



Supervisor: Dr. Muhammed Arif

Authors:

Malak Muhammad basaad	434018269
Khulood Daifallah Al-Otaibi	434000814
Shoroog balgaith Al-Bargi	434002510
Doa'a Turkistani	435024517
Nebras talat sindi	434001666

Contact Information:

This project's report is submitted to the Department of Computer and Information System at Umm Al-Qura University in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

Team name:**Team Logo**

Estate Guiders

Credit Hours: four



Author(s):

Leader: Malak Muhammad basaad.

looka-14161@hotmail.com

Khulood Daifallah Al-Otaibi.

kokos20152@outlook.sa

Shoroog Balgaith Al-Bargi.

sho0sha1166@gmail.com

Doa'a Turkistani.

doaa.turkistani@hotmail.com

Nebras talat sindi.

npooo_t_s1416@hotmail.com

University Supervisor:

Dr. Muhammed Arif

mahamid@uqu.edu.sa

Department of Computer Science

Dept. of Computer Science Internet: <http://uqu.edu.sa>

Faculty of Computer and Information Systems Phone: +966 57576578

Umm Al Qura University Fax: +966 545487765

Kingdom of Saudi Arabia

INTELLECTUAL PROPERTY

RIGHT DECLARATION

This is to declare that the work under the supervision of Dr. Muhammad Arif having title “Saudi Estate” carried out in partial fulfillment of the requirements of Bachelor of Science in Computer Science is the sole property of the Umm Al-Qura University and the respective supervisor and is protected under the intellectual property right laws and conventions. It can be only considered/used for purposes such as an extension for further enhancement, product development, adoption for commercial/organizational usage, etc., with the permission of the University and respective supervisor.

This above statement applies to all students and faculty members.

Date: _____

Author(s):

Name: _____

Signature: _____

Supervisor(s):

Name: Dr. Muhammad Arif.

Signature: _____

ANTI-PLAGIARISM DECLARATION

This is to declare that the above publication produced under the supervision of Dr. Muhammad Arif having the title “Saudi Estate” is the sole contribution of the author(s) and no part thereof that has been reproduced illegally (cut and paste) which can be considered as Plagiarism. All the referenced parts will use to argue the idea and will cite properly. I/We will be responsible and liable for any consequence if a violation of this declaration is proven.

Date: _____

Author(s):

Name: _____

Signature: _____

ACKNOWLEDGMENTS

All praise due to Allah for giving us the strength to complete this research part from a graduation project. This effort is dedicated to the Department of Computer Science, many thanks and gratitude for the help going through this experience, also to the great supervisor Dr. Muhammad Arif for supervision, guidance, contributions and patiently guiding us in the writing of this research and make it as good as possible.

ABSTRACT

In today's technology world, Clients are much more knowledgeable about the concept of communicating and easy access in the business field that play a significant role. Today the real estates are grappling with a number of challenges such as shortage of rapidly access and escalating commission cost, lack of information and resources needed. Meanwhile, developing an application for real estates will help and contribute to avoid these impurities and to react to the changing market conditions more effectively and efficiently under the goal of satisfying clients. Thus, we propose an application for the real estates around Saudi Arabia with the availability of data to facilitate the search with the specific information required and a direct channel to reach them, and a travel flow data from Google map to find the best location in short time and other services as well described in the upcoming chapters. The proposed application has quite satisfactory performance in comparison with other existing applications.

GLOSSARY

Database (DB)	A Collection of data that is organized.
DFD	Data Flow Diagrams.
Application	A Program can be for general or special purpose.
UML	Unified modelling language diagrams.
UCD	Use Case Diagram.
IDE	Integrated development environment.
Functional requirements	Functional requirements characterize what your system should do.
Non-functional requirements	Non-functional requirements characterize how your system works.

CONTENTS

ACKNOWLEDGMENTS	iv
ABSTRACT	v
GLOSSARY	vi
Chapter 1 – INTRODUCTION	
1.1 Purpose of the Project.....	14
1.2 Project Aims and Objectives	14
1.2.1 Provide Multi-objective pare to search of estate.....	15
1.2.2 Provide location of the estate and visual Information about the estate.....	16
1.2.3 Provide Online Chat Service	16
1.2.4 Provide Notifications	17
1.2.5 Provide customizable search tool	17
1.2.6 Sharing selected estate to other social media applications or through SMS message	17
1.2.7 Provides real estate listing and locations	17
1.2.8 Provides nearby School and shopping	17
1.3 Purpose of This Document	19
1.4 Overview of the Document.....	19
1.5 Literature Review	20
1.5.1 existing Applications.....	21
1.5.2 Detailed Features.....	22
1.5.3 Problems in the Existing Applications.....	22
1.6 Expected Outcome.....	22
1.7 Method/Approach.....	23
1.7.1 Requirement Analysis.....	23
1.7.2 System Design	24
1.7.3 Implementation.....	24
1.7.4 Testing	24
1.7.5 Deployment	24
1.7.6 Maintenance	24
Chapter 2 –SYSTEM ANALYSIS	
2.1 Data Analysis.....	26
2.1.1 System Requirements	26

	Saudi Estates
2.1.1.1 Requirements Elicitation	26
2.1.1.2 Functional and Data Requirements.....	32
2.1.1.2.1 Project Stakeholders.	32
2.1.1.2.2 Functional Requirements.....	32
2.1.1.2.3 Non-Functional Requirements.....	34
2.1.1.2.3.1 Look and Feel Requirements.	34
2.1.1.2.3.2 Usability Requirements	35
2.1.1.2.3.3 Access control	35
2.1.1.2.3.4 Performance Requirements.....	35
2.1.1.2.3.5 Portability Requirements.	35
2.1.1.2.3.6 Accessibility	35
2.1.1.2.3.7 Availability.....	35
2.1.1.2.3.8 Privacy Requirements.....	35
2.1.2 Data Flow Diagrams.....	35
2.1.3 Unified Modeling Language Diagrams.	40
2.1.4 Proposed Solutions and Alternative Solutions.....	53
Chapter 3 –DESIGN CONSIDERATION	
3.1 Database Constraints	55
3.1.1 Database Schema	55
3.2 Design Constraints.....	58
3.2.1 Hardware and Software Environment.....	58
3.2.2 End User Characteristics.	60
3.3 Architectural Strategies.	61
3.3.1 Algorithm to be used	61
3.3.2 Project Management Strategies.	63
3.3.3 Development Method	63
3.3.4 Future Enhancements/Plans.....	64
Chapter 4 - SYSTEM DESIGN	
4.1 System Architecture and Program Flow	66
4.1.1 Major Modules.	66
4.1.2 Sub Modules.....	67
4.2 Detailed System Design.....	68
4.2.1 Interface Description	68
Chapter 5 - IMPLEMENTATION AND TESTING	
5.1 Implementation Phase	85
5.1.1 Connecting part.....	85

5.1.1.1 Connecting the Application with Database.....	85
5.1.2 Designing Interface.....	88
5.1.2.1 User Interface.....	88
5.1.2.2 Admin Interface.....	
5.2 Test.....	35
5.2.1 Test Tools and Environment.....	35
5.2.2 Testing Scenarios.....	35
5.2.2.1 Exception Handling.....	36
5.2.2.2 Completed Scenarios.....	35
5.3 Challenges.....	
5.4 Conclusion.....	

TABLES**Chapter 1 – INTRODUCTION**

Table 1- Comparison between Applications functionalities.....	21
---	----

Chapter 2 –SYSTEM ANALYSIS

Table 2: log in.....	42
Table 3: log out.....	42
Table 4: Delete Estates.....	42
Table 5: sign up	42
Table 6: Add Estates.....	43
Table 7: Update Estates.....	43
Table 8: Data Best Traffic	43
Table 9: Edit Profile	43
Table 10: log in for Admin	44
Table 11: Communicate by chat	44
Table 12: Location.....	44
Table 13: View Report	45
Table 14: Specifications Estate.....	45
Table 15: View Estate.....	46
Table 16: Send Report	46
Table 17: Add to Black list.....	46
Table 18: Send notification.....	47
Table19: Trigger notification	47

Chapter 3 –DESIGN CONSIDERATION

Table 20: show admin database details.....	56
Table 21: show search by location database details.....	56
Table 22: show specific search database details	56
Table 23: show location database details.....	57
Table 24: show estate database details	57
Table25: show user database details.....	57

Chapter 4 - SYSTEM DESIGN

Table 26: Major modules.....	66
Table 27: submodules.....	67
Table 28: Details task	87
Table29: project timeline.....	88

FIGURES

Chapter 1 – INTRODUCTION

Figure 1: pare to search.....	15
Figure 2: waterfall model phases.....	23

Chapter 2 –SYSTEM ANALYSIS

Figure 3: Context Diagram of DFD.....	36
Figure 4: Level 0 of DFD	37
Figure 5: Level 0 of DFD	38
Figure 6: Child Diagram for process 4 in level 0(user entity)	39
Figure 7: Child Diagram for process 6 in level 0(user entity)	40
Figure8: use case diagram	41
Figure 9: Activity Diagram.....	49
Figure 10: ERD Diagram.....	50
Figure 11: User Sequence Diagram.....	51
Figure 12: Admin Sequence Diagram	51
Figure 13: Class Diagram.....	52

Chapter 3 – DESIGN CONSIDERATION

Figure14: show user database.....	55
Figure 15: show admin database.....	55
Figure 16: show search by location database.....	55
Figure 17: show estate database	55
Figure 18: specific search database	56
Figure 19: show location database.....	56
Figure 20: firebase analytics.....	59
Figure 21: How Everything work	60

Figure22: A population of five solutions 62

Chapter 4 - SYSTEM DESIGN

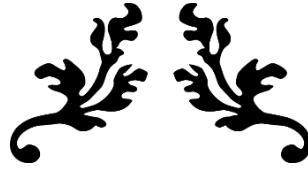
Figure 23: Interface of the welcome page.....	68
Figure 24: Interface to enter the user	68
Figure 25: Interfaces of Login form	69
Figure 26: Interfaces of sign up forms.....	70
Figure27: Interfaces of Home page	71
Figure 28: Interfaces of User profile.....	72
Figure 29: Interfaces of My Estates.....	73
Figure 30: Ad posting agreement interface.....	73
Figure 31 Interfaces of Add Estates.....	74
Figure 32: Interfaces of Add Estates.....	74
Figure 33: Interfaces of Advertising.....	74
Figure 34: Interfaces of Advertising.....	75
Figure 35: Interfaces of Google map.....	76
Figure36: Interfaces of Settings.....	77
Figure 37: Interfaces of Chat	78
Figure 38: Interfaces of Admin.....	79
Figure 39: lock screen notification	80
Figure 40: trigger notification.....	81
Figure 41: specific search	82
Figure 42: Interfaces of Log out	83

