users prefer a social media part to communicate with each other so we will provide this feature in our app.

2.4.2 System Requirements

ANDROID

As the application works on the least versions of the android ice cream sandwich because we use a working framework which is responsive to the lowest versions of Android.

IOS

The application works on IOS 8 as we use Cordova package and it is the minimum stable version working on it.

WINDOWS PHONE:-

Our app works on windows phone version 8 and it is the lowest version of windows phone in smart phone versions.

Camera:-

This app needs a mobile camera as it is very useful in taking pictures for listings to make it easy to the user to choose the suitable listing and it doesn't matter which version of camera you will use, but it will be awesome if you use a good one.

GPS:-

You will use the GPS to locate the listing you want to buy and set a location for your listing that you want to sell so it will be better if you use your smart phone GPS

Internet connection:-

This app uses a cloud server to store data so you need to be connected to the internet to fetch and view it.

A part of it considered a social media network so the whole data will be on the server [firebase] also you need it to enable the GPS service and upload photos.

2.4.3 Domain Requirements

1. Multiple users must be able to use the application simultaneously without corrupting the database (whatever form it may be).

2. The necessary software required to run the application. For example

    * if the software is written in ionic or android (java) , Android system must be installed before the application can be used.

    * if the software is written in ionic or swift , IOS system must be installed before the application can be used.

3. A server must be set up to host the database, and the server must be
accessible by all the systems running the inventory tracking software.

4. The database should be backed up every once in a while in case the original does become corrupt.

5. The application must verify all values before making the change in the database.

6. The application must have update capabilities for future models and accessories.

2.4.4 Functional Requirements

1- Sign up:

Actor: client.

Pre: open the application, client should enter a valid data in order to create an account such first name, last name, password, user name, email, and phone.

Description: user can create an account to be a part of the application, and login to the application for the second time he open the application without the need of sign up again.

Post: the user should have an account and enter to home page and his profile that has its data which saved when the account created.

2- Login:

Actor: client, admin.

Pre: open the application, enter login Button, and enter a valid e-mail and password.

Description: user should enter his email and his password for authorizing that he has an account.

Post: enter to home page then he can go to his profile that has its data which saved in the firebase.

3- Update Profile:

Actor: client.

Pre: sign up or login, open his profile.

Description: help users that using the application for updating their information whenever they want.

Post: the updated information will displayed in the profile and saved in the firebase.

4- Delete estate:

Actor: client, admin.

Pre: open the application, login, user must have an account and the estate that he wants to delete is already exists in the firebase.

Description: All estate will display and user can delete any estate he want then he will go to delete form and choose the reason for deleting.

Post: estate will be deleted from firebase.

5- Log out:

Actor: client, admin.

Pre: open the application, login, enter log out Button.

Description: when he make login he will go to home and then go to his profile then click on log out button.

Post: user will log out successfully and could not use the application without doing login again.

6- Search for Estate:

Actor: client, admin.

Pre: sign up or login and write estate's name.

Description: help the user to know more about any estate.

Post: all details about the estate will display .

7-Forget Password:

Actor: client, admin.

Pre: open the application, user should has an account, click on forget password button .

Description: help the user to change password when he forget it, the user should click on forget password and then enter his e-mail to make sure that he is the correct user and sent the change password form to his mail .

Post: this password should saved in the firebase to use it when you make login before.

8- View Feedback:

Actor: client.

Pre: open the application, login
if client : he should go to his profile, choose estate and view reviews .

Description: user go to his profile and can view all feedback that sent to him.

Post: all feedbacks that sent to him will display .

9- View all Estates:

Actor: client, admin.

Pre: open the application, login.

Description: when the user make log in or sign up he will go to home page that contains of all estates.

Post: all estates' details will display.

10- Approve Post:

Actor: Admin.

Pre: Open application, Login, Post must be done by client and that client must send request to the admin to approve that post.

Description: Admin check the post before approving by checking the owner of the estate (National ID card) and the ownership of his estate

Post:
We have two cases:
First , If the post sounds good , the admin will approve it and it will appear in the home page of the application .

Second, if the post improper, the admin will Ignore it.

11- Block User:

Actor: Admin.

Pre: Open application, Login, User must be reported many reports by other users.

Description: Admin check the user who had a lot of reports and check his behaviors, posts and his messages.

Post: We have two cases:
First , If that user already did a bad behaviors , the admin will block him from the system .

Second, If that user did nothing, the admin will not do anything .

12- Add Estate:

Actor: client.

Pre: open the application, login, user must have an account then the client will post his estate including (photo , description and price) .

Description: The client will wait till the admin approve or disapprove his post.

Post:
We have two cases:
First, if the post sounds good, the admin will approve it and it will appear in the home page of the application.

Second, if the post improper, the admin will ignore it.

13- Buy or Rent Estate:

Actor: client.

Pre: open the application, login, user must have an account then the client will enter any post he want, then he read the full description and see other photos.

Description: he will show mobile number of the owner and call him or message the owner on the application.

Post: acceptance or rejection depends on the deal between them.

14- Update Estate:

Actor: client.

Pre: open the application, login, user must have an account then enter his profile and select the estate that he want to update.

Description: After selecting an estate that he want to update, he will enter the new information and save it.

Post: estate will be updated in firebase.

15- Send Message:

Actor: client.

Pre: open the application, login, user must have an account then go to the post that he want to message the owner estate.

Description: the client message the owner in the application.

Post: the message will arrive to the owner.

16- Add to Favorites:

Actor: client.

Pre: open the application, login, user must have an account then the client will enter any post he want.

Description: If he likes the post and want to save it to see it later he will click on star (the symbol of favorites).

Post: The post will be saved to his favorites to see it whenever he want.

16- Send Feedback (Client):

Actor: client.

Pre: open the application, login, owner must have an account then choose any post he want to delete then he go to delete form.

Description: choose the reason for deleting and write feedback if he want (optional) then submit.

Post: The post will be deleted and feedback will be sent to the admin.

17- Send Feedback (Client):

Actor: Client.

Pre: the client will receive notification automatically after the transaction to ask him for his feedback.

Description: when the client open the notification, the form of feedback will be opened and he will fill it.

Post: the feedback will be sent the admin.

2.4.5 Non-functional Requirements

Availability:

Our application can be used in any time and it always “uptime” is the amount of time that it is operational and available for use. This is specified because some systems are designed with expected downtime for activities like database upgrades and backups.

Efficiency:

Our application utilizes scarce resources: CPU cycles, disk space, memory, bandwidth, etc.

Flexibility:

If the team intends to increase or extend the functionality of the software after it is deployed, that should be planned from the beginning; it influences choices made during the design, development, testing, and deployment of the system.

Performance:

Our application is specify the timing characteristics of the software. Certain tasks or features are more time sensitive than others; the nonfunctional requirements should identify those software functions that have constraints on their performance.

Reliability:

Our application is capable of the software to maintain its performance over time. Unreliable software fails frequently, and certain tasks are more sensitive to failure.

Robustness:

Our application is able to handle error conditions gracefully, without failure. This includes a tolerance of invalid data, software defects, and unexpected operating conditions.

Scalability:

Our application is scalable has the ability to handle a wide variety of system configuration sizes. The nonfunctional requirements should specify the ways in which the system may be expected to scale up (by increasing hardware capacity, adding machines, etc.).

Usability:

Our application is Ease to use requirements address the factors that constitute the capacity of the software to be understood, learned, and used by its intended users.

2.5 Advantages of the new system

1. Settle mobile app enables users to rent or buy estates from their home within only few clicks.

2. Settle mobile app uses GPS to rent or buy a flat and this point makes renting easier because it does not force users to know the places or street names which they need to rent or buy in.

3. Settle mobile app doesn’t need to be used in a specific place.

4. Settle mobile app contains the part of social media to make users communicate with each other and exchange experience.

5. Settle mobile app has a feature of ranking the result of search according to your priorities.

6. Settle mobile app works for both flats and shops and also for renting or selling.

2.6 Risk and Risk Management

Risk management process:

1- Risk identification:

Risk type	Possible risks
Techno logy	The database used in the system cannot process as many transactions per second as expected. Software components that should be reused contain defects that limit their functionality.
People	It is impossible le to recruit staff with the skills required. Key staff are ill and unavailable at critical times. Required training for staff is not available.
Organizational	The organization is restructured so that different management are responsible for the project.
Tool	The cod e generated by CASE tools is inefficient. CASE tools cannot be integrated.
Requirements	Changes to requirements that require major design rework are proposed. Customers fail to understand the impact of requirements changes
Estimation	The time required to develop the software is underestimated. The rate of defect repair is underestimated. The size of the software is underestimated

2- Risk analysis :

Risk	Probability	Effects
Organizational financial problems force reductions in the project budge t.	low	Catastrophic
It is impossible to recruit staff with the skills required for the project	high	Catastrophic
Key staff are ill at critical times in the project	moderate	Serious
Software components that should be reused contain defects which limit their functionality.	moderate	Serious
Changes to requirements that require major design rework are proposed	moderate	Serious
The organization is restructured so that different manage met are responsible for the project	high	Serious
The database used in the system cannot process as many transactions per second as expected	moderate	Serious
The time required to develop the software is underestimated	high	Serious
CASE tools cannot be integrated	High	Tolerable
Customers fail to understand the impact of requirements change	moderate	Tolerable
The rate of defect repair is underestimated	moderate	Tolerable
The size of the software is underestimated	high	Tolerable

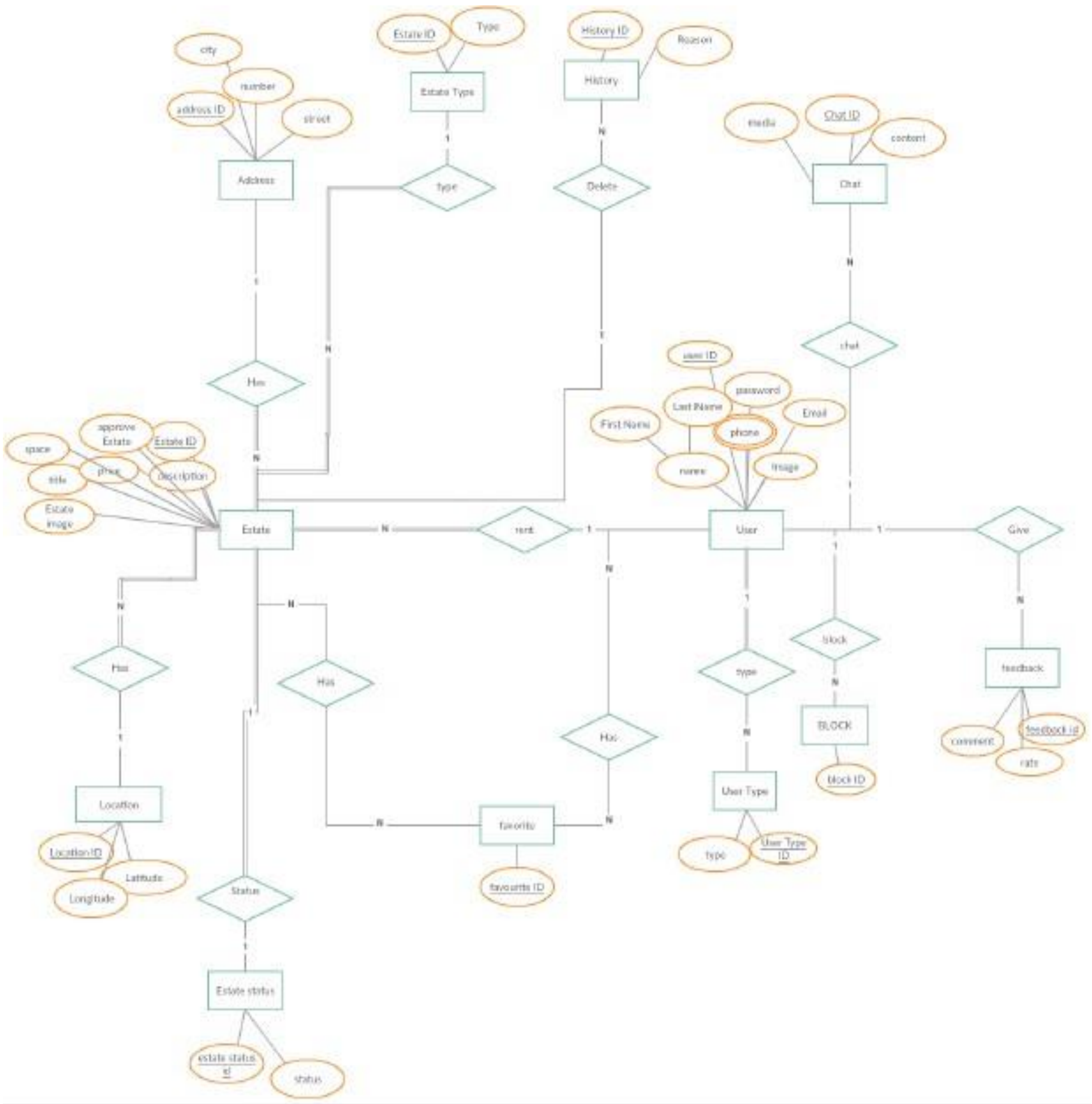
3- Risk planning :

Risk	Strategy
Organizational financial problems	Prepare a briefing document for senior manage met showing how the project is making a very important contribution to the goals of the business.
Recruitment problems	Alert customer of potential difficulties and the possibility of delays, investigate buying- in components
Staff illness	Reorganize team so that there is more overlap of work and people therefore understand each other's jobs.
Defective components	Replace potentially defective components with bough tin components of known reliability
Requirements changes	Derive traceability information to assess requirements change impact, maximize information hiding in the design
Organizational restructuring	Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business.
Database performance	Investigate the possibility of buying a higher performance database

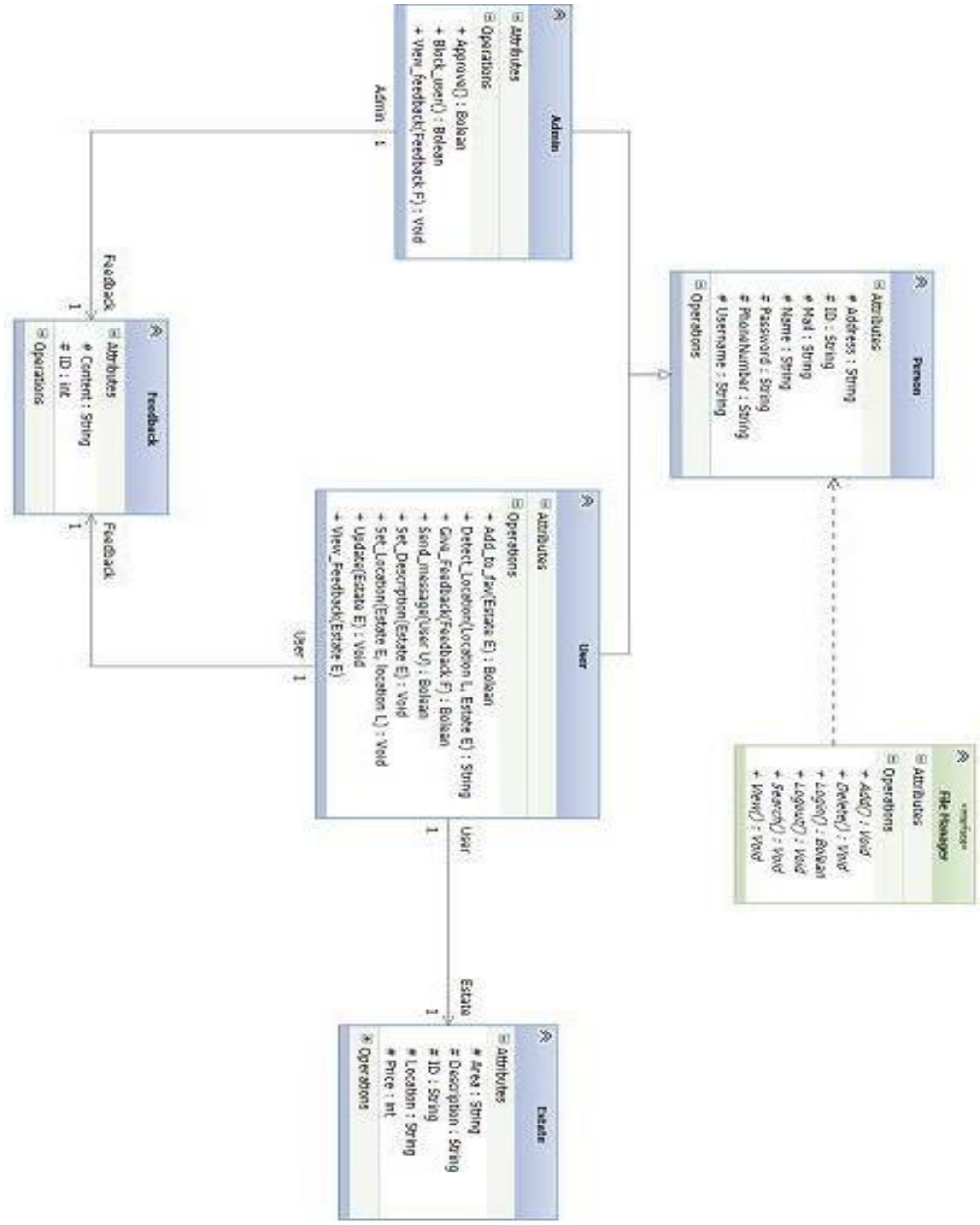
Chapter 3: Software Design

In this chapter we're going to discuss and go deeper in Settle mobile application's design, and present its diagrams and database.