Chapter 5 - IMPLEMENTATION AND TESTING

Figure 43: The Assistant tool window in Android Studio.....	85
Figure44: Firebase Console part1.....	86
Figure45: Firebase Console part2.....	87
Figure46: splash screen.....	88
Figure47: login, signup and skip.....	89
Figure48: sign up.....	89
Figure49: create new account.....	90
Figure50: verify your email.....	90
Figure51: email has been verified.....	91
Figure 52: log in.....	92

Figure 53: home interface.....	94
Figure 54 :search result.....	95
Figure 55 :navigation drawer.....	96
Figure 56: My Estate.....	98
Figure 57: Terms and policies.....	98
Figure 58: Add Estate	99
Figure 59: assess permission.....	99
Figure 60: select location	100
Figure 61: Estate details.....	101
Figure 62: Estate details.....	102
Figure 63: share button.....	102
Figure 64: share to whats up.....	103
Figure 65: Estate details.....	104
Figure 66: report problem.....	104
Figure 67: delete button.....	105
Figure 68: favorite list.....	107
Figure 69: chat.....	108
Figure 70: Settings.....	110
Figure 71: change password.....	111
Figure 72: change password email.....	111
Figure 73: reset password.....	112
Figure 74: password changes.....	112
Figure 75: Edit profile.....	114
Figure 76: terms.....	115
Figure 77: messag1.....	116
Figure 78:message2.....	116
Figure 79: messag3.....	117
Figure 80: message4.....	117
Figure 81: message5.....	117
Figure 82: message6.....	117

Figure 83: message7	118
Figure 84: message8	118
Figure 85: message9	118
Figure 86: message10	118
Figure 87: message11	118
Figure 88: message12	119
Figure 89: message13	119
Figure 90: message14	119
Figure 91: message15	119
Figure 92: message16	119
Figure 93: message17	120
Figure 94: message18	120
Figure 95: project Poster	133



CHAPTER1:

INTRODUCTION

1.1 Purpose of The Project

1.2 Project Aims and Objectives

- 1.2.1 Provide Multi-objective pare to Search of estate.
- 1.2.2 Provide location of the estate and visual Information About the estate.
- 1.2.3 Provide Online Chat Service.
- 1.2.4 Provide Messaging Service.
- 1.2.5 Provide Notifications.
- 1.2.6 Provide customizable search tools.
- 1.2.7 Sharing selected estate to other Social media applications.
- 1.2.8 Provides real estate listing and locations.
- 1.2.9 provides nearby School and Shopping.

1.3 Purpose of This Document

1.4 Overview Of the document

1.5 Literature Review

- 1.5.1 Existing applications.
- 1.5.2 Detailed Features.
- 1.5.3 Problems in the existing system.

1.6 Expected Outcome

1.7 Method/Approach

- 1.7.1 Requirement analysis.
- 1.7.2 System design.
- 1.7.3 Implementation.
- 1.7.4 Testing.
- 1.7.5 Deployment.



CHAPTER ONE: INTRODUCTION

In the present era, the idea of searching for an estate became frustrating and tiring idea for anyone to find the estate that suits him and suits his budget, therefore we think it is rare for people to resort to real estate's offices because it could be a waste of time and effort.

Today Real Estate Industry has been growing rapidly and mobile app plays a very important role in the real estate company at this time.

Our project falls under the part: information system, it is an application so we will use the programming and database design. Thus, we aim to develop an application able to create desired interactivity to encourage users to explore the best possible results out of their searches with minimum efforts in very less time.

The very first use of these real estate's mobile apps is that we give users a direct channel to reach them and the users will be able to use these apps at any time right the clock. In addition, feature visualization of the estate with our interactive maps and the users will become informed on the estate and area demographics and most of the real estate applications display their estates attached with pictures, video, and description.

1.1 PURPOSE OF THEPROJECT

Each year the commercial real estate market changes and the estate apps keep on the increase with different features to attract people and due to the large area of Saudi Arabia and plenty of inhabitants means a larger number of houses, white lands and the larger number of people who are looking for a house for live in it. The purpose of this project to make it easier for people to look for what suits them from houses and lands and to increase the welfare of residents in the Kingdom Saudi Arabia, where they will be able to be searching for houses and lands any place any time, whether in their home or work in the morning or evening easily pleased and for free. People who are planning to buy or rent a house or land they will be able to follow the news and prices.

The buyers will be informed about the lands offered in many areas, and making it easy for those who want to rent or buy a house from a city far away from his city where he will see the location of the house or land and see what commercial centers and hospitals that near to him and he will be able to calculate the time to go from the house to many places that he can identify it on the map like his children's school and his work.

1.2 PROJECT AIMS AND OBJECTIVES

Use of mobile devices to create an effective, useful and appropriate real estate's environment for users to stay in permanent touch. We will design and develop an application for Android operating system to help

many mature people that like to buy and get estates without wasting much time, we establish application to be easy to learn and Provides good and useful content and a consistent appearance and support travel flow data from Google to find the best location of estates in term of travel time, in addition, the application does not take any amount of money in any sale or rent, and it is done to encourage people to use modern technologies of communication and see their good aspects. Moreover, we aim to support all other platforms in the future. In this project, we are making an application for buying and selling estate with the following aims.

1.2.1 Provide Multi-objective pare to search of estate

Whenever we are searching for a suitable home for our comfortable residence, we consider a lot of decision variables or objectives like house rent, house location, accommodation, status of the building, locality etc. Finding the best solution is a very tedious job and we must visit lots of estate offices physically. In our application, we are providing a framework to select different objectives to select the best possible estate. The objective functions can be integers, real-valued or categorical.

Let the vector-valued objective function is defined as

$$f: X \rightarrow \mathbb{R}^n, (x) = (f_1(x), f_2(x), \dots, f_n(x))^T$$

A multi-objective search can be formulated as follows,

$$\min(f_1(x), f_2(x), \dots, f_n(x)), x \in X$$

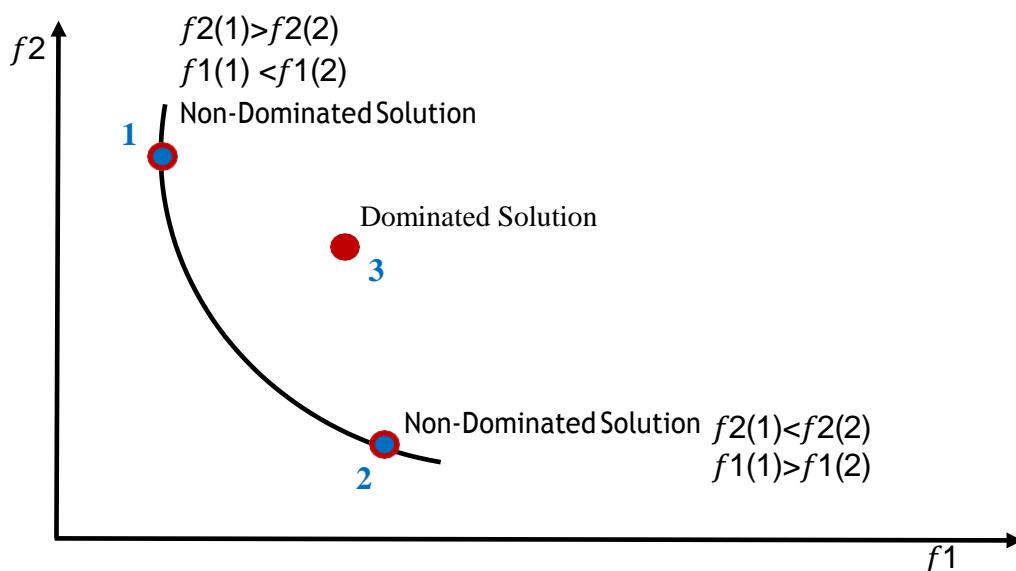


Figure 1: pare to search

If some objective function is to be maximized, then it is equivalent to minimize its negative. In our context, feasible solutions are all the available estates entered by the users for sell or buy. Pareto solutions are set of non-dominated solutions. Non-dominated solution means that this solution is not dominated by other solution in all its attributes. In the figure below there are three solutions plotted for two objective functions. Solutions 1 and 2 are non-dominated whereas; solution 3 is dominated by solution 1.

In our application, objective function corresponds to user-selected objectives which may include all or any of house rent, house location, accommodation, status of the building, locality etc. Our application will provide a list of non-dominated solutions (estates) by searching all the available estates stored in our database. In case of non-conflicting objectives, non-dominated solutions will be one but in case of conflicting objectives, non-dominated solutions can be many. The user can select one of the non-dominated solution depending on his preference about the objectives.

There can be multiple scenarios of multi-objective pareto search.

Scenario 1: Finding best estate according to the properties of the estate

In this scenario, user can select various attributes of the estate. For example, for hiring a house, attributes can be house rent, house location, accommodation, status of the building, locality etc. Whereas, for buying the estate, attributes can be price of estate, location of estate, accommodation or area, status of the building, locality etc. The search algorithm of the application will search the available estates and provide the user the best solution whenever the objectives are non-conflicting or provide a set of non-dominated solutions.

Scenario 2: Finding best estate according to the location of the estate

This is an interesting and novel scenario of our application. In this scenario, the user wants to find the best estate in the city that is the best located in the city according to a set of objectives. For example, in the family the husband goes to place P1 for work in the morning at time T1 and return in the evening at time T2. His wife goes to the place P2 in the morning at time T3 and return at time T4. His children go to the school located at P3 in the morning at time T5 and return at time T6. Our application will connect to the Google map and Google traffic flow data (if available) and decides the objectives as amount of traveling time spent on going and returning from different places the family members are going according to the distance and traffic flow. For every estate available in the database, the application will calculate the objective functions for every member of the family and suggests an estate, which optimize the travelling time for all the family members.

1.2.2 Provide location of the estate and visual Information about the estate

Application can show the location of the estate, any pictures, audios or videos associated with this real estate. Also on the Google map, application can show you different estates locations. If the user select one objective, color-coded locations of the real estate in the locality can be shown according to that objective.

1.2.3 Provide Online Chat Service

The user can chat online in the application to communicate with the advertiser and other users.

1.2.4 Provide Notifications

There can be different types of notification services. It can be a notification of new addition of the estate or removal of the estate. It can be a notification about the decrease in the price of a selected estate. A user can select notification options like the addition of new estate in a selected area, decrease in the price of a selected estate, availability of the estate according to the user specifications.

1.2.5 Provide customizable search tool

There can be different customizable search tools, it can be a Specification search tool for the user he can search by his specification, for example, the user chooses for sale/for rent estate, Estate type, price (he can determine the maximum/minimum price) and city/area. A search tool by location, for example, the user can search with locality name and list of estates will be appeared. The user can get the benefit of near me feature to find the estates near his location.