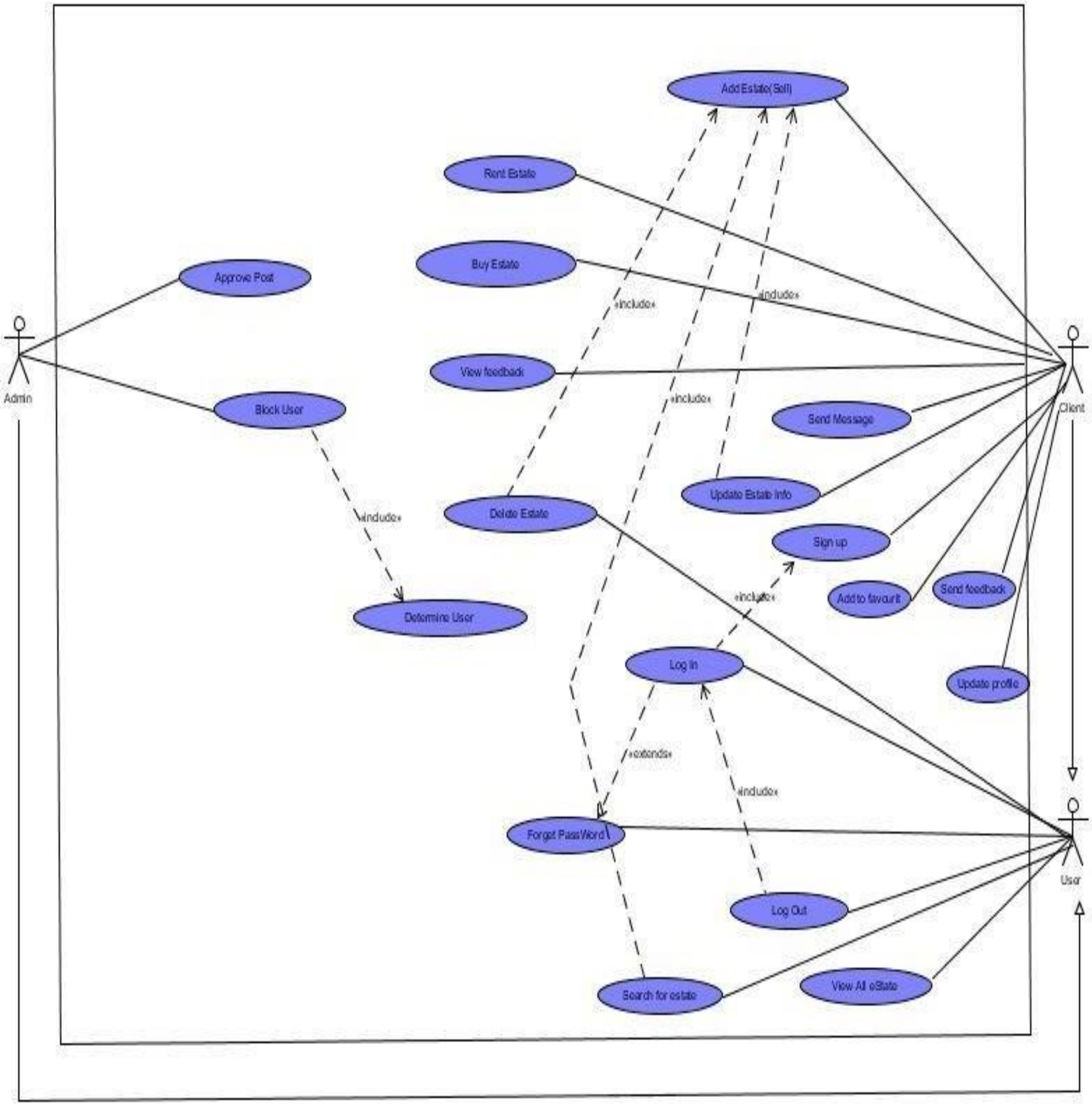
3.1 Design of database



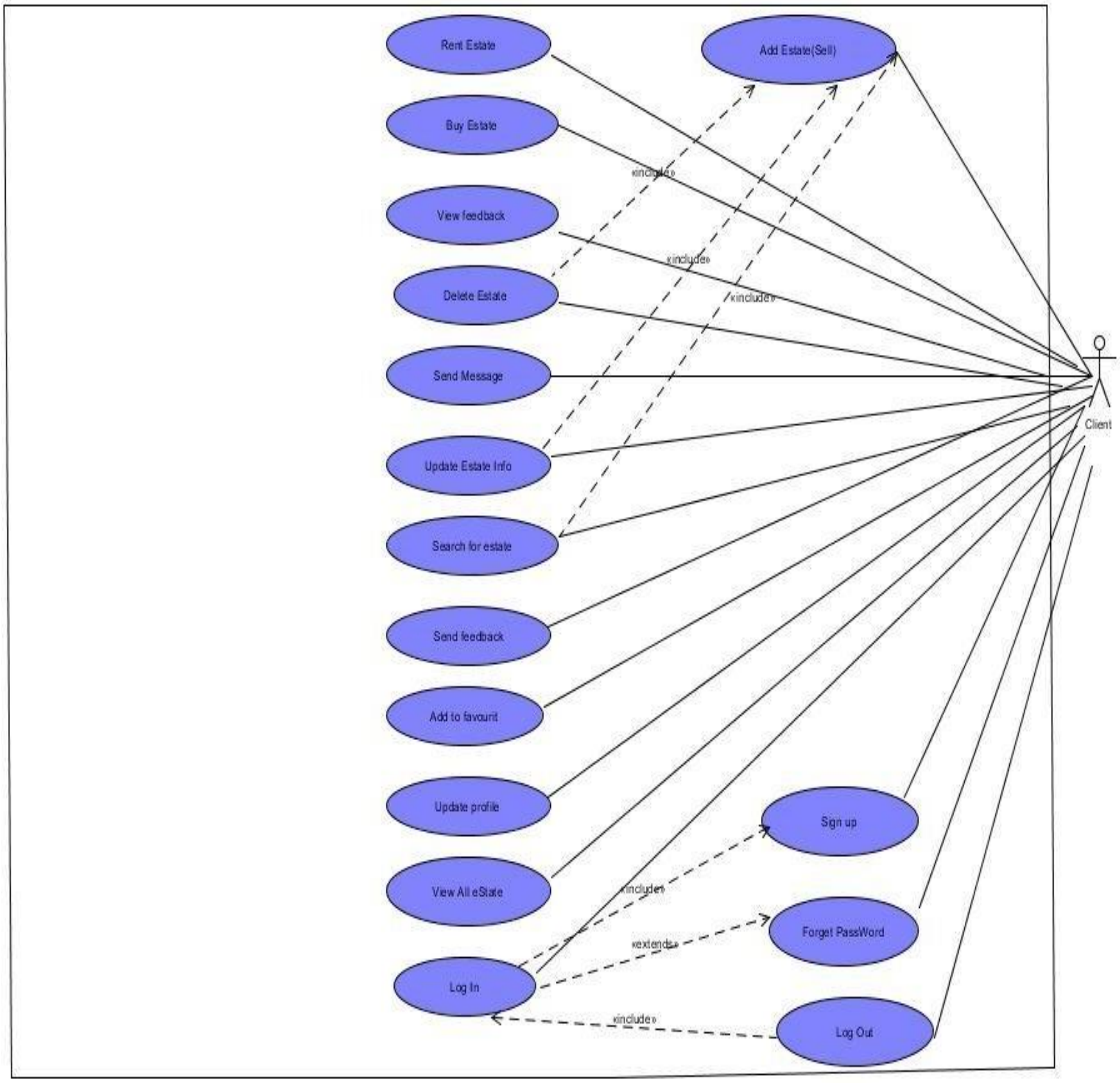
3.2 Class Diagram



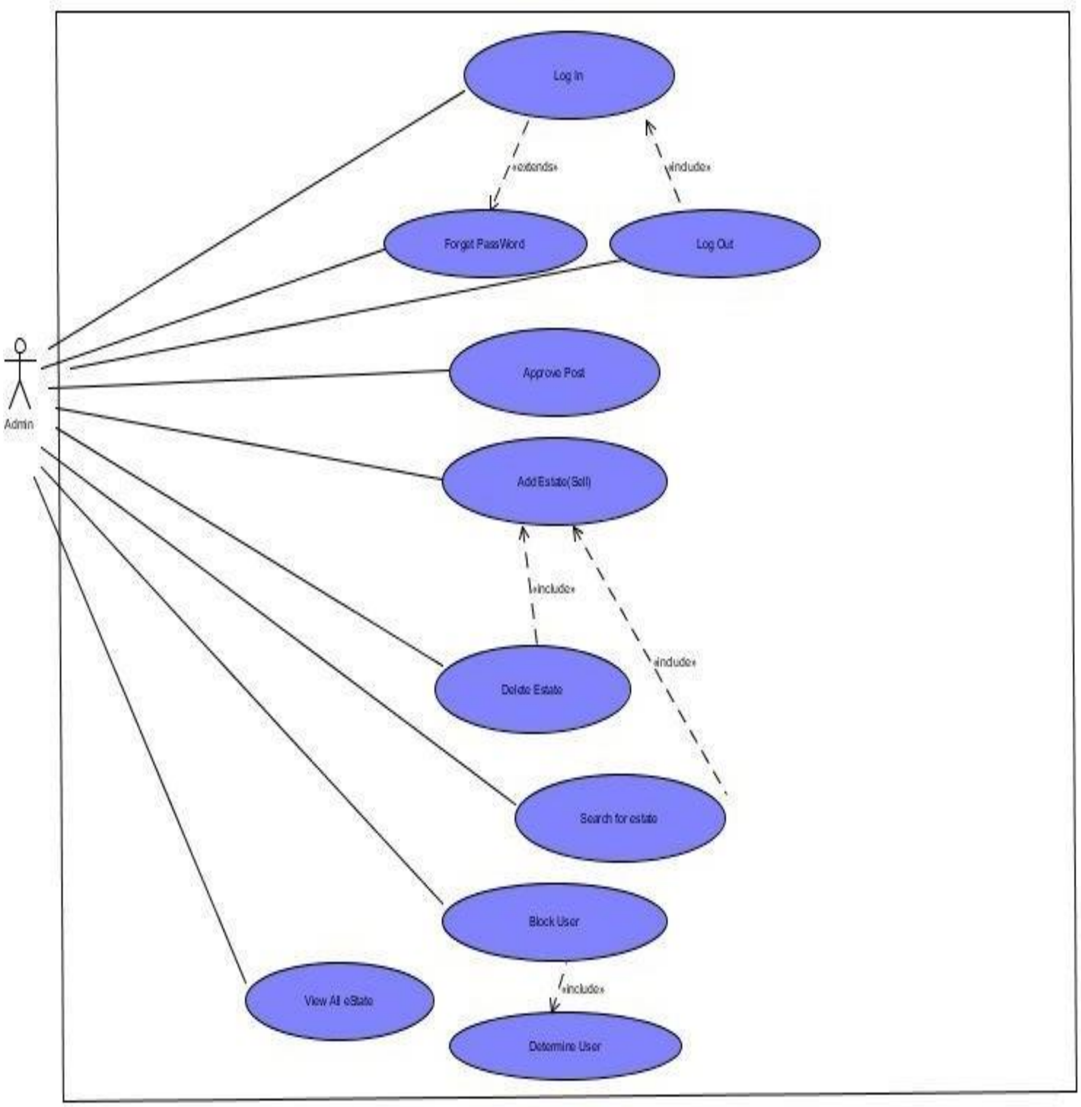
3.3 Use-case Diagram



3.3 Use-case Diagram (Cont.)

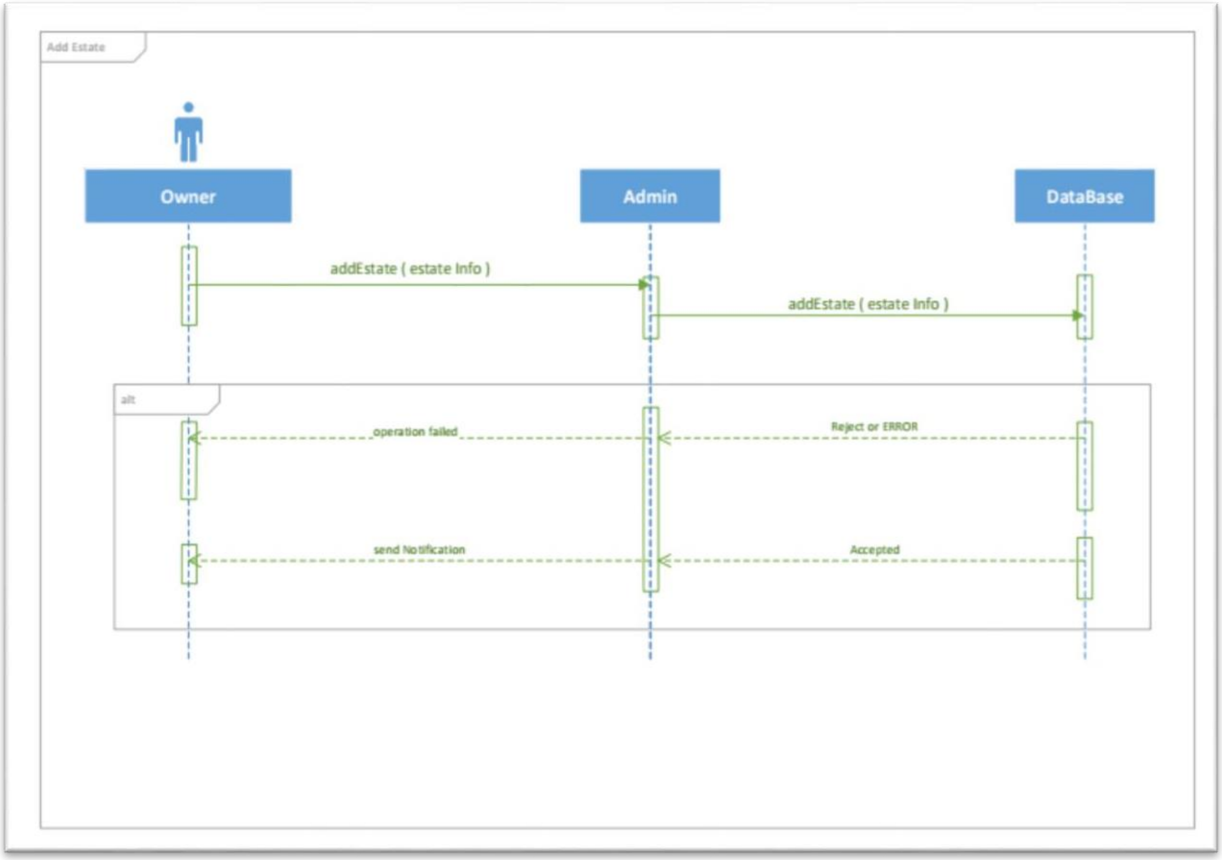


3.3 Use-case Diagram (Cont.)

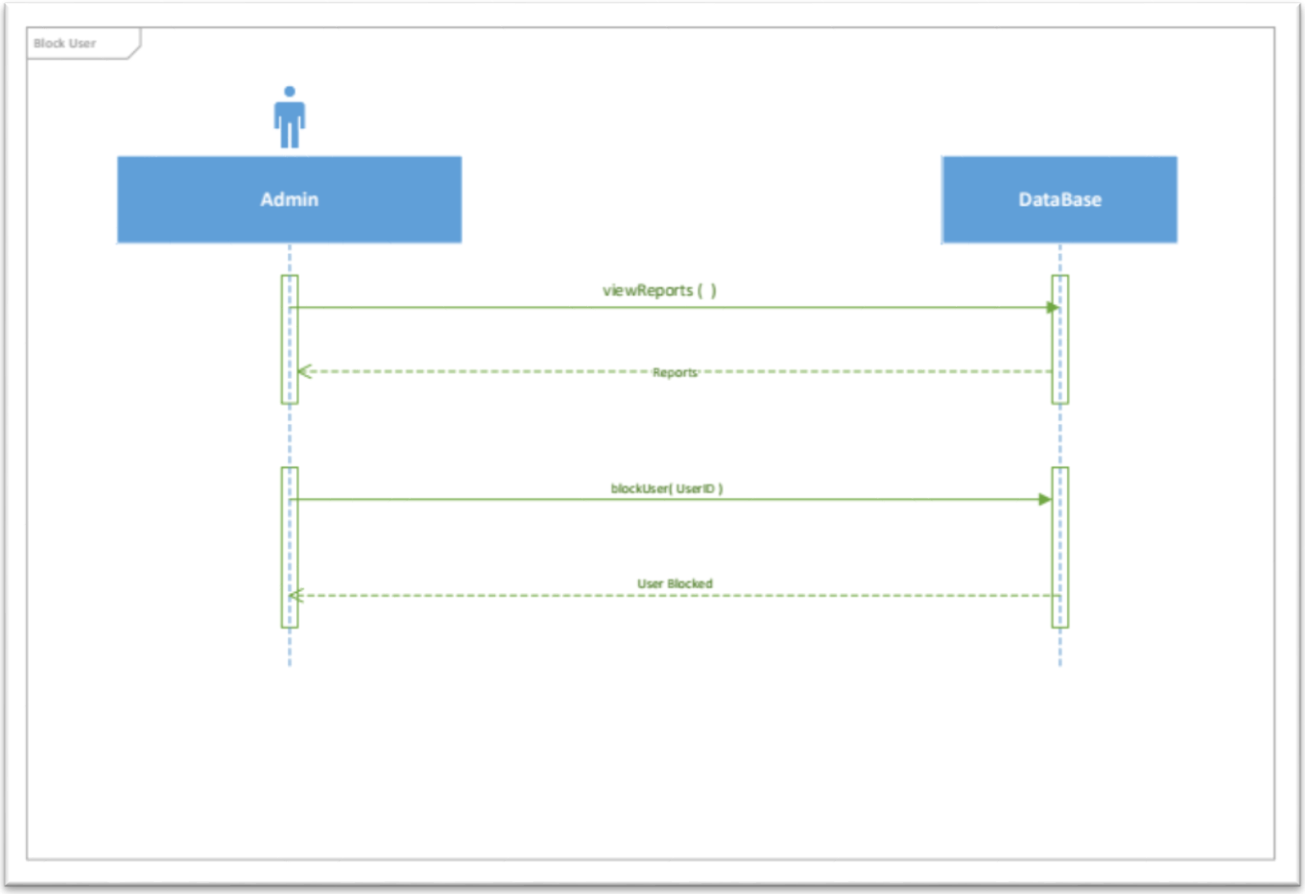


3.4 Sequence Diagram

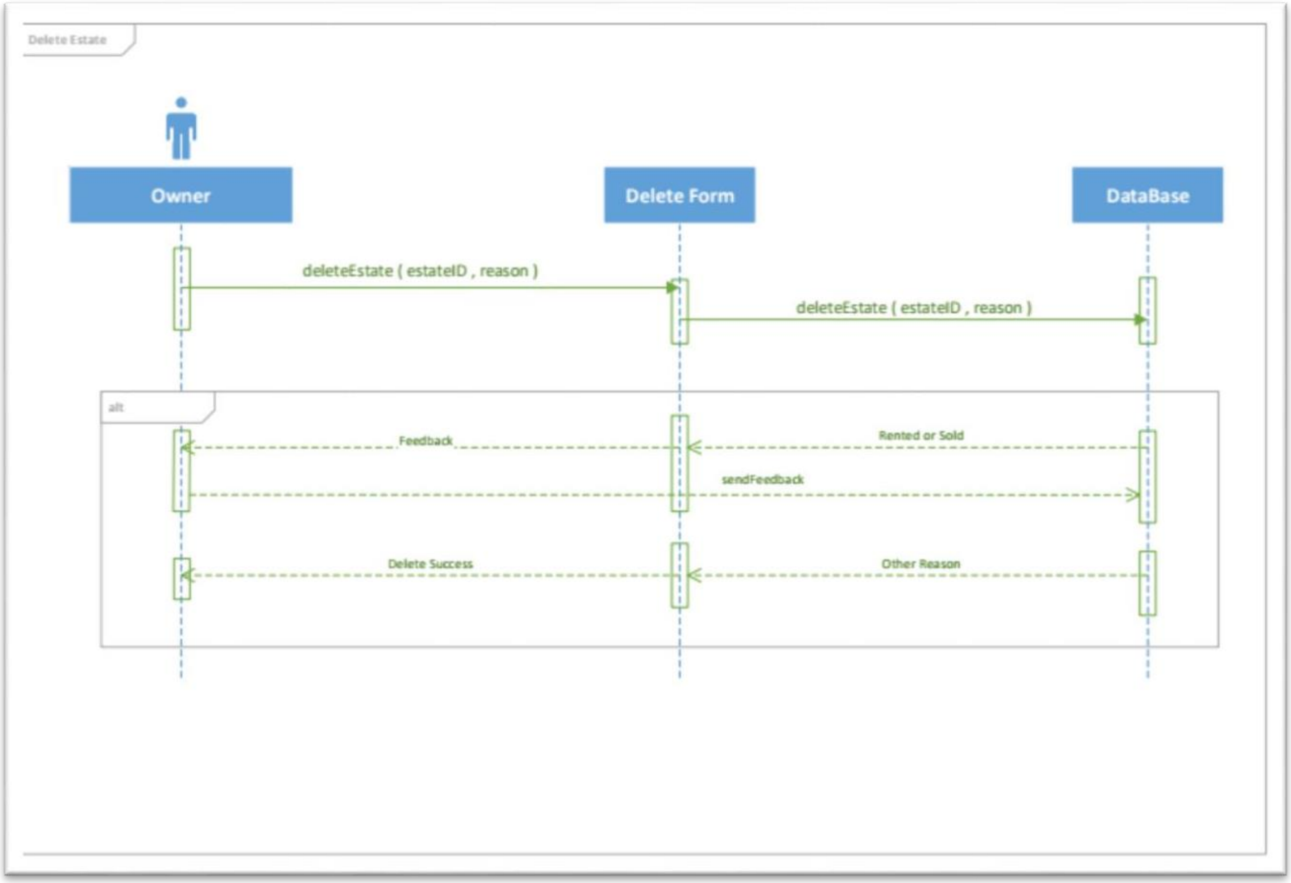
1- Add Estate:-



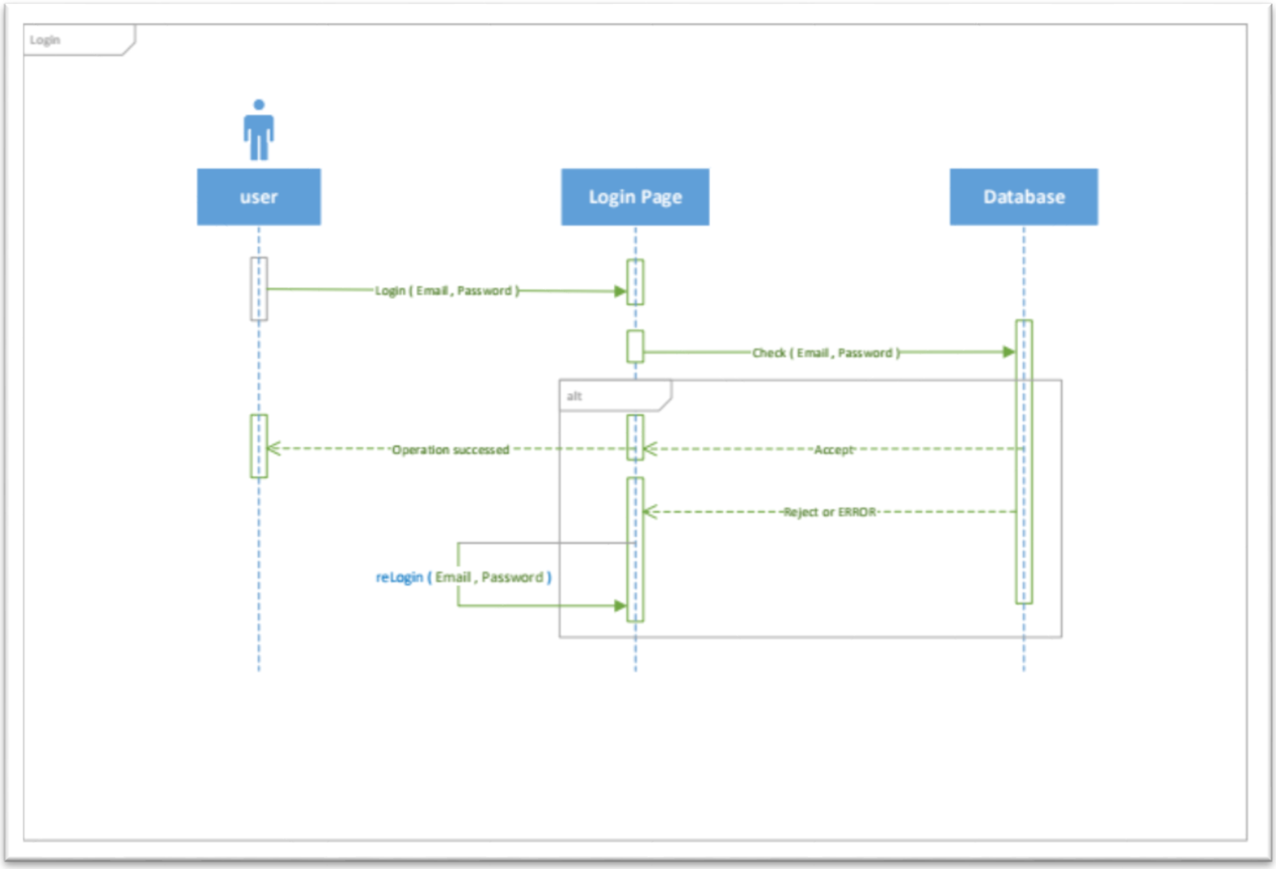
2-Block User:-


Block User
Admin
DataBase
viewReports ( )
Reports
blockUser(UserID)
User Blocked

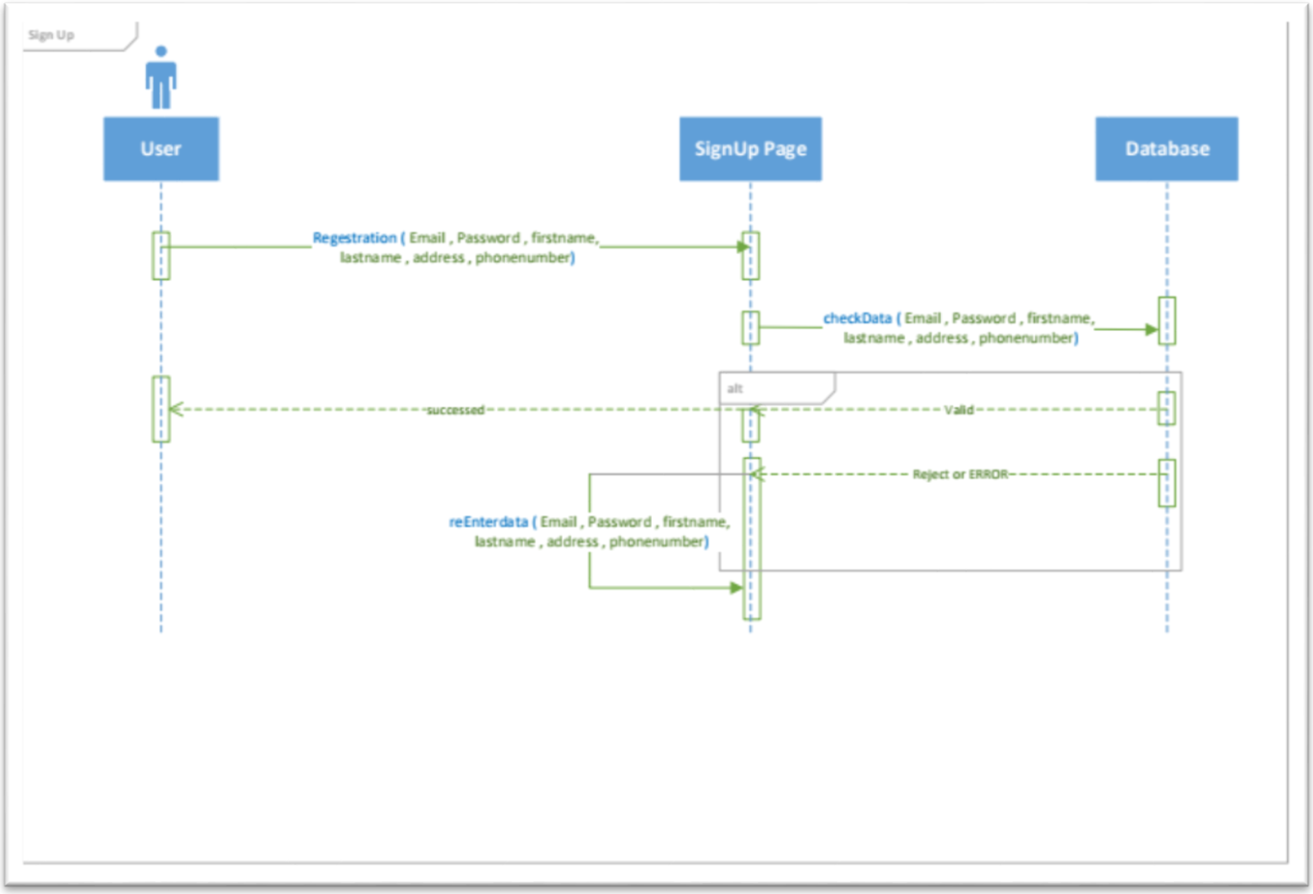