1.2.6 Sharing selected estate to other social media applications or through SMS message

The user can get instant feedback for given estate when sharing it with other user or/and through social media application like WhatsApp, twitter etc.

1.2.7 Provides real estate listing and locations

In our application, there will be a list of new estates recently added with the details. Moreover, a map of Saudi Arabia, if a user clicks on city/area on the map it will show the available estates.

1.2.8 Provides nearby School and shopping ratings

The Google provides rating for the nearby services that intended to stay and hire house around.

The Saudi Estates:

There are some of the main features of our project:

- Search based on specifications for the estate such as specific city, type of estate (like:

land, villa, apartment and building), cost of the estate, number of floors, rooms, baths..etc).

- Search by location with near me feature that enables the user to discover real estates around him.
- Support travel flow data from Google Map to find the best location in term of travel time (for example: if you want to search for a house by location, you can access to find location easily in minimum time and that's will apply to the other services as well like (Hospitals, Schools, work ...etc.)).
- Provide adequate information about the desired area such as if there are any available services nearby the estate like educational, medical complexes...etc.
- Access to the advertiser contact by chat online and completion the purchases process easily.
- Allow the user to add an estate and offer it for sale or rent attached with the announced estate picture, video, and audio of the description and a map of the estate location.
- The app has a notification feature, it will send notifications about the recent estates, direct Messages and welcome messages for the new users.
- The user can send report to the administrators about the estates in the application for example: if he saw an advertisement and used the Google map to access to that estate and it was not like what he described in the advertisement.
- The administrator will receive the report from the user and start verifying it by taking estate ID and decide whether it followed the terms of our application or not, to take the proper action for the advertiser like add him to black list or decline the user's report.
- The admin will delete the estate from the database when the advertiser does not update the estate after passing period amount of time.
- Share feature that allows a user to share the estate with social media applications (for example: WhatsApp, Facebook and twitter ...etc.) and through SMS messages.

- Allow users to make a list of favorite estates.
- The application supports both Arabic and English languages according to the user's choice.

With these features, it will contribute to enhancing people trust in these applications and to be aware of its importance. In addition, we will solve the traditional search problem for real estates whether for sale or rent and allow users to display their estate easily, this contributes in their marketing with a new concept, and technology and more developed.

Advantages of the proposed system:

- Save time for investors, buyers, and renters in searching and choosing the required estate.
- Speed and easy access between all the users (by: chat online).
- The ability to see the estate without having to go there.
- Provide usage instructions for the users to make them capable to understand the application.

1.3 PURPOSE OF THIS DOCUMENT

The purpose of this document is to explain fully about our project and the mechanism of our project in building and using the application and how to reach the expected outcome and also describes the features available to both the seller and the renter.

1.4 OVERVIEW OF THE DOCUMENT

In this section, we will sketch out a complete picture of the project. The document should involve these chapters:

Chapter 1 Introduction

In this chapter, we will focus on the main ideas of the project and its purposes, describing the existing system mentioning its problems and comparing our APP with the other. In addition, we will take an overview of Expected outcome and Method /Approach.

Chapter 2 System Analysis

In this chapter, we will talk about the requirement of the project, identifying system-modeling diagrams, defines the functional and non-functional requirement specification and describe the proposed system solutions with alternatives.

Chapter 3 Design Consideration

In this chapter, we will present hardware and software environment, the architectural strategies and mention the plans of the project.

Chapter 4 System Design

In this chapter, we will present the major and subcomponents. Also, will display different interfaces for the proposed system.

1.5 LITERATUREREVIEW

Smart applications have become important in all areas and their use in the field of estates and it has an effective and important role in strengthening and speeding up the business process so many of the applications that can be used in this area have emerged. We will explain some of the real estate and the functions they have:

- Matterport [1]: is a new 3D camera that is designed to scan the interior of homes. A new age of real estate's really helps show off the home in a way we haven't really been able to experience before. When selling a home, you can upload it easily when log in to your matterport by scanning and moving the camera through the home, it will shoot clearly and then stitch it all together within a few hours to produce your own virtual home tour.
- Homesnap Real Estate [2]: its application can help you find houses. There are three different ways to search, first, you can search by how many rooms you need the second search by the name of the neighborhood, third you can search by school zones and find dates for open houses. There are pictures of the houses from inside and outside. The application includes a chat to communicate, Material Design.
- Lamudi [3]: is the one-stop shop for finding your new home. Easily search properties according to your criteria or area and contact the agent directly by this app you can customize your search via filter and find properties that meet your criteria. You can filter by location, buy or rent property type, price, size, and facilities.
- Housing-Real Estate & Property [4]: The application helps to search to buy or rent flats, apartments, and villas. In this app, you can see the bus stop, schools, malls restaurants. Explore home listing for sale and rent and each list has detail information along with photographs.

1.5.1 Existing applications

Many applications are related to our application. Therefore, we will make comparison table of some of the real estate apps with common functions in the [Table1](#).Also, some of the good features and problem of the existing apps we will mention in section 1.6.2 and section 1.6.3 in the row.

Application name Functionnalities	Matterport [1].	Homesnap Real Estate [2].	Redfin [5].	Houses.com [6].	Saudi Estates
Multi-objective search	---	---	---	---	YES
share feature that you can share it with your relative or friends and other applications.	YES	YES	YES	---	YES
offer properties for sale.	YES	YES	YES	YES	YES
sale and purchase lands.	---	---	---	---	YES
allows the buyer or tenant to see the pictures information about the properties.	YES	YES	YES	YES	YES
support travel flow data from Google to find the best location in term of travel time.	---	---	---	---	YES
Search based on objectives and all possible options such as (location, an area of estate, cost, land area...etc).	---	YES	YES	---	YES

Table 1- Comparison between Applications functionnalities

1.5.2 Detailed features

Some of the detailed features of the existing applications:

- "New home" for the last homes that were added to the app.
- "Notify me when the price change" to be aware of the last change about the price.
- Estimate the value of the house according to market conditions.
- provide the "Update feature" to update the advertisement after a period amount of time.

Get to know the area and find shops, schools and places of convenience close to the property with our "Explore this neighborhood" function.

1.5.3 Problems in the existing applications

Looking at the current problems in applications we found some weaknesses such as:

5.1.2.1 The advertisement doesn't update continuously to avoid being deleted after passing a specific period of time, and therefore the estate could have been sold oriented.

5.1.2.2 doesn't provide a full review about estates (such as price).

5.1.2.3 Some of the application doesn't expand to a lot of areas and doesn't show the nearby services.

5.1.2.4 don't clarify the selling process between the seller and buyer.

5.1.2.5 Some of the app like lamudi: display the estates that are located in some cities for example if you choose to display estates in Saudi Arabia you must select one city from three cities (Makkah, Jeddah, and Riyadh).

5.1.2.6 Some of the app does not provide a user profile to enable them to modify their personal information if needed, just keep the mobile number that entered at the beginning of registration in the application.

5.1.2.7 There are app support only popular cities when the user set filters for locality the filter doesn't work, no map view to get a larger picture of available properties in an area.

1.6 EXPECTED OUTCOME

Developing an application that uses travel flow data on maps to enhance finding locations of estates easily. therefore, saves time and effort. Moreover, rapidly access to the contacts of an owner and schedule an appointment with advertiser to complete the purchase process, and provide estates from around the kingdom.

1.7 METHOD/APPROACH

We will use the waterfall methodology. We have defined the major phases the project needs to go through. The following is a diagram that shows the phases we will go through in the waterfall model [7]:

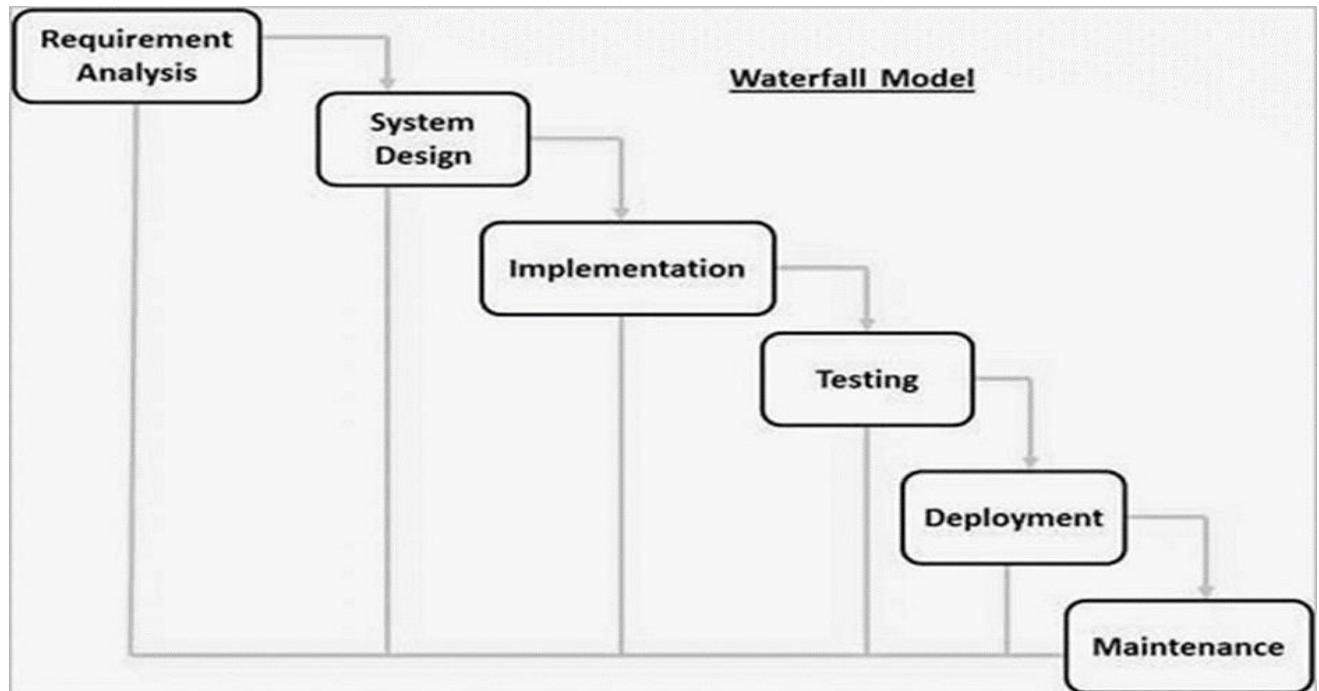


Figure 2: waterfall model phases

The waterfall model is a design process often used in software development. This model takes its name from the cascading "flow" of its life cycle, which moves downward (akin to a waterfall) through the stages of development.