3-Delete Estate:-



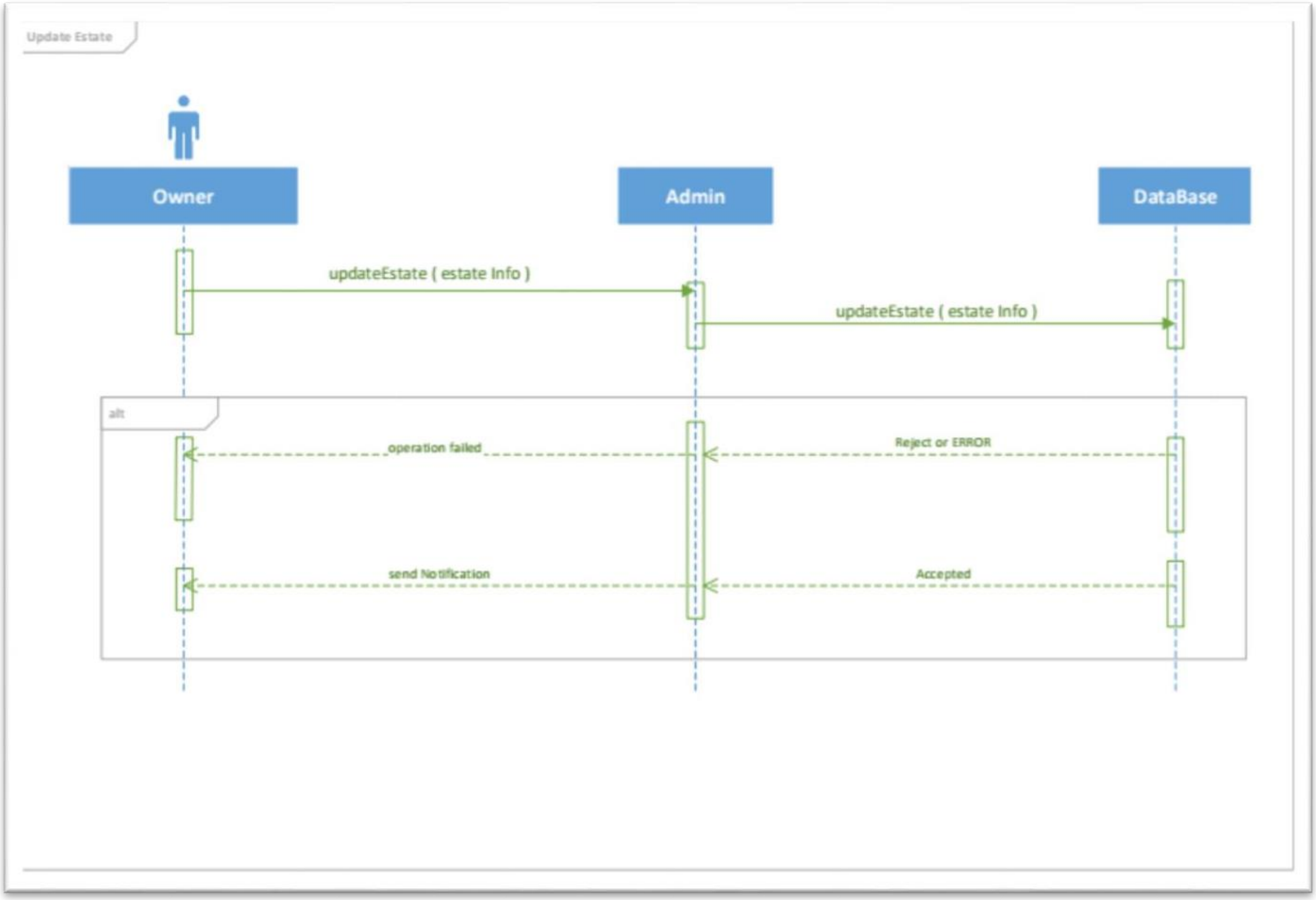
4- Login:-


Login
user
Login Page
Database
Login (Email , Password)
Check (Email , Password)
alt
Operation successed
Accept
Reject or ERROR
reLogin (Email , Password)

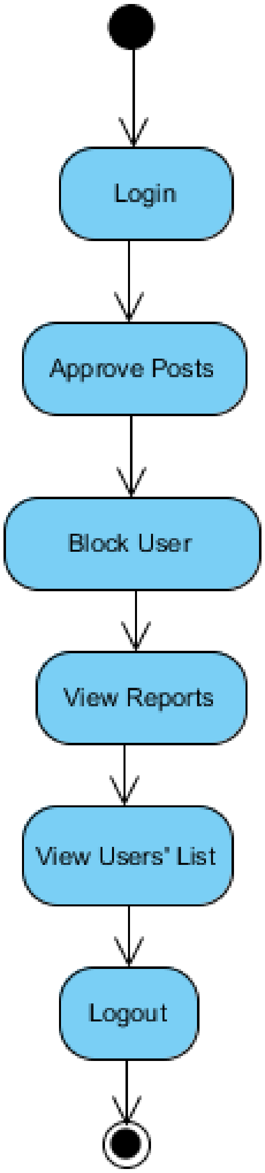
5- Sign-up


Sign Up
User
SignUp Page
Database
Regestration (Email , Password , firstname, lastname , address , phonenumber)
checkData (Email , Password , firstname, lastname , address , phonenumber)
alt
successed
Valid
Reject or ERROR
reEnterdata (Email , Password , firstname, lastname , address , phonenumber)