In the waterfall model, the phases of development do not overlap. Each stage must be completed before the next "level" of the waterfall is begun. This makes the waterfall model most useful for smaller projects with well-defined requirements and fewer uncertainties.

The waterfall model was developed in the 1970s by Winston Royce to prevent costly revisions late in a product's development. With this model, if a revision is needed, it is easier and cheaper to make that revision early. When using the waterfall model, the emphasis lies in making sure the requirements and design fit the needs of the project before moving on to the later stages of development.

1.7.1 Requirement analysis

In this phase, we will gather information about what has been done in education/history applications field in Saudi Arabia, USA, and the UK. In addition, we will make a survey to differentiate between application and other similar applications also reviewed the negative aspects and the problems these

applications have in order to avoid them while developing our application as well as adding some missing valuable features. Once the information is gathered, we will start documenting what info has been decided to use in order to explain the project's goals and reasons of choosing this project and to preview the desired goals of the document.

1.7.2 System design

After identifying the goals of the project and requirement analysis, we will do a blueprint for the coders with a plan for meeting the requirements. We will define architectural strategies in the project, reusing existing software components, project management strategies, and development method to be used. Furthermore, we will present the detailed system design and the component description.

1.7.3 Implementation

After the designing phase is completed we will start coding, but before we will design the wireframes of the application, after that we will use Android Developing to implement the interfaces besides the application itself.

1.7.4 Testing

Once the application is implemented, we will do a software stress-tested and combed over for errors, and make sure the product meets the client's requirements.

1.7.5 Deployment

After testing and debugging are completed, we will implement the product according to the agreed-upon requirements. We will do another round of testing and verification often follows after implementation.

1.7.6 Maintenance

After we deliver the product to the client, we will put a scheduled maintenance and support into place, to ensure the product continues to work as designed [8].



CHAPTER 2:

SYSTEM ANALYSIS

2.1 Data Analysis

2.1.1 System requirements

- 2.1.1.1 Requirements elicitation
- 2.1.1.2 Functional and data requirements
 - 2.1.1.2.1 Project stakeholders
 - 2.1.1.2.2 Functional requirements
 - 2.1.1.2.3 Non-Functional requirements
 - 2.1.1.2.3.1 Look and feel requirements
 - 2.1.1.2.3.2 Usability requirements
 - 2.1.1.2.3.3 Access control
 - 2.1.1.2.3.4 Performance requirements
 - 2.1.1.2.3.5 Portability requirements
 - 2.1.1.2.3.6 Accessibility
 - 2.1.1.2.3.7 Availability
 - 2.1.1.2.3.8 Privacy Requirements

2.1.2 Data flow diagrams

2.1.3 Unified Modeling Language Diagrams

2.1.4 Proposed Solutions and Alternative Solutions



2.1 Data Analysis

For any application or project, you should specify requirements to make it easier later for the designer to build his project fully and properly.

The system requirements and its specification will be discussed in this chapter.

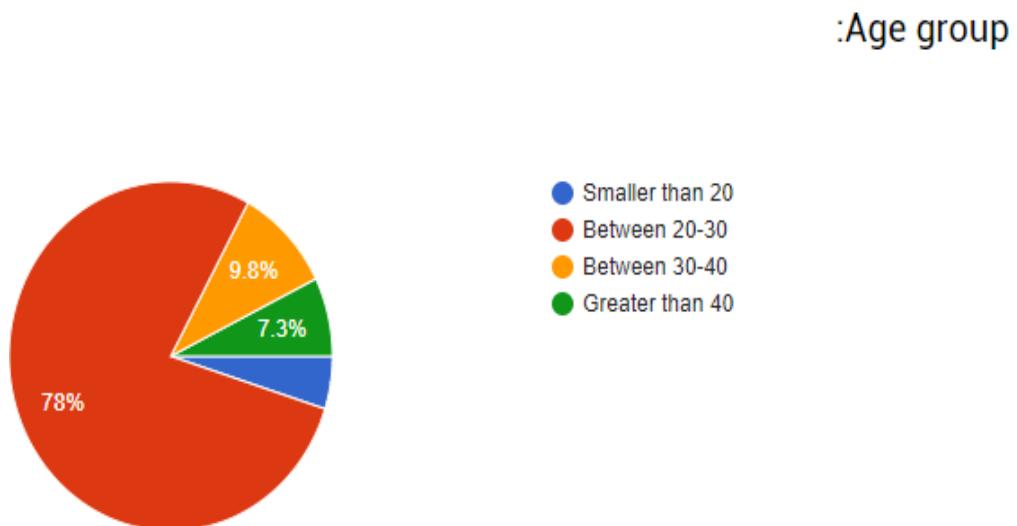
2.1.1 System requirements

2.1.1.1 Requirements elicitation

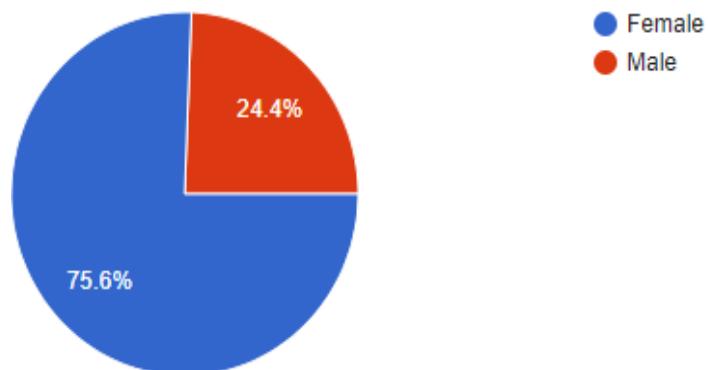
We have gathered data through conducting a questionnaire about estates applications and some people's view of them.

Questionnaire: List of research or survey questions asked to respondents, and designed to extract specific information. we chose the questionnaire to know people's opinion since it serves purposes:

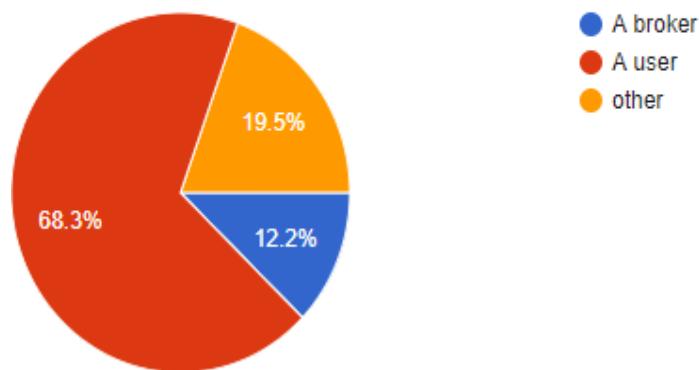
- collect the appropriate data.
- make data comparable and amenable to analysis.
- The information can be collected from a large number of people in a short period of time and relatively cost-effective way.



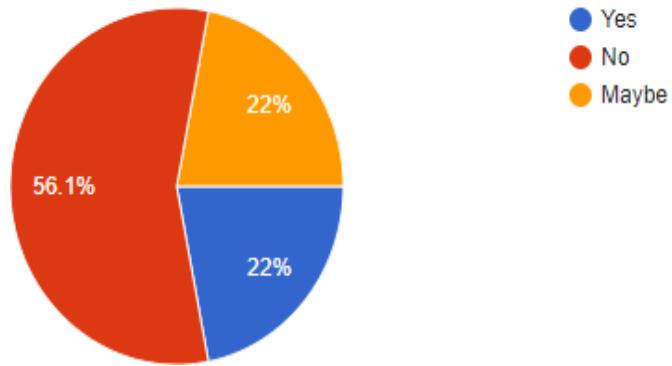
:gender



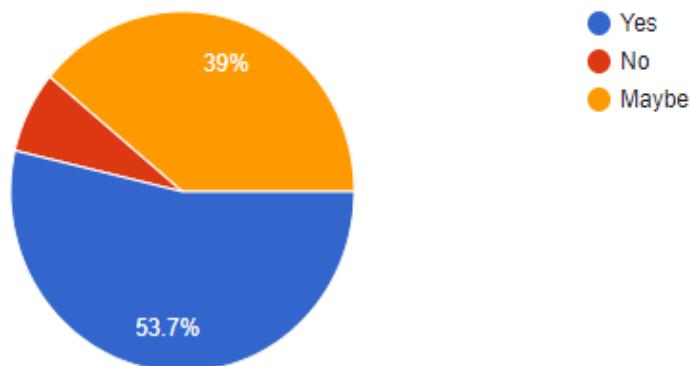
..Describe yourself, are you



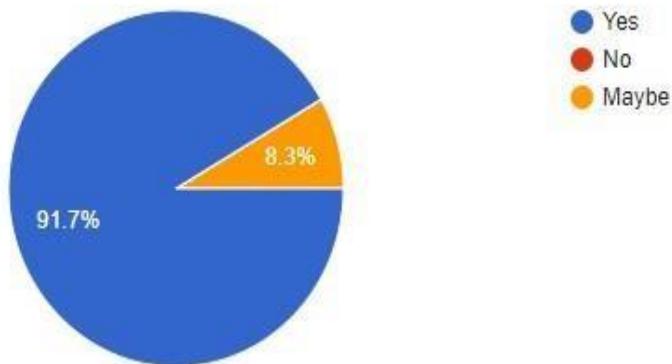
?Do you have an experience of buying and selling real estate



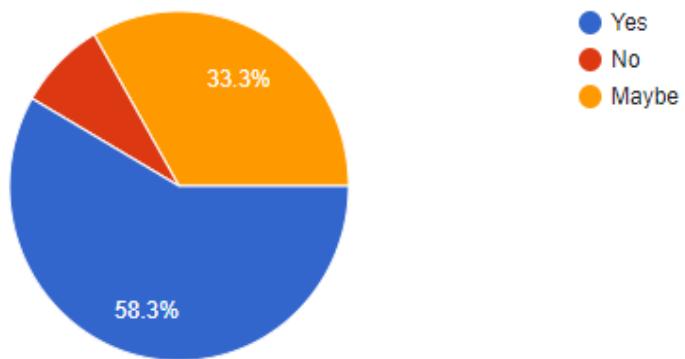
?If we design an application for real estate will you use it or not



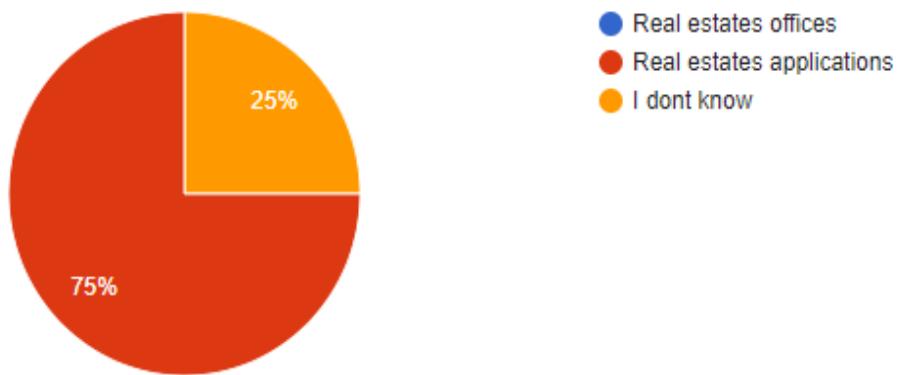
Does the real estate application will help on saving costs , efforts and
?time



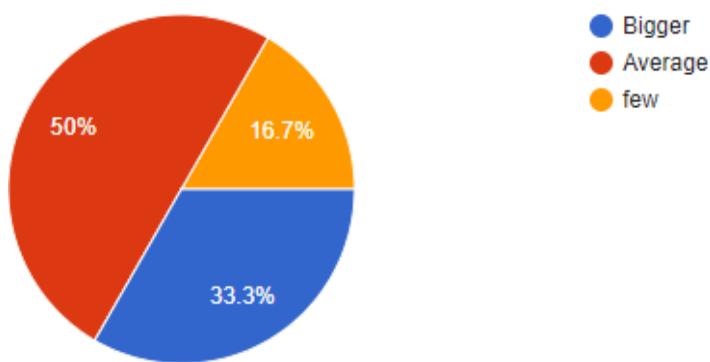
?Do you find difficulties in informing or receiving real estates offers



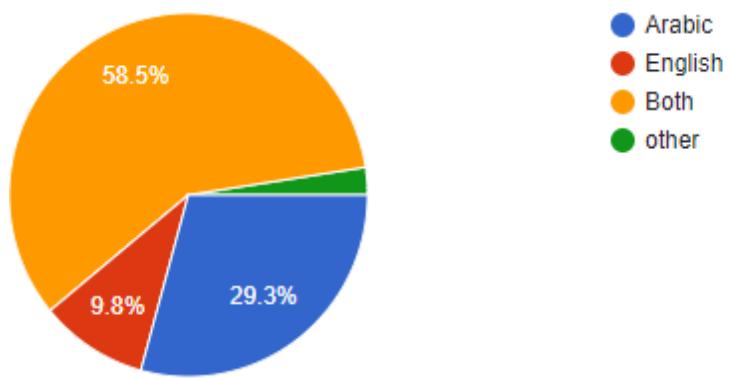
Which one do you think is better real estates offices or real estates ?applications



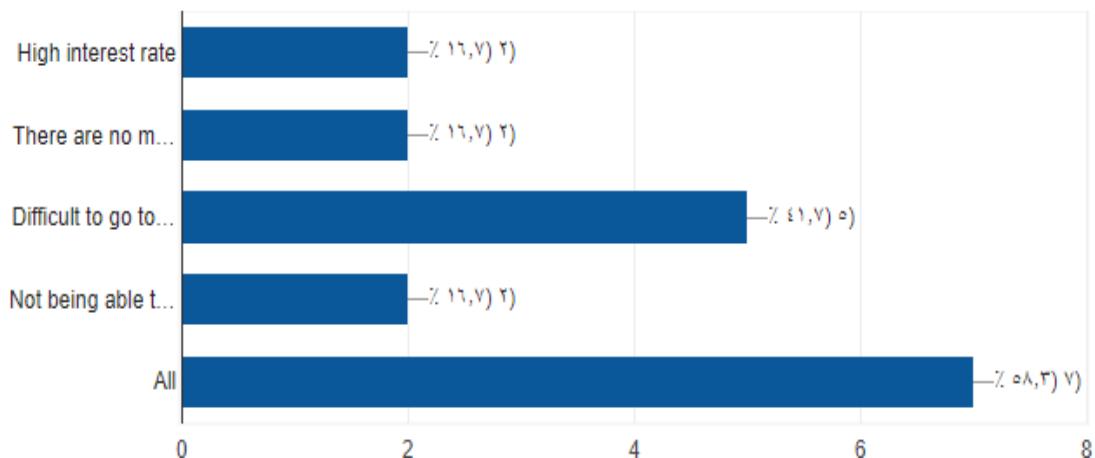
How do you see people's turnout to the real estates electronic ?applications



?Do you prefer Arabic applications or English



?What are the negativities of real estates offices



The result that we get from the Questionnaire:

the results of the survey which the most participated age group was 78% between 20-30 and 9.8 between 30-40, where 75.6% from females and 24.4% from males and 68.3% are users and 12.2% are brokers, 56.1% of the respondents don't have experience on selling and buying estates and 22% have. And 53.7% respondents would like to use an application for real estates ,91.7% agreed that the real estate's applications will help in saving cost, effort and time. 58.3% of respondents are found difficulties in informing or receiving real estate's offers. 75% think that real estate's applications are better than real estate's offices and 25% don't. and 50% amount of people turn out to these applications are average. 58.5% preferred both languages English and Arabic in the real estate applications.

2.1.1.2 Functional and data requirements

2.1.1.2.1 Project stakeholders

- **Administer:** who has the right to enter into the database and update it, which responsible to manipulate tables, receives and view report, put the users who violate the terms and condition of the app in blacklist, send the notification.
- **User:** the user of the system it can be anyone register in the app or just browse it. who are can add estate with pictures and video and delete or update it, search for a particular estate, evaluate the advertiser, communicate with other users (by chats) and share collection & favorite estates with others. Also, the users registered in the app can update their profile if they desire or delete it.

2.1.1.2.2 Functional requirements

The functional requirements determine the processes and tasks which the Administrators and users must perform. For Saudi Estate it will be divided into 2 parts: the users, and the admins.

For the user:

Registration:

Sign up:

If the user wants to create an account, the proposed application will ask the user to enter his information like (first name, last name, username, password, confirm the password , email, phone number (optional)).

Log in:

The proposed application will ask the user to enter his username and password for the account created earlier.

Log out:

The proposed application will allow the user to log out from it and before that it will ensure that the user wants to log out or not.

Search:

The user can search for the estate by:

- his required specifications for the estate by filtering in search bar to enter his specifications such as (price, number of room,etc.).
- write the location in the search bar and he can get benefit from feature 'near me' that enables the user to discover the real estates around them.

Access to estate:

To enhance the term of saving time and easy access to the estate, the user can use travel flow data from Google Map to find the best location to the estate.

Edit profile:

The proposed application allows the users to make changes in their profile and save it anytime. Also, he can delete his profile, but the other functionalities won't be available for him anymore such as adding an estate.

View estates:

The proposed application will allow the user to view recent estates which have been added by the advertiser and view the result of the chosen estate in page contains adequate information and the details of estate description with Multimedia and its location.

Send report:

The proposed application will allow the user to send report such as about the advertiser's estate if he found out it's nothing like what he added in the application or there's a slight flaw in the application.

Share and favorite:

The proposed application will allow the user to make a list of favorite estates and sharing it with others to take their opinions about it.

delete estates:

After the advertiser adds estate in the application he can delete it after selling/renting it or he changed his mind about selling/renting it.

Receive notifications:

The user will receive notifications from application about any updates like new estates added, direct messages .etc.

For the admin:**Registration:****Log in:**

The admin can log in to the system.

Receive report:

The admin receives a report that sent by the user and makes the verifying process by taking estate ID and decide the necessary action.

- If the report terms are complete the admin will send warming message to the user after five warnings the user will addto blacklist.
- If the report terms are incomplete it will be declined.

Send notifications:

The admin will send notifications to the users.

Delete estate:

deleting estates will be made automatically after a period amount of time if the advertisement does not update it.

2.1.1.2.3 Non-Functional requirements***2.1.1.2.3.1 Look and feel requirements:***

The application will be easy and good looking and consistent colors and fonts and clear the data and what data must be entered by marking (*) We try to focus on all important

aspects and interfaces for the user to gain confidence and comfortable reactions.

2.1.1.2.3.2 Usability requirements:

Easy to identify application icons and what to show for those who have little experience in mobile applications.

2.1.1.2.3.3 Access control:

The application prohibits access to and manipulation of personal data of users.

And no one can hack the database or change the data about users for example advertisement information.

2.1.1.2.3.4 Performance requirements:

The application will be interactive and has Fast responses for different interactions.

2.1.1.2.3.5 Portability requirements:

The system determines the ability to install the software on all Android devices because the application will work on this system and can download it from the Play Store.

2.1.1.2.3.6 Accessibility:

Anyone can install, access the application and use its features as they like.

2.1.1.2.3.7 Availability:

The application will be free, the customer will be enable to download it and install it.

2.1.1.2.3.8 Privacy requirements:

The privacy policy for data entered into the application will protect this private data that will be stored in the database and allow only the developer to update and delete it.

2.1.2 Data flow diagrams

Context diagrams:

The Context Diagram shows the system under consideration as a single high-level process and then shows the relationship that the system has with other external entities (systems, organizational groups, external data stores, etc.). Context diagram in figure 3.