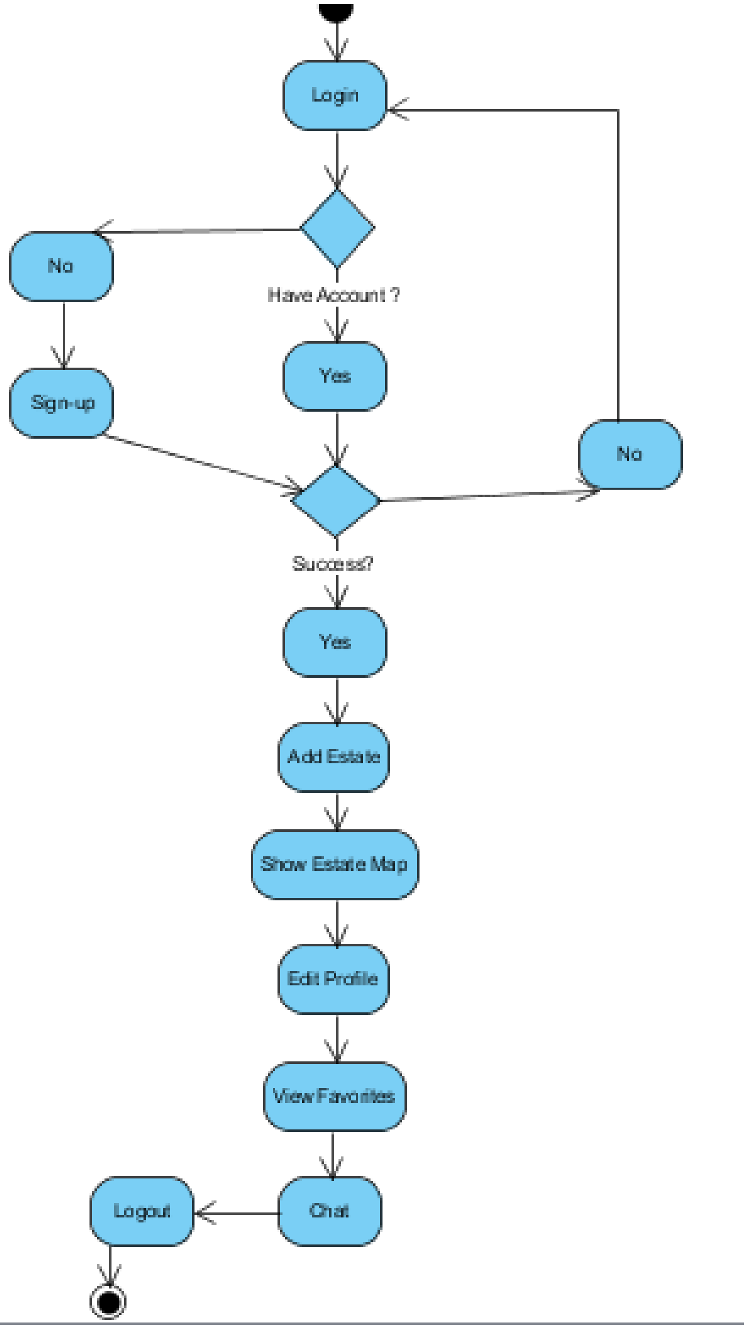
6-Update Estate:-



3.5 Activity Diagram



3.5 Activity Diagram (Cont.)

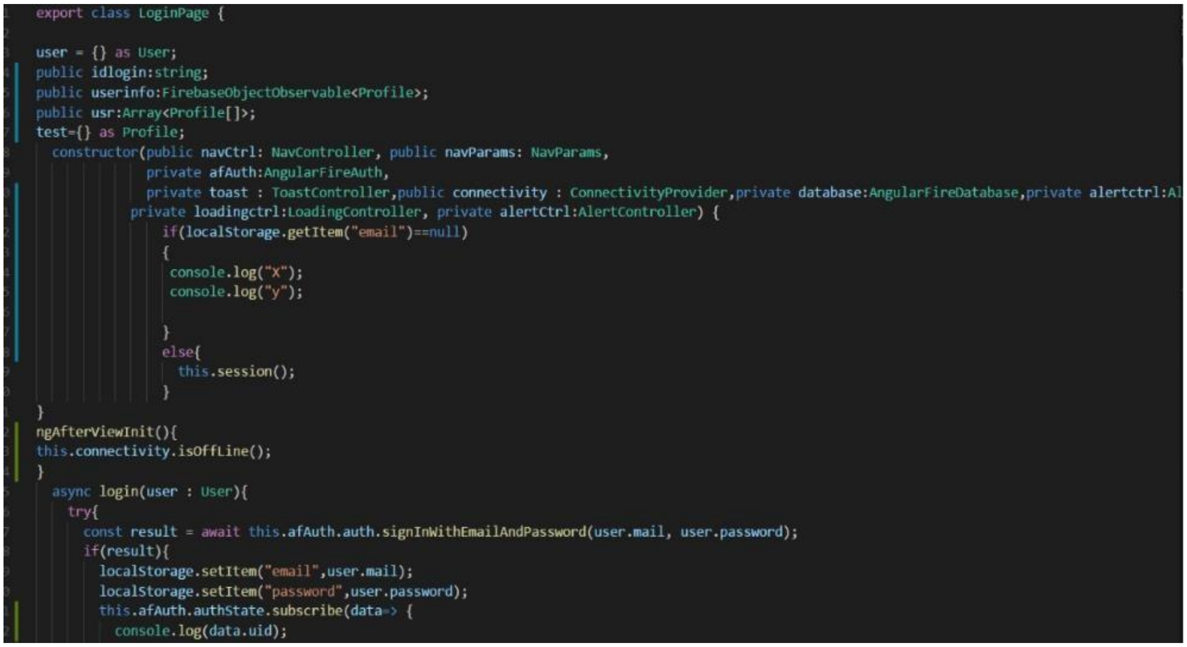


Chapter 4
Implementation

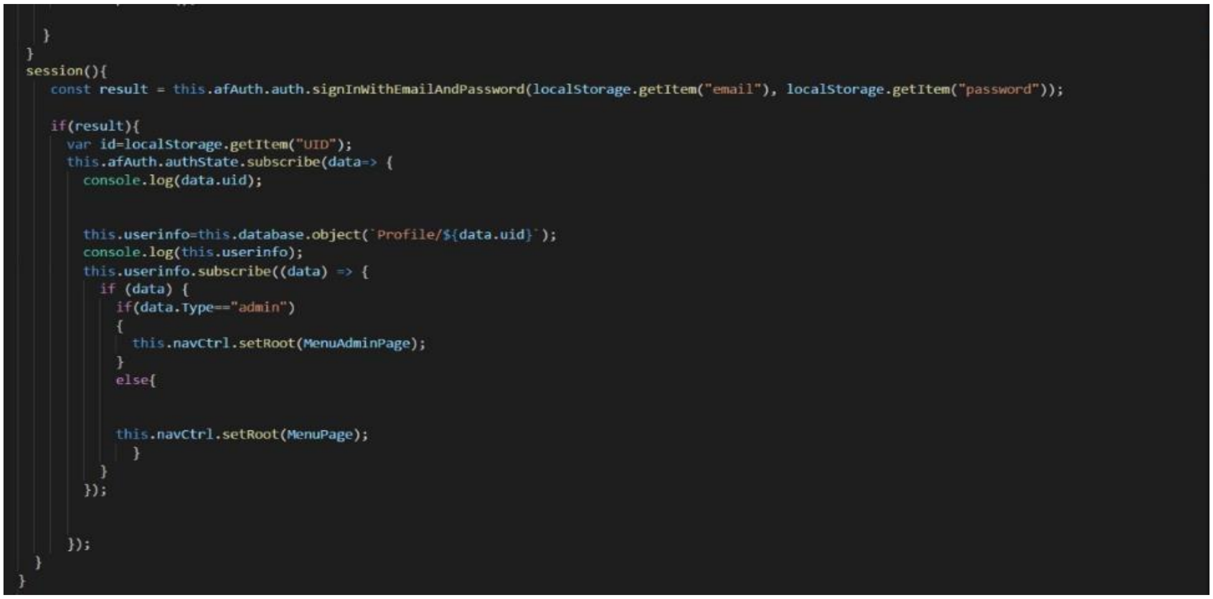
In this chapter we're going to discuss and go deeper in Settle mobile application's implementation, and present its code and the algorithms used to build it.

4.1 Software Architecture

1. Login:-

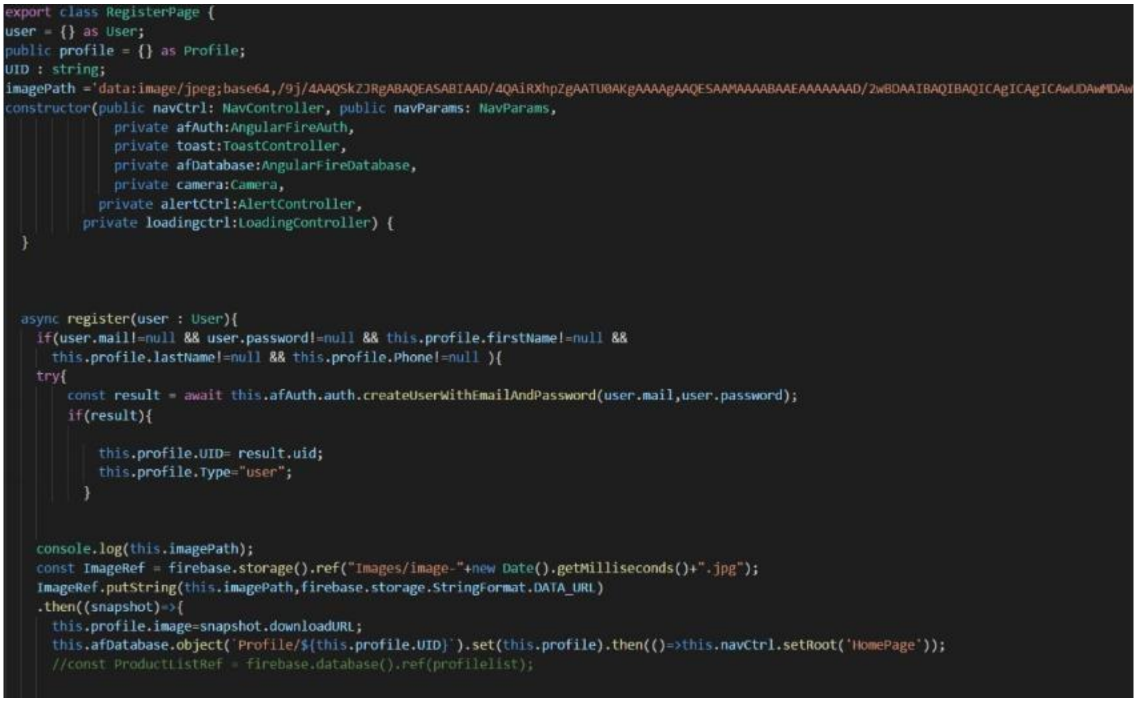


```
export class LoginPage {

user = {} as User;
public idlogin:string;
public userinfo:FirebaseObjectObservable<Profile>;
public usr:Array<Profile[]>;
test={} as Profile;
  constructor(public navCtrl: NavController, public navParams: NavParams,
              private afAuth:AngularFireAuth,
              private toast : ToastController,public connectivity : ConnectivityProvider,private database:AngularFireDatabase,private alertctrl:Al
            private loadingctrl:LoadingController, private alertCtrl:AlertController) {
              if(localStorage.getItem("email")==null)
              {
               console.log("X");
               console.log("y");

              }
              else{
                this.session();
              }
}
ngAfterViewInit(){
this.connectivity.isOffLine();
}
  async login(user : User){
    try{
      const result = await this.afAuth.auth.signInWithEmailAndPassword(user.mail, user.password);
      if(result){
        localStorage.setItem("email",user.mail);
        localStorage.setItem("password",user.password);
        this.afAuth.authState.subscribe(data=> {
          console.log(data.uid);
```

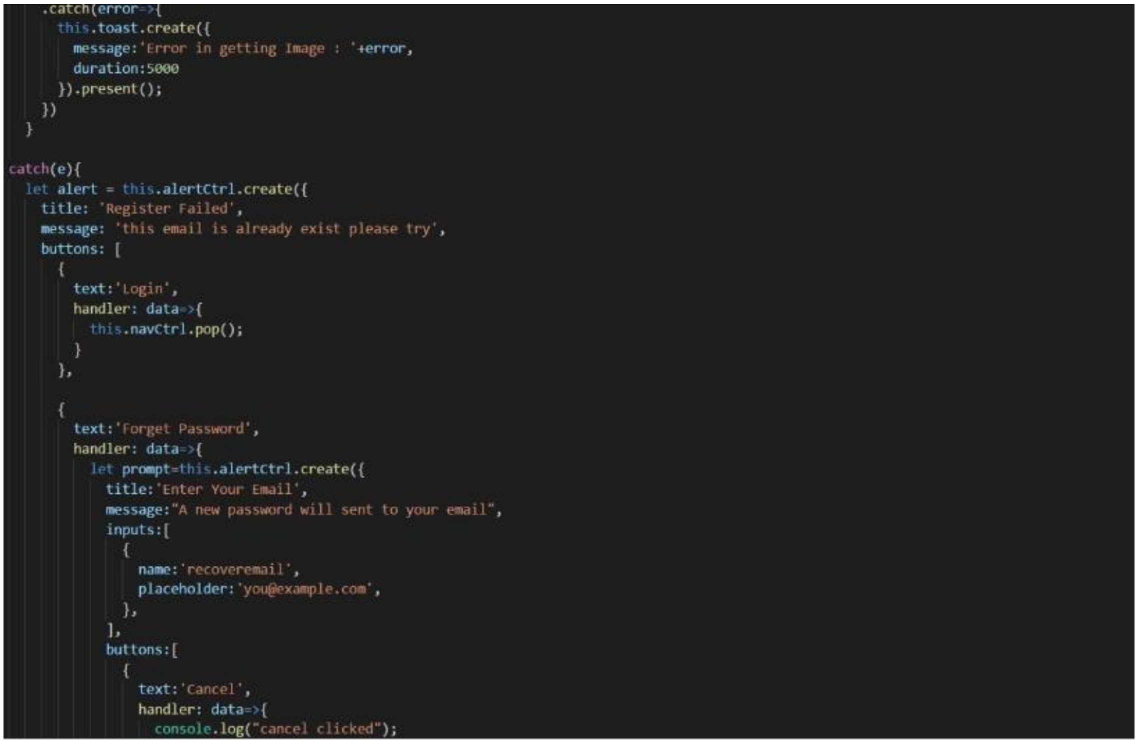


```
  }
}
session(){
   const result = this.afAuth.auth.signInWithEmailAndPassword(localStorage.getItem("email"), localStorage.getItem("password"));

   if(result){
     var id=localStorage.getItem("UID");
     this.afAuth.authState.subscribe(data=> {
       console.log(data.uid);


       this.userinfo=this.database.object(`Profile/${data.uid}`);
       console.log(this.userinfo);
       this.userinfo.subscribe((data) => {
         if (data) {
           if(data.Type=="admin")
           {
             this.navCtrl.setRoot(MenuAdminPage);
           }
           else{


           this.navCtrl.setRoot(MenuPage);
             }
         }
       });


     });
  }
}
```