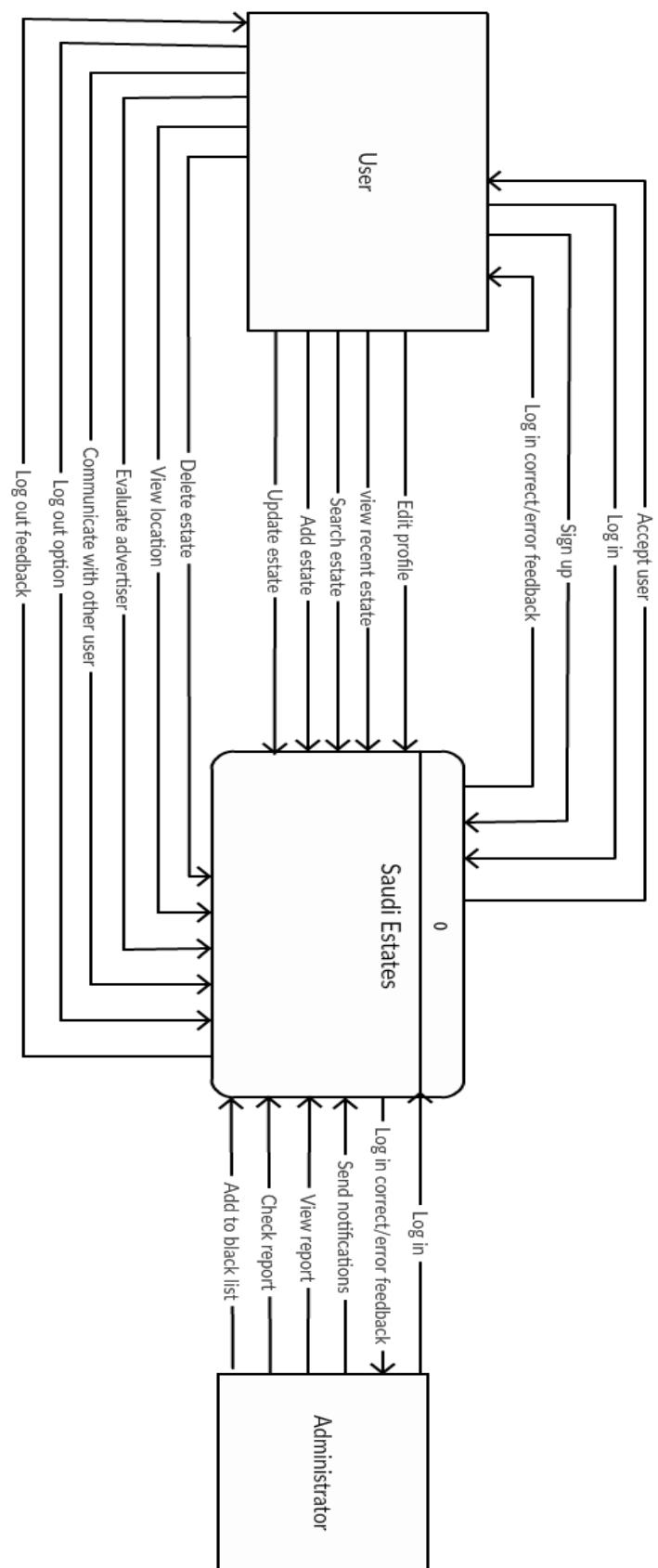


Figure 3: Context Diagram of DFD

Level 0 diagram:

The movement of data between external entities and the processes and the data stores within a system. We divided it into two diagrams one in figure 4 and other in figure 5 to be clearer to any other programmer to analyze.

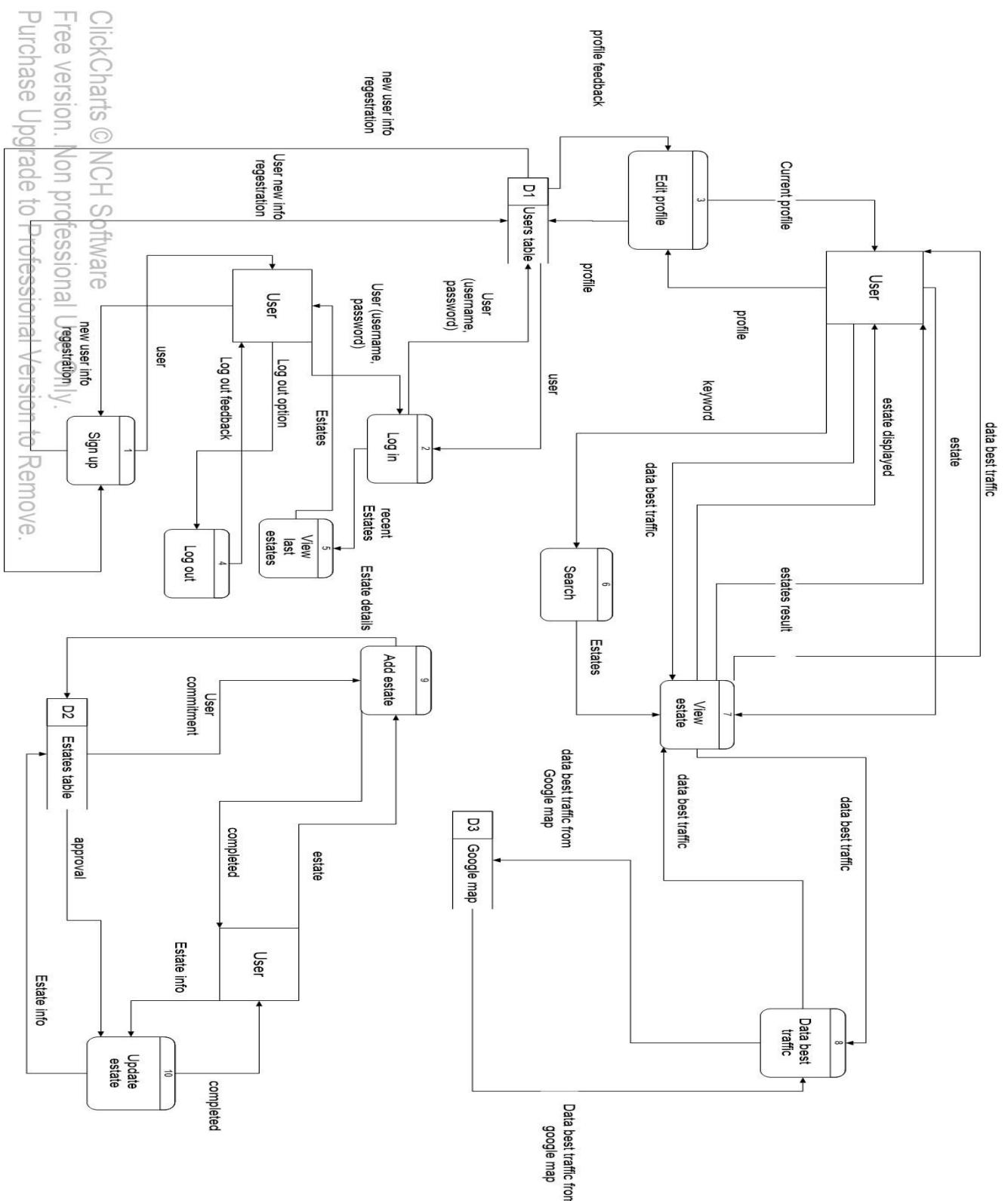


Figure 4: Level 0 of DFD

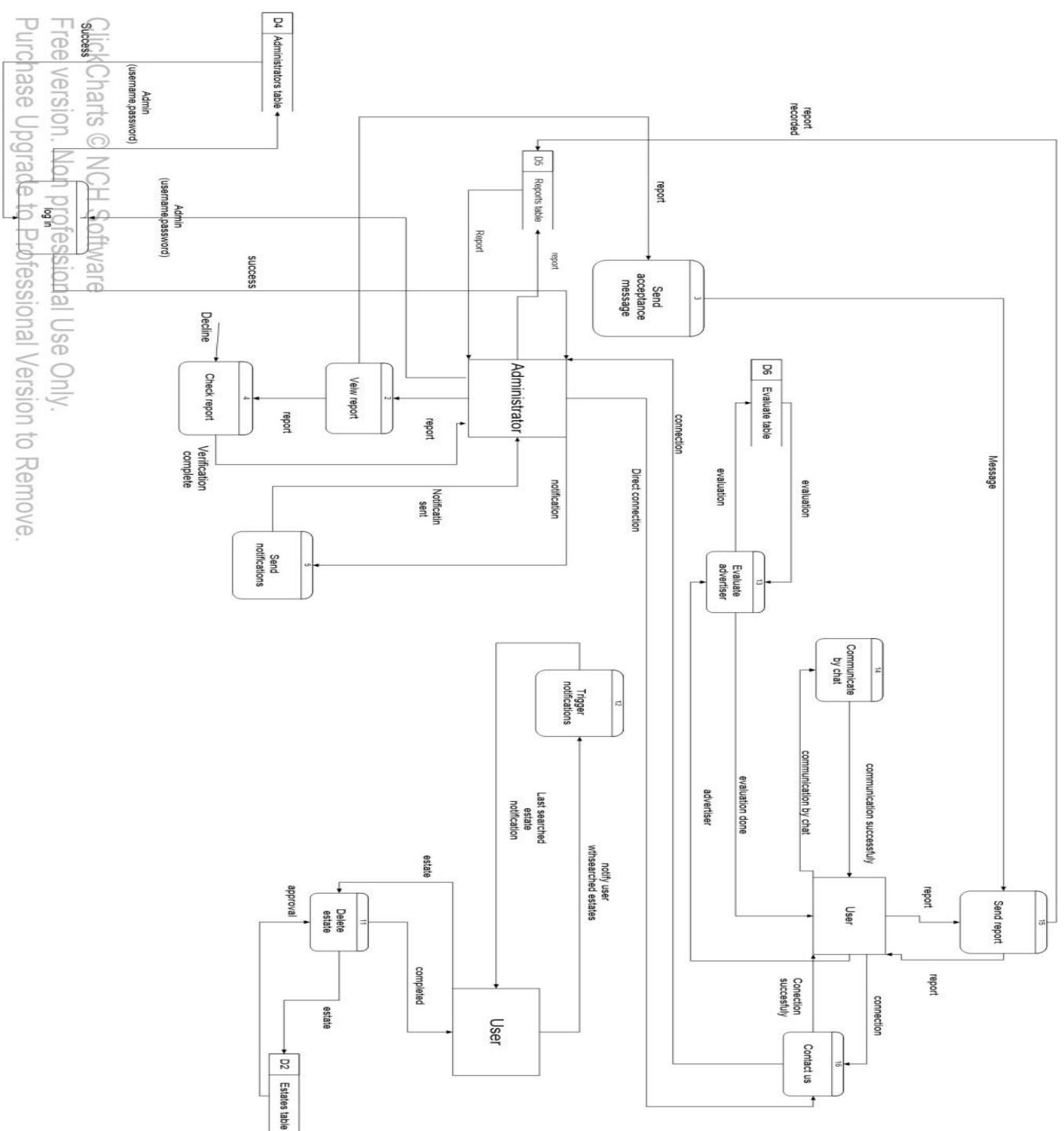


Figure 5: Level 0 of DFD

Child diagram:

The process 4 and process 6 on Diagram 0 (user entity) turned be exploded to create a more detailed child diagram. both processes 4 and 6 that is exploded is called the parent process, and the diagram that results is called the child diagram represent in figure 6 and figure 7.

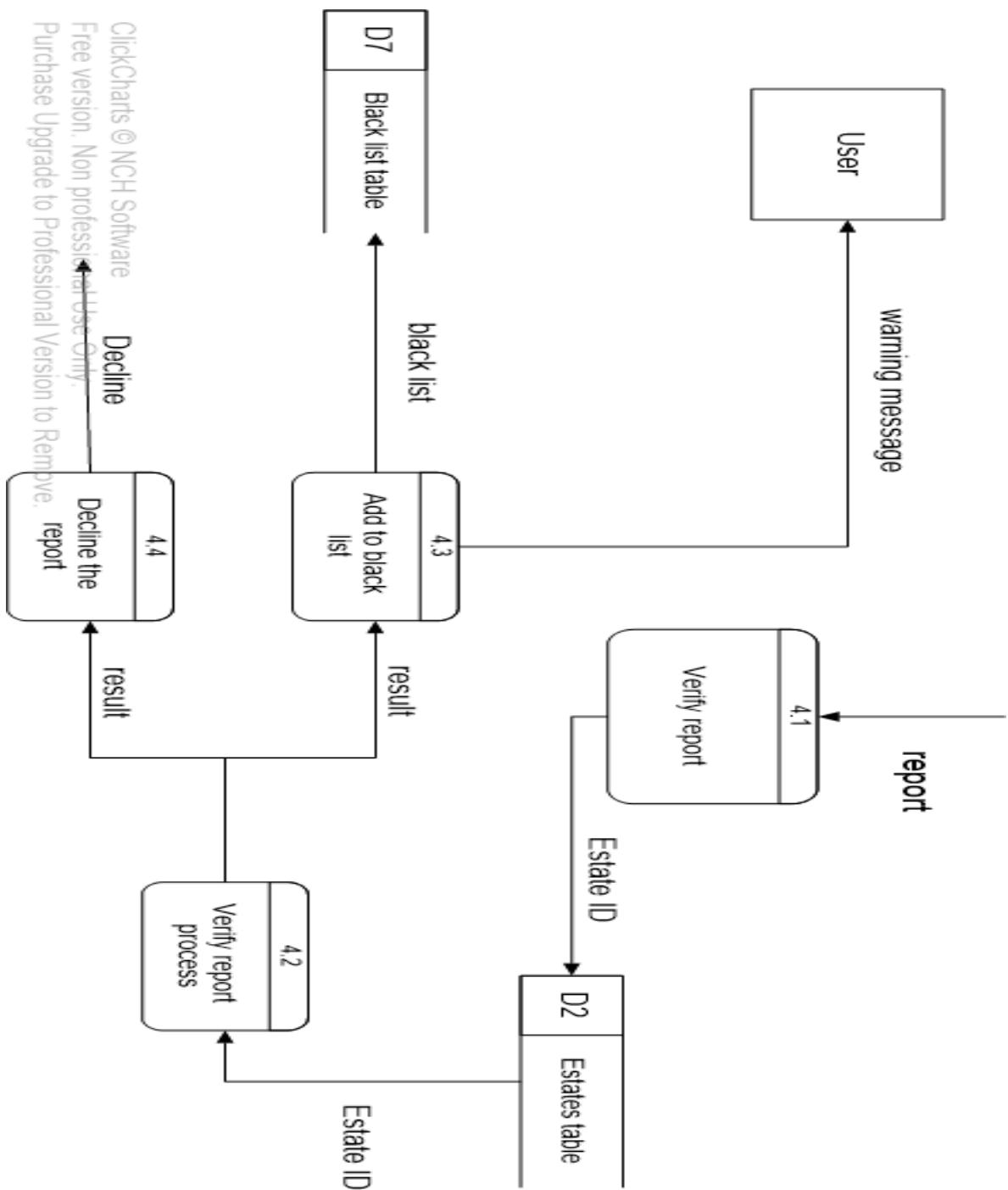


Figure 6: Child Diagram for process 4 in level 0(user entity)

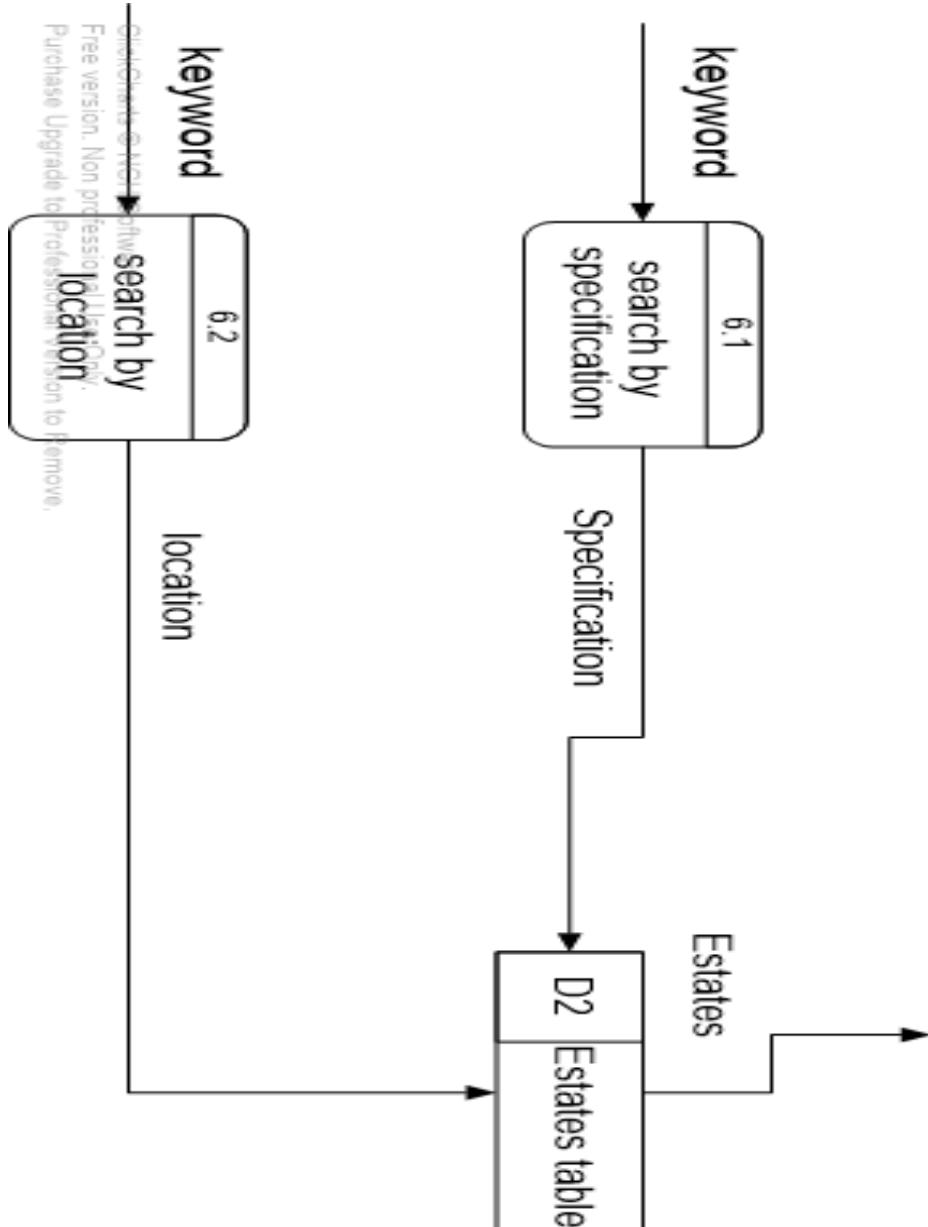


Figure 7: Child Diagram for process 6 in level 0(user entity)

2.1.3 Unified Modeling Language Diagrams

UML stands for Unified Modeling Language that can be used to model an application structures, behavior ,and even business processes. We develop a number of models in order to explain our system and its behavior.

Use Case Diagram(UCD):

Use case was developed originally to support requirement elicitation and the main purpose of UCD is to help development teams visualize the functional requirement of a system. We will show it in figure 8.

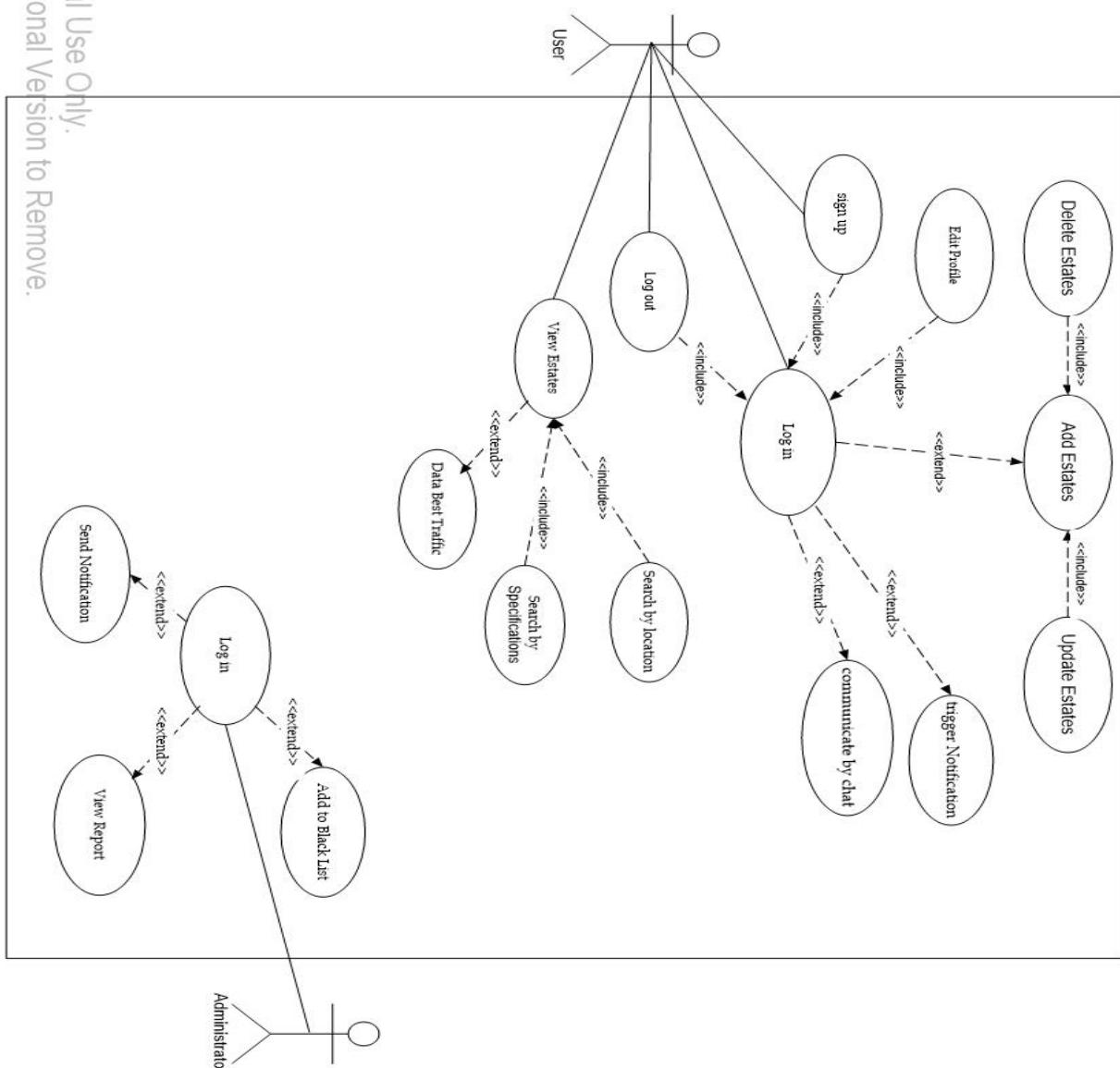


Figure8: use case diagram.