2- Registration:-

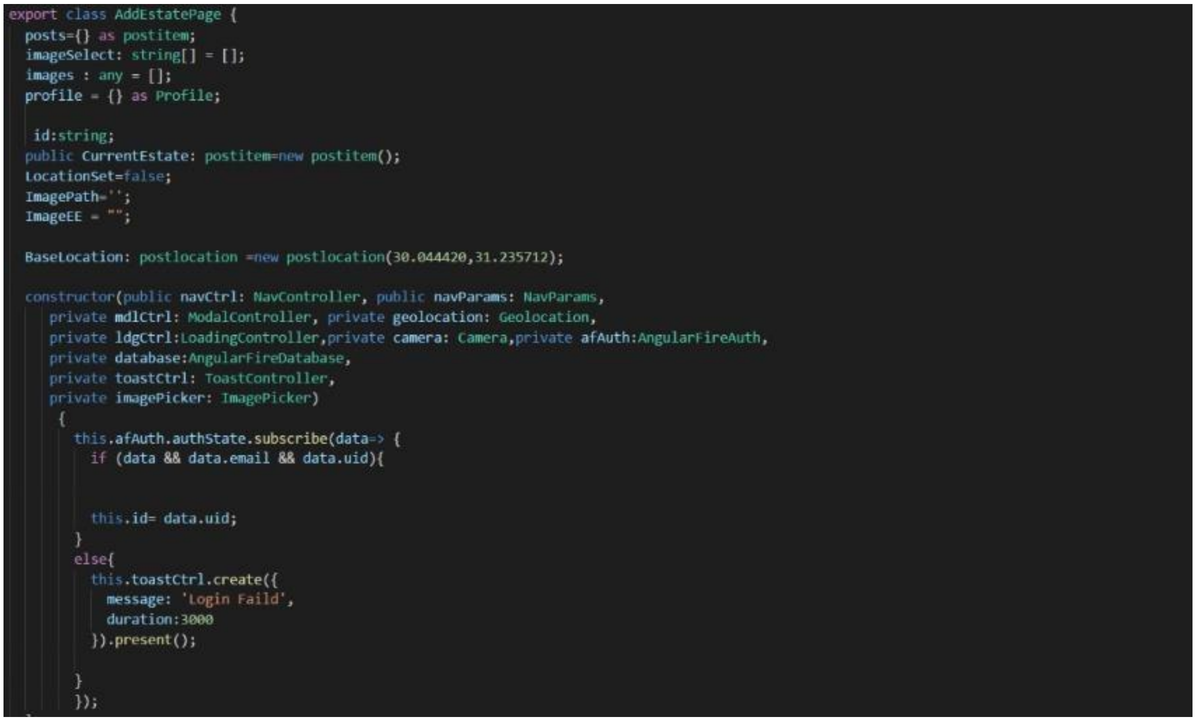


```
export class RegisterPage {
user = {} as User;
public profile = {} as Profile;
UID : string;
imagePath ='data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEASABIAAD/4QAiRXhpZgAATU0AKgAAAAgAAQESAAMAAAABAAEAAAAAAAD/2wBDAAIBAQIBAQICAgICAgICAwUDAwMDAw
constructor(public navCtrl: NavController, public navParams: NavParams,
              private afAuth:AngularFireAuth,
              private toast:ToastController,
              private afDatabase:AngularFireDatabase,
              private camera:Camera,
            private alertCtrl:AlertController,
          private loadingctrl:LoadingController) {
 }



  async register(user : User){
    if(user.mail!=null && user.password!=null && this.profile.firstName!=null &&
      this.profile.lastName!=null && this.profile.Phone!=null ){
    try{
        const result = await this.afAuth.auth.createUserWithEmailAndPassword(user.mail,user.password);
        if(result){

            this.profile.UID= result.uid;
            this.profile.Type="user";
          }


    console.log(this.imagePath);
    const ImageRef = firebase.storage().ref("Images/image-"+new Date().getMilliseconds()+".jpg");
    ImageRef.putString(this.imagePath,firebase.storage.StringFormat.DATA_URL)
    .then((snapshot)=>{
      this.profile.image=snapshot.downloadURL;
      this.afDatabase.object(`Profile/${this.profile.UID}`).set(this.profile).then(()=>this.navCtrl.setRoot('HomePage'));
      //const ProductListRef = firebase.database().ref(profilelist);
```



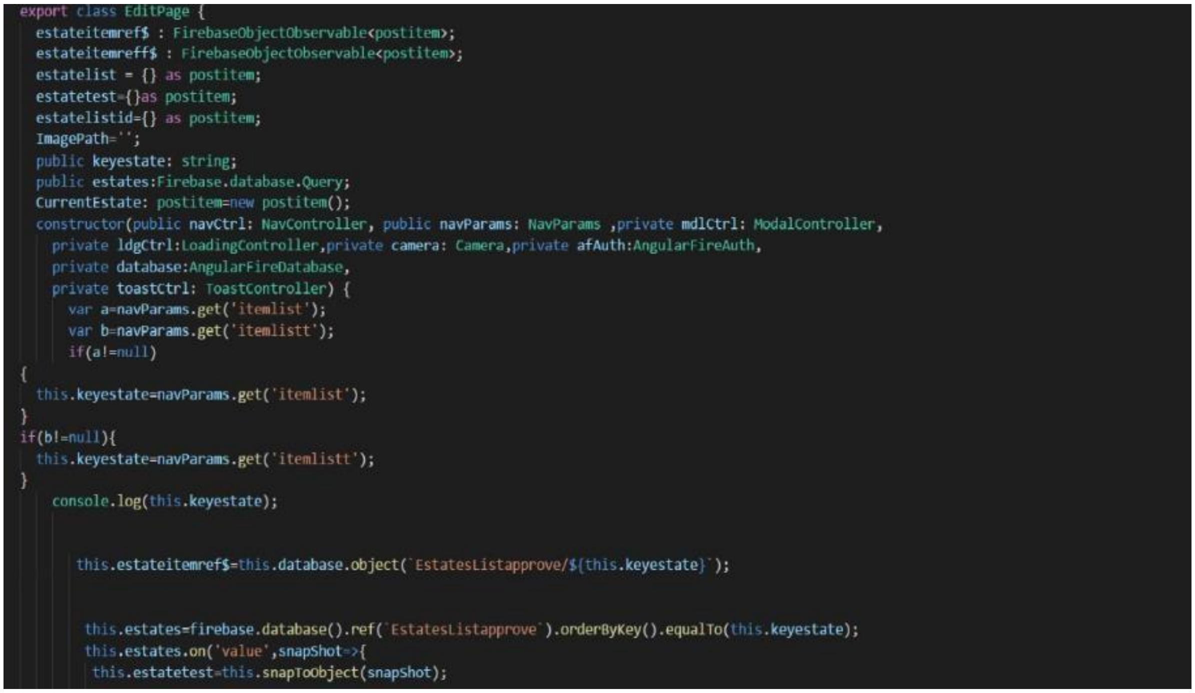
```
    .catch(error=>{
      this.toast.create({
        message:'Error in getting Image : '+error,
        duration:5000
      }).present();
    })
  }

catch(e){
  let alert = this.alertCtrl.create({
    title: 'Register Failed',
    message: 'this email is already exist please try',
    buttons: [
      {
        text:'Login',
        handler: data=>{
          this.navCtrl.pop();
        }
      },

      {
        text:'Forget Password',
        handler: data=>{
          let prompt=this.alertCtrl.create({
            title:'Enter Your Email',
            message:"A new password will sent to your email",
            inputs:[
              {
                name:'recoveremail',
                placeholder:'you@example.com',
              },
            ],
            buttons:[
              {
                text:'Cancel',
                handler: data=>{
                  console.log("cancel clicked");
```

3- Adding an estate:-



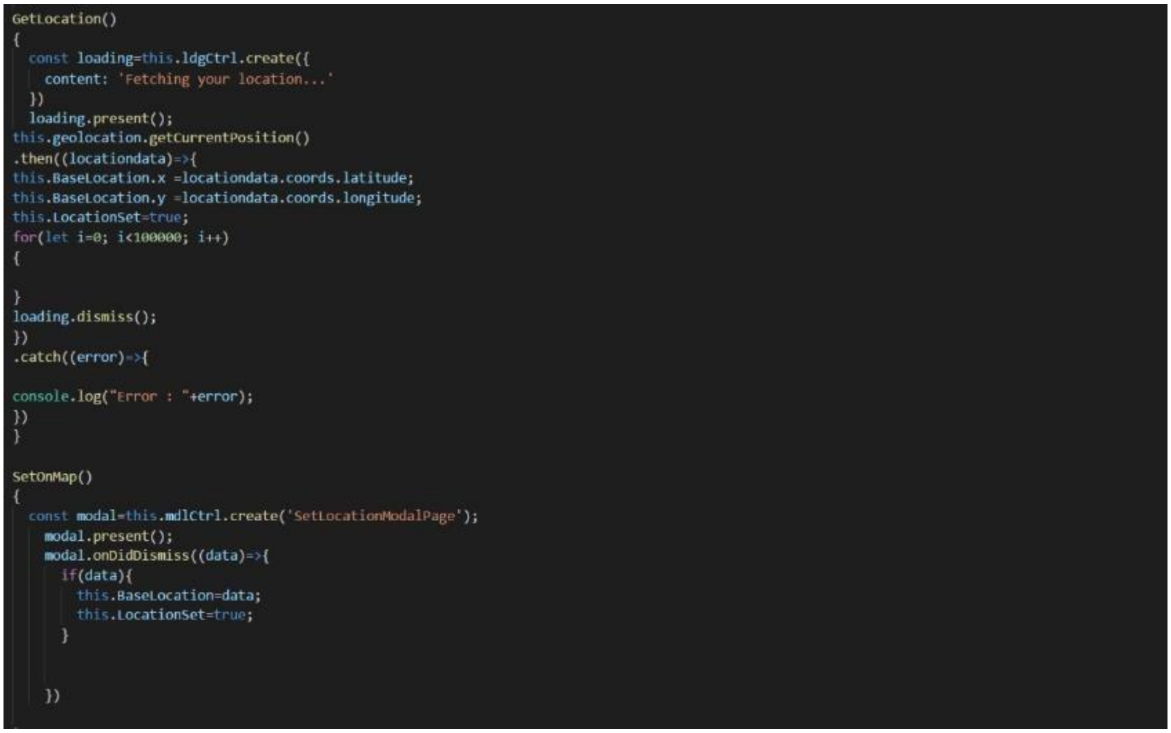
```
export class AddEstatePage {
  posts={} as postitem;
  imageSelect: string[] = [];
  images : any = [];
  profile = {} as Profile;

   id:string;
  public CurrentEstate: postitem=new postitem();
  LocationSet=false;
  ImagePath='';
  ImageEE = "";

  BaseLocation: postlocation =new postlocation(30.044420,31.235712);

  constructor(public navCtrl: NavController, public navParams: NavParams,
     private mdlCtrl: ModalController, private geolocation: Geolocation,
     private ldgCtrl:LoadingController,private camera: Camera,private afAuth:AngularFireAuth,
     private database:AngularFireDatabase,
     private toastCtrl: ToastController,
     private imagePicker: ImagePicker)
      {
        this.afAuth.authState.subscribe(data=> {
          if (data && data.email && data.uid){


          this.id= data.uid;
        }
        else{
          this.toastCtrl.create({
            message: 'Login Faild',
            duration:3000
          }).present();

        }
        });
```

4-Editing an estate:-



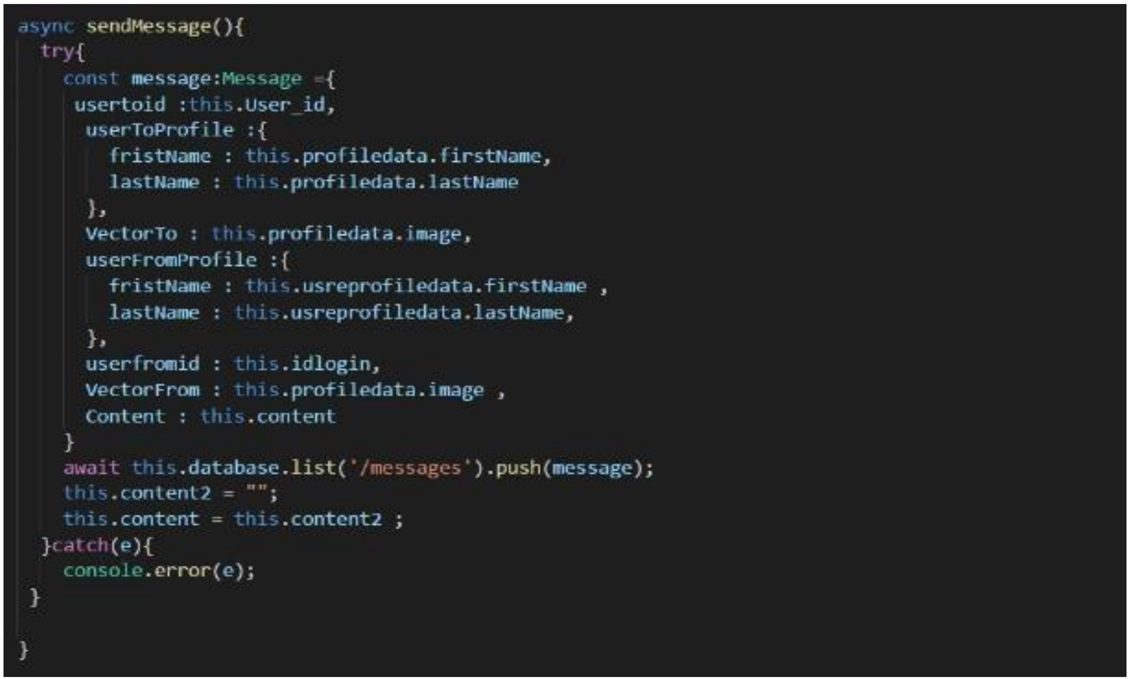
```
export class EditPage {
  estateitemref$ : FirebaseObjectObservable<postitem>;
  estateitemreff$ : FirebaseObjectObservable<postitem>;
  estatelist = {} as postitem;
  estatetest={}as postitem;
  estatelistid={} as postitem;
  ImagePath='';
  public keyestate: string;
  public estates:Firebase.database.Query;
  CurrentEstate: postitem=new postitem();
  constructor(public navCtrl: NavController, public navParams: NavParams ,private mdlCtrl: ModalController,
    private ldgCtrl:LoadingController,private camera: Camera,private afAuth:AngularFireAuth,
    private database:AngularFireDatabase,
    private toastCtrl: ToastController) {
      var a=navParams.get('itemlist');
      var b=navParams.get('itemlistt');
      if(a!=null)
{
  this.keyestate=navParams.get('itemlist');
}
if(b!=null){
  this.keyestate=navParams.get('itemlistt');
}
    console.log(this.keyestate);


       this.estateitemref$=this.database.object(`EstatesListapprove/${this.keyestate}`);


        this.estates=firebase.database().ref(`EstatesListapprove`).orderByKey().equalTo(this.keyestate);
        this.estates.on('value',snapShot=>{
         this.estatetest=this.snapToObject(snapShot);
```

5- Getting location:-



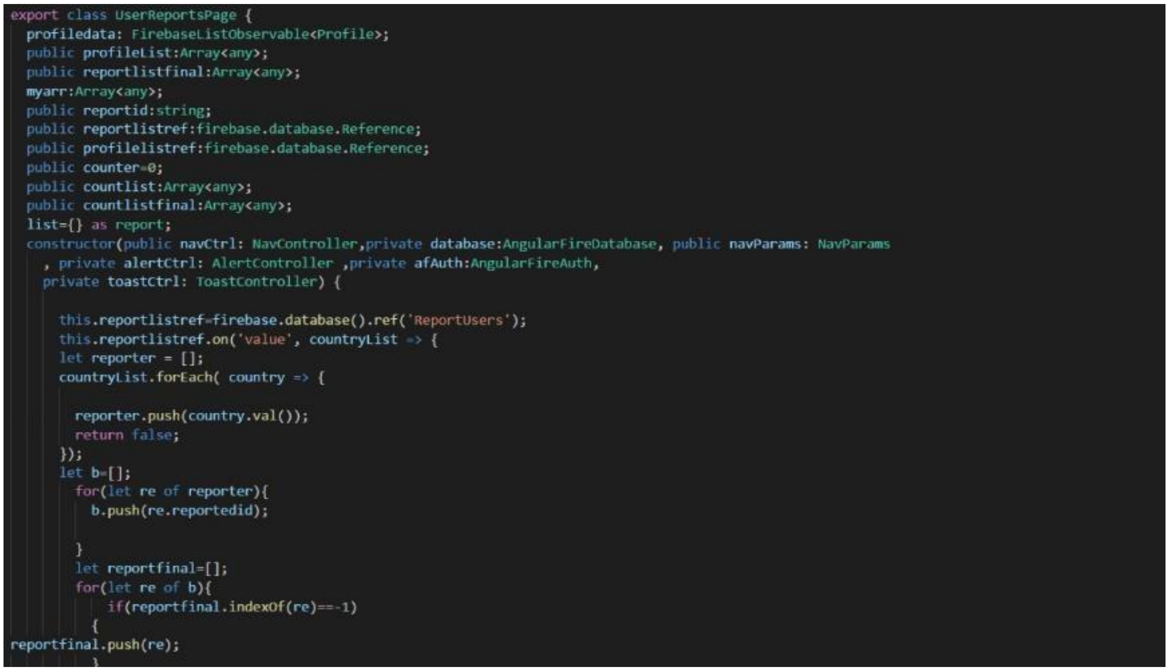
```
GetLocation()
{
  const loading=this.ldgCtrl.create({
    content: 'Fetching your location...'
  })
  loading.present();
this.geolocation.getCurrentPosition()
.then((locationdata)=>{
this.BaseLocation.x =locationdata.coords.latitude;
this.BaseLocation.y =locationdata.coords.longitude;
this.LocationSet=true;
for(let i=0; i<100000; i++)
{

}
loading.dismiss();
})
.catch((error)=>{

console.log("Error : "+error);
})
}

SetOnMap()
{
  const modal=this.mdlCtrl.create('SetLocationModalPage');
    modal.present();
    modal.onDidDismiss((data)=>{
      if(data){
        this.BaseLocation=data;
        this.LocationSet=true;
      }


    })
```

6- Sending Message:-

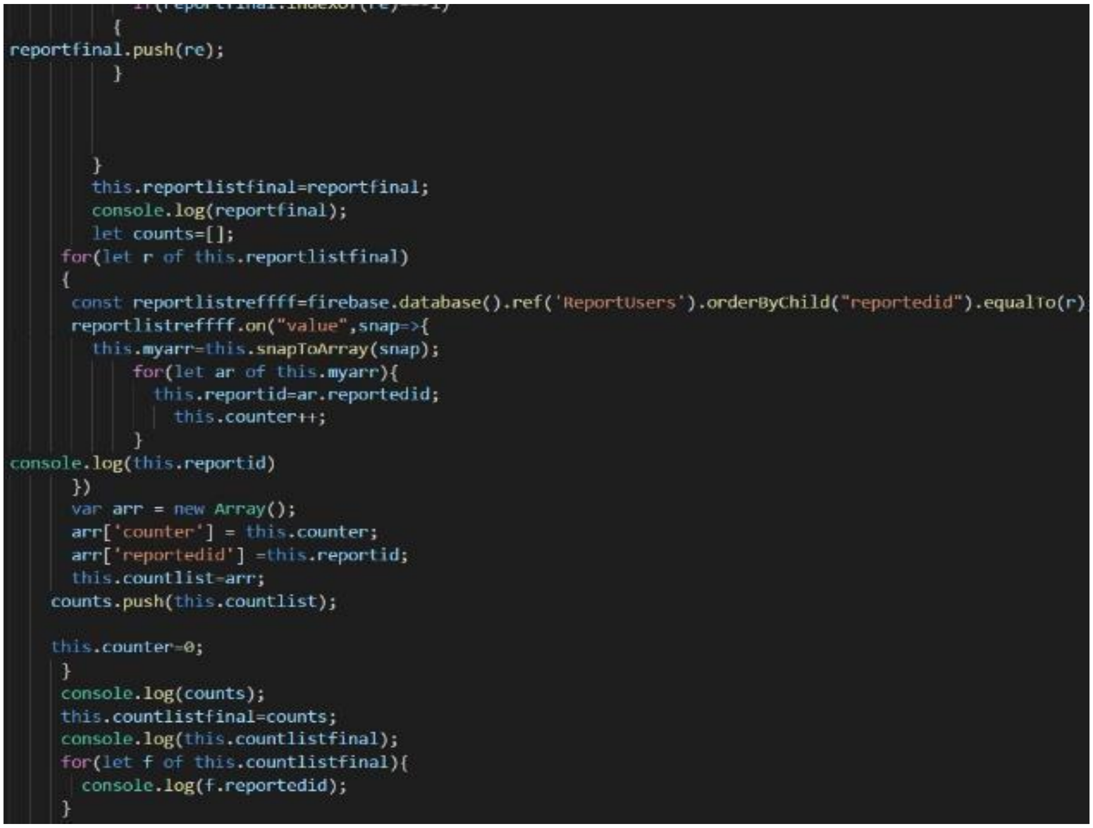


```
async sendMessage(){
  try{
    const message:Message ={
     usertoid :this.User_id,
      userToProfile :{
        fristName : this.profiledata.firstName,
        lastName : this.profiledata.lastName
      },
      VectorTo : this.profiledata.image,
      userFromProfile :{
        fristName : this.usreprofiledata.firstName ,
        lastName : this.usreprofiledata.lastName,
      },
      userfromid : this.idlogin,
      VectorFrom : this.profiledata.image ,
      Content : this.content
    }
    await this.database.list('/messages').push(message);
    this.content2 = "";
    this.content = this.content2 ;
  }catch(e){
    console.error(e);
  }

}
```

7- Reporting:-



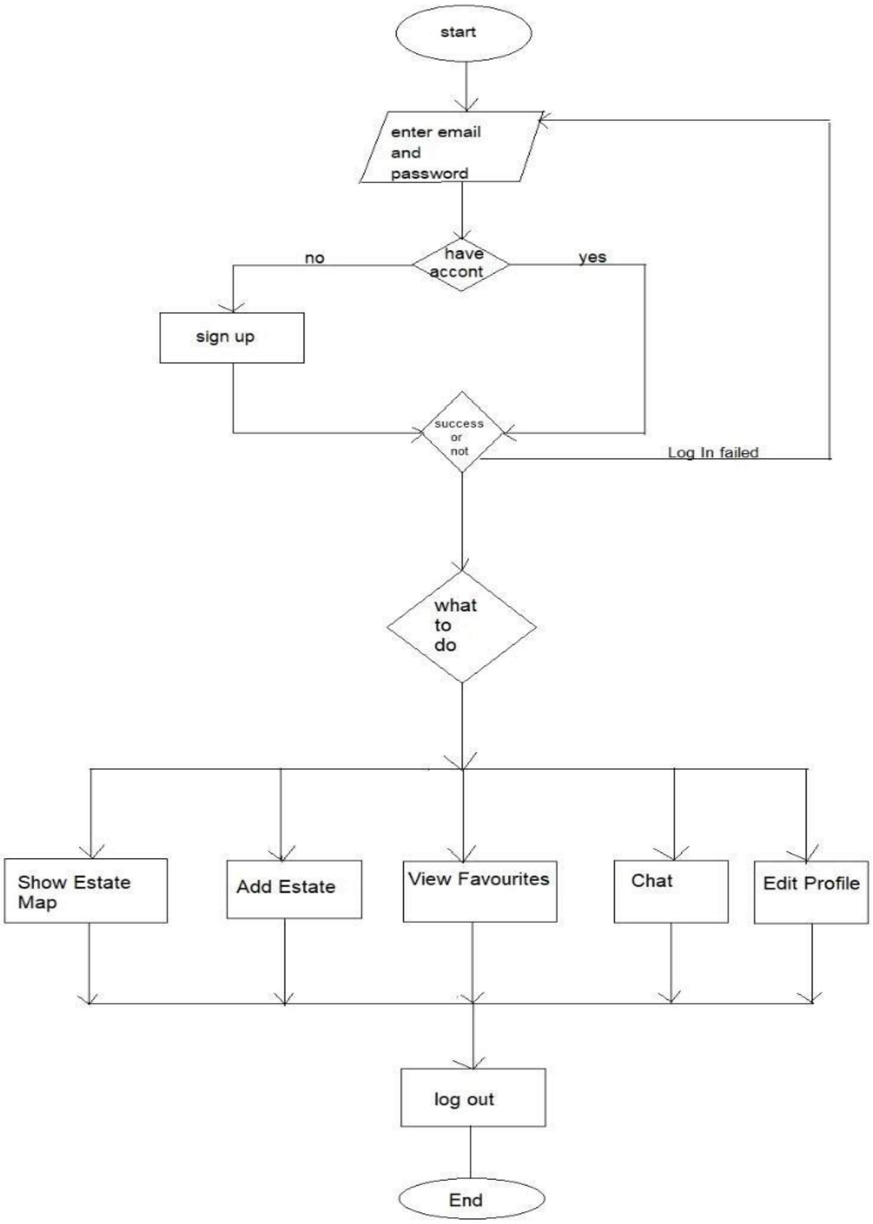
```
export class UserReportsPage {
  profiledata: FirebaseListObservable<Profile>;
  public profileList:Array<any>;
  public reportlistfinal:Array<any>;
  myarr:Array<any>;
  public reportid:string;
  public reportlistref:firebase.database.Reference;
  public profilelistref:firebase.database.Reference;
  public counter=0;
  public countlist:Array<any>;
  public countlistfinal:Array<any>;
  list={} as report;
  constructor(public navCtrl: NavController,private database:AngularFireDatabase, public navParams: NavParams
    , private alertCtrl: AlertController ,private afAuth:AngularFireAuth,
    private toastCtrl: ToastController) {

      this.reportlistref=firebase.database().ref('ReportUsers');
      this.reportlistref.on('value', countryList => {
      let reporter = [];
      countryList.forEach( country => {

        reporter.push(country.val());
        return false;
      });
      let b=[];
        for(let re of reporter){
          b.push(re.reportedid);

        }
        let reportfinal=[];
        for(let re of b){
            if(reportfinal.indexOf(re)==-1)
          {
reportfinal.push(re);
          }
```



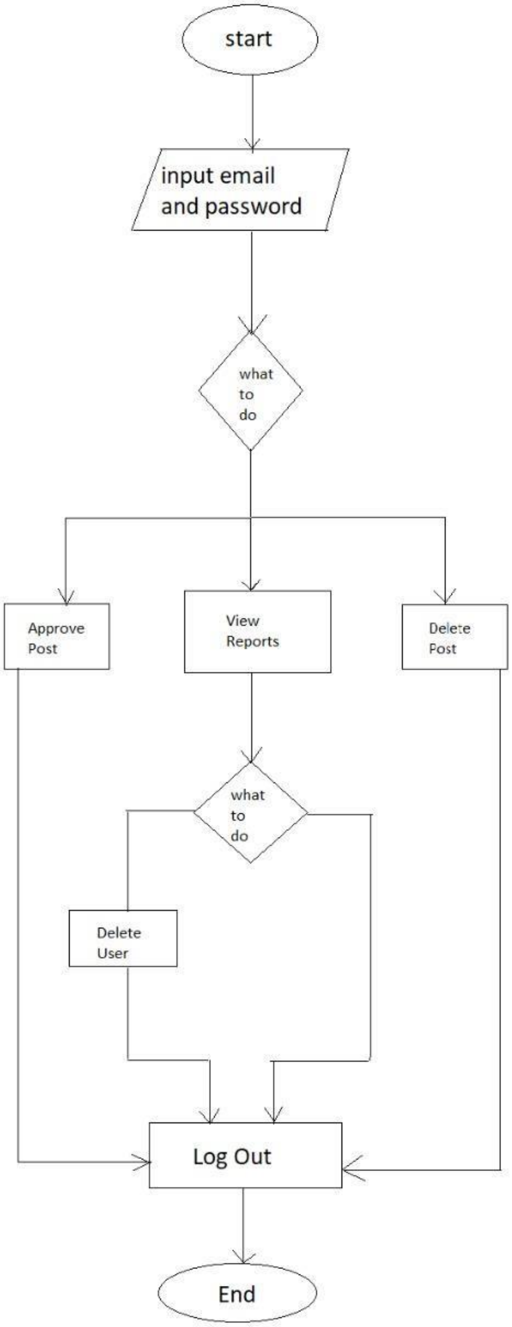
```
          {
reportfinal.push(re);
          }



          }
          this.reportlistfinal=reportfinal;
          console.log(reportfinal);
          let counts=[];
      for(let r of this.reportlistfinal)
      {
       const reportlistreffff=firebase.database().ref('ReportUsers').orderByChild("reportedid").equalTo(r);
       reportlistreffff.on("value",snap=>{
          this.myarr=this.snapToArray(snap);
              for(let ar of this.myarr){
                this.reportid=ar.reportedid;
                  this.counter++;
              }
console.log(this.reportid)
       })
       var arr = new Array();
       arr['counter'] = this.counter;
       arr['reportedid'] =this.reportid;
       this.countlist=arr;
     counts.push(this.countlist);

     this.counter=0;
      }
      console.log(counts);
      this.countlistfinal=counts;
      console.log(this.countlistfinal);
      for(let f of this.countlistfinal){
        console.log(f.reportedid);
      }
```

4.2 Flowchart

User Flowchart:-

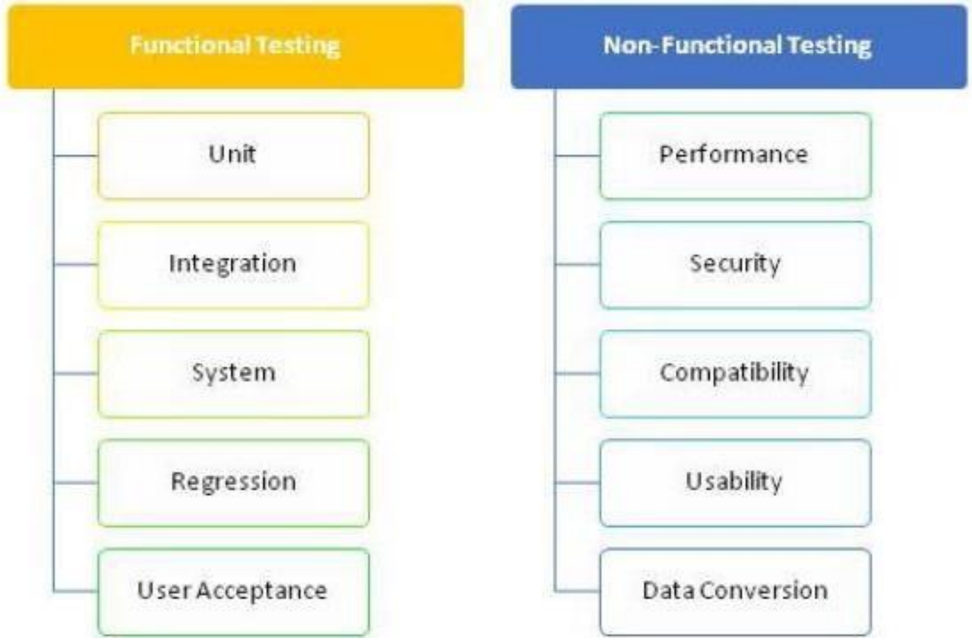


Admin Flowchart:-



Chapter 5 Testing

In this chapter we're going to discuss and go deeper in Settle mobile application's testing, and present the types of testing to be used and test cases we examined our application through.



Functional Testing:

5.1 Unit Testing

Testing of individual items (e.g. modules, programs, objects, classes, etc.)Usually as part of the coding phase, in isolation from other development item sand the system as a whole.

5.2 Integration testing

Testing the interfaces between major (e.g. systems level application modules) and minor (e.g. individual programs or components) items with in an application which must interact with each other.

System testing

Testing a system behavior as a whole when development is finished and the system can be tested as complete entity.

Regression Testing

To check older functionality after integrating new functionality.

Acceptance testing

Testing to ensure that a development is ready to be deployed into the business, operational or production environment.

Non-Functional Testing

Performance Testing

Accomplished a designated function regarding processing time and through put rate.

Load Testing

Measuring the behavior of within creasing load which can be handled by the component or system.

Stress Testing

Evaluate a system or component at or beyond the limits of its specified requirements.

Security Testing

Testing how well the system protects against unauthorized internal or external access.

Test cases

- **Test Scenario Objective:**

Verify log in successfully with correct email and correct password

- **Assumptions/Dependencies:**
  Valid email: shirin@yahoo.com
  Valid password: 123456

Step#	Description	Expected Result	Actual Result	Error Type
1.	Navigate to log in screen.	User navigate to login page		
2.	Enter email : shirin@yahoo.com	User can enter email		
3.	Press tab	Move to text field called password		
4.	Enter password : 123456	User can enter password		
5.	Click on login button	Log in will make successfully ,	Passed	

		home page will open		

- **Test Scenario Objective:**

Make sure that message will appear to tell you that you should enter

Valid email when you enter email not in firebase.

- **Assumptions/Dependencies:**
  Invalid email: d@yahoo.com
  Valid password: 123456

Step#	**Description**	**Expected Result**	**Actual Result**	**Error Type**
1.	Navigate to log in screen.	User navigate to login page		
2.	Enter email : d@yahoo.com	User can enter email		
3.	Press tab	Move to text field called password		
4.	Enter password : 123456	User can enter password		
5.	Click on login button	Message will appear to tell you that this email is not valid ,there is	Passed	

		no user in firebase with this email ,and still in log in page		

- **Test Scenario Objective:**

  Make sure that message will appear to tell you that you should enter

  Valid password when you enter wrong password.
- **Assumptions/Dependencies:**
  Valid email: shirin@yahoo.com
  Wrong password: 156

Step#	Description	Expected Result	Actual Result	Error Type
1.	Navigate to log in screen.	User navigate to login page		
2.	Enter email : shirin@yahoo.com	User can enter email		
3.	Press tab	Move to text field called password		
4.	Enter password : 156	User can enter password		

5.	Click on login button	Message will appear to tell you that this password is wrong, and still in log in page.	Passed	

- **Test Scenario Objective:**

  Make sign up successfully
- **Assumptions/Dependencies:**
  Valid email: shirin@yahoo.com
  Valid password: 123456
  Valid first name: Shirin
  Valid last name: Mahmood
  Valid phone: 01009675228

Step#	**Description**	**Expected Result**	**Actual Result**	**Error Type**
1.	Navigate to sign up screen.	User navigate to sign up page		
2.	Enter email : shirin@yahoo.com	User can enter email		

3.	Press tab	Move to text field called password		
4.	Enter password : 123456	User can enter password		
5.	Press tab	Move to text field called first name		
6.	Enter first name : Shirin	User can enter first name		
7.	Press tab	Move to text field called last name		
8.	Enter last name : Mahmood	User can enter last name		
9.	Press tab	Move to text field called phone		
10.	Enter phone : 01009675228	User can enter phone		
11.	Press tab	Move to next line called capture photo		
12.	Press on take photo using gallery or camera	User can capture photo		
13.	Click on register button	Sign up will make successfully , home page will open	Passed	

- **Test Scenario Objective:**

  Add new estate successfully

- **Assumptions/Dependencies:**

  Valid title : estate

  Valid address : 29st abd el hamed osman el nam

  Valid price : 500000

  Valid description : estate estate

Step#	Description	Expected Result	Actual Result	Error Type
1.	Navigate to add new estate screen.	User navigate to add new estate page		
2.	Choose type of estate	User can choose the type		
3.	Press tab	Move to text field called title		
4.	Enter title : estate	User can enter estate title		
5.	Press tab	Move to text field called address		
6.	Enter address : 29st abd el	User can enter address		

	hamed osman el naam			
7.	Press tab	Move to text field called price		
8.	Enter price : 5000000	User can enter price		
9.	Press tab	Move to text field called description		
10.	Enter description : estate estate	User can enter description		
11.	Press tab	Move to next line called choose location		
12.	Press on locate estate and this locate the estate using GPS or press on use the map to locate your estate	User can locate the estate		
13.	Press tab	Move to next line called capture photo		
14.	Press on take photo using gallery or camera	User can capture photo to your estate and you		

		can capture more than one photo		
15.	Click on save button	Add estate will make successfully , and message will appear to tell you that you added estate successfully	Passed	

Chapter 6
Results and Discussion

In this chapter we're going to find out the results of the project whether they're achieved or not and also the differences between the desired results and the actual ones.

6.1 Results

6.1.1 Expected Result

- The App shows the available estates for sale or rent
- The owner shows his estate and details about it with no need to a broker
- The customer searches the GPS to find the desired estate in the location he chooses
- The App connects the customers to the estate owner with no need to a broker “which is the App goal”
- The App should determine the availability duration of each estate so that the user can make a reservation

6.1.2 Actual Result

- The App shows the available estates for sale or rent
- The owner shows his estate and details about it with no need to a broker
- The customer searches the GPS to find the desired estate in the location he chooses
- The App connects the customers to the estate owner with no need to a broker “which is the App goal”
- The users can communicate with each other through chatting or phone calls
- Each user has the ability to block, report or rate another user or post.

6.2 Discussion

We managed to get the same expected result except for the reservation part and displaying the availability duration of each estate.

We’ve added new features into our application such as: Chatting, phone calls, blocking, reporting and rating

Chapter 7
Conclusion

Eventually, in this project we tried to make the best we can to avoid exploitation and it’s just a way of expressing our duty towards our society as we felt responsible enough to participate in making this community a better place with one of the most powerful tools we can use and we chose as our studying field years ago which is **TECHNOLOGY**.

In summary, our society would be more convenient if we got rid of exploitation and more people used this application. We should continue to try to help people make the best choices with no need to any kind of blackmail and we tried to do that with our limited fund and resources and we’re sure we’re going to expand this application and enhance its performance and fame if we’re just given the right resources and fund.

Finally, realizing the fact that **“a few clicks”** can do a lot to a society, will definitely make us more aware of the technology role and become better versions of ourselves.

Chapter 8
Future Work

-We are going to add more and more features to our application such as: Displaying the availability duration of each estate so that the user will be able to make a reservation

-We are seeking for having a fund for our project by sending proposals to companies or joining competitions.
-We also are going to enhance its performance and try deploy it on any of the available stores (as it's a cross-platform App).

-We're going to provide statistics functionality to calculate the number of estates sold or rented through our applications as well as the number of users and so on.

So we won't miss any chance to keep working as a team on that project and enhancing it after graduation as we're looking forward to turning this application into a start-up.
Our dream started with this idea months ago, and we'll insist on not letting go of it and not letting it be just a dream.

Bibliography

http://www.elfagr.com/2301875

https://www.masress.com/rosadaily/92316

https://creately.com

https://www.sciencedirect.com/science/article/pii/S0164121208000897

https://en.wikipedia.org/wiki/SRS

https://erdplus.com

https://en.m.wikipedia.org/wiki/Cost_estimate

https://www.smartdraw.com/sequence-diagram

https://www.youtube.com/watch?v=18_kVlQMavE

http://rushrash.com/domain-requirement/

https://play.google.com/store/apps/details?id=com.olxmena.horizontal

https://en.wikipedia.org/wiki/User_requirements_document

“After the darkness came the light.
After the exploitation came the solution!”

SETTLE

WHERE YOU SETTLE IN!

THANK YOU!