Application Use Case Diagram Details:

Use case name	Log in.
Use case name	Log in.
Actors	User.
Triggering event	User requests to log in.
Preconditions	User must have an account.
Flow of activities	1- User clicks on log in button. 2- User will enter his username and password. 3- Application will verify the information. 4- User will be logged in application.

Table 2: log in

Use case name	Log out.
Actors	User.
Triggering event	User requests to log out.
Preconditions	User must be logged in.
Flow of activities	1- User clicks on log out button. 2- User confirm the log out 3- User will be logged out of application.

Table 3: log out

Use case name	Delete Estates.
Actors	User.
Triggering event	User requests to delete Estates.
Preconditions	User must be logged in and has added estates.
Flow of activities	1- User view his estates. 2- User clicks on delete button. 3- User confirm the delete. 4-the application will delete the estate.

Table 4: Delete Estates

Use case name	Sing up.
Actors	User.
Triggering event	User requests to create an account.
Preconditions	The user must enter the required Information.
Flow of activities	1- User clicks on sing up button. 2- user will enter his information's. 3- Application will verify the information. 4- User will be have an account in application.

Table 5: sign up

Use case name	Add Estates.
Actors	User.
Triggering event	User requests to Add Estates.
Preconditions	User must be logged in.
Flow of activities	1- User clicks on Add Estates button. 2- User will enter all information about the estate. 3- User confirm the sell. 4- the application will Add the estate.

Table 6: Add Estates.

Use case name	Update Estates.
Actors	User.
Triggering event	User requests to update Estates.
Preconditions	User must be logged in and has added estates.
Flow of activities	1- User view his estates. 2- User clicks on update button. 3- User confirm the update. 4-the application will update the estate.

Table 7: Update Estates.

Use case name	Data Best Traffic.
Actors	User.
Triggering event	When the user wants to see the shortest way to estates he chooses.
Preconditions	User must click view estates.
Flow of activities	1- User click on view estate. 2- The application will view the estate Information and details. 3- User click on best traffic button. 4- The application will open Google map and view the shortest way.

Table 8: Data Best Traffic.

Use case name	Edit profile.
Actors	User.
Triggering event	User requests to edit his information in his account.
Preconditions	User must be logged in.
Flow of activities	1- User clicks on profile button. 2- User clicks on Edit profile button. 3- user will enter his new information's. 4- Application will confirm and change the user information. 5- User will be edited his profile.

Table 9: Edit Profile.

Use case name	Log in.
Actors	Administrator.
Triggering event	Administrator requests to log in.
Preconditions	Administrator must have an account.
Flow of activities	1- Administrator clicks on log in button. 2- Administrator will enter his username and password. 3- Application will verify the information. 4- Administrator will be logged in application.

Table 10: log in for Admin.

Use case name	Communicate by chat.
Actors	User.
Triggering event	User requests to communicate with another user.
Preconditions	User must be logged in.
Flow of activities	1- User clicks on chat button in another user profile. 2-The application will open his chat. 3-user will can communicate with this user.

Table 11: Communicate by chat.

Use case name	Location.
Actors	User.
Triggering event	User requests search in the application by the location or Near me feature.
Preconditions	User clicks the search button.
Flow of activities	1- User clicks on search button. 2- The application will open the ways of search. 3- User will clicks location to search by location. 4- Application will open Google map, if user want to search by estates near him, user will click on Near me button. 5- Application will search in database and view the results.

Table 12: Location

Use case name	View Report.
Actors	Administrator
Triggering event	Administrator want to see reports send from the users.
Preconditions	Administrator must be logged in and received the report.
Flow of activities	1- Administrator click on received report. 2- The application will view the list of report.

Table 13: View Report.

Use case name	Specifications Estate.
Actors	User.
Triggering event	User requests search in the application by specifications in estates he want it.
Preconditions	User clicks the search button.
Flow of activities	1- User clicks on search button. 2- The application will open the ways of search. 3- User will enter the specifications he wants it. 4- User clicks on ok button. 5- Application will search in database and view the results Depending this specification.

Table 14: Specifications Estate.

Use case name	View Estate.
Actors	User.
Triggering event	When the user wants to see the estates details.
Preconditions	User clicks the estates.
Flow of activities	1- User clicks on the estate. 2- The application will view the estate Information and details.

Table 15: View Estate.

Use case name	Send Report.
Actors	User.
Triggering event	User send a report for the administrator.
Preconditions	see something wrong in estate advertising.
Flow of activities	1-User see our information email, phone number to send the report.

Table 16: Send Report.

Use case name	Add to Black list.
Actors	Administrator.
Triggering event	The administrator add the user in black list.
Preconditions	Administrator must be logged in
Flow of activities	1-Administrator receive the user report. 2-Administrator checks if it true. 3- Administrator will send warning message offending user. 4- Administrator will add the user to the black list after 5 warning, the user can't add advertising again.

Table 17: Add to Black list.

Use case name	Send notification.
Actors	Administrator.
Triggering event	The administrator send notification.
Preconditions	Administrator must be logged in
Flow of activities	1- Administrator receive the user request. 2- Administrator will send notification to him.

Table 18: Send notification.

Use case name	Trigger notification.
Actors	User.
Triggering event	User requests to send to him a notification when new estates has added have same specifications he search about.
Preconditions	User must be logged in.
Flow of activities	1- User clicks on Trigger notification button in his profile. 2- user will enter the specifications he searches about. 3- Application will verify the information. 4- Application will send notification if found estates with same specifications.

Table19: Trigger notification.

Activity Diagram:

First user can sign up or sign in (if he has an account), also he can search for estates without sign in or sign up but he will not be able to use all user specifications like (offer his estates for sell or rent, make collections, make favorite list, share his favorite and collections, evaluate estates, turn on notifications, chats with other users, send reports).

After that user can search for estates and if he did not find what he wants then he can sign in if he doesn't then turn on notifications, and if he found an estate he can show information, see it on map, show best traffic data and calculate distance, and if this estate is suitable for him then he can contact with advertiser (the user who offers his estate for rent or sell). if the deal is done then the estate will be deleted and the user can evaluate the advertiser. if the deal is not done then also the user can evaluate the advertiser and if there is a problem like (The advertiser is not honest and put the wrong information's) then the user can send a report, then the advertiser will be put into a blacklist (we will count until 5 times he puts there then we will prevent him from offering any other announcement).also user after sign in he can offer his estate for sell or rent (after 1 month he must update his estate to make us know that it is still available and if the advertiser didn't update his estate then it will be deleted automatically), the user can also make his collections and share it with other users, make a favorite list which other users can see his likes under the estate and contact us. Activity Diagram is shown in figure9.

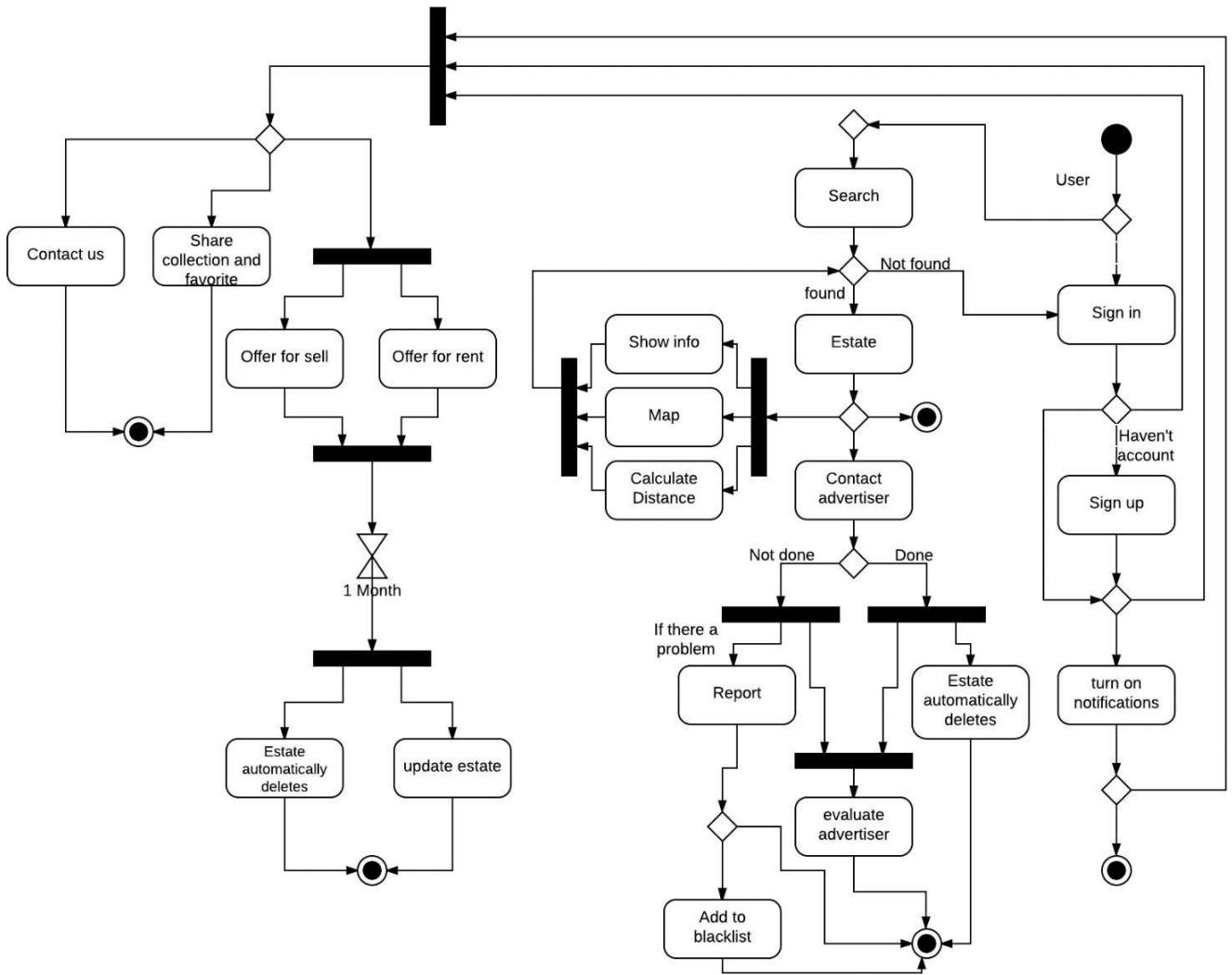


Figure 9: Activity Diagram

ERD Diagram:

Is a semantic data model, used for the graphical representation of conceptual database design, the ERD for our proposed system in figure 10.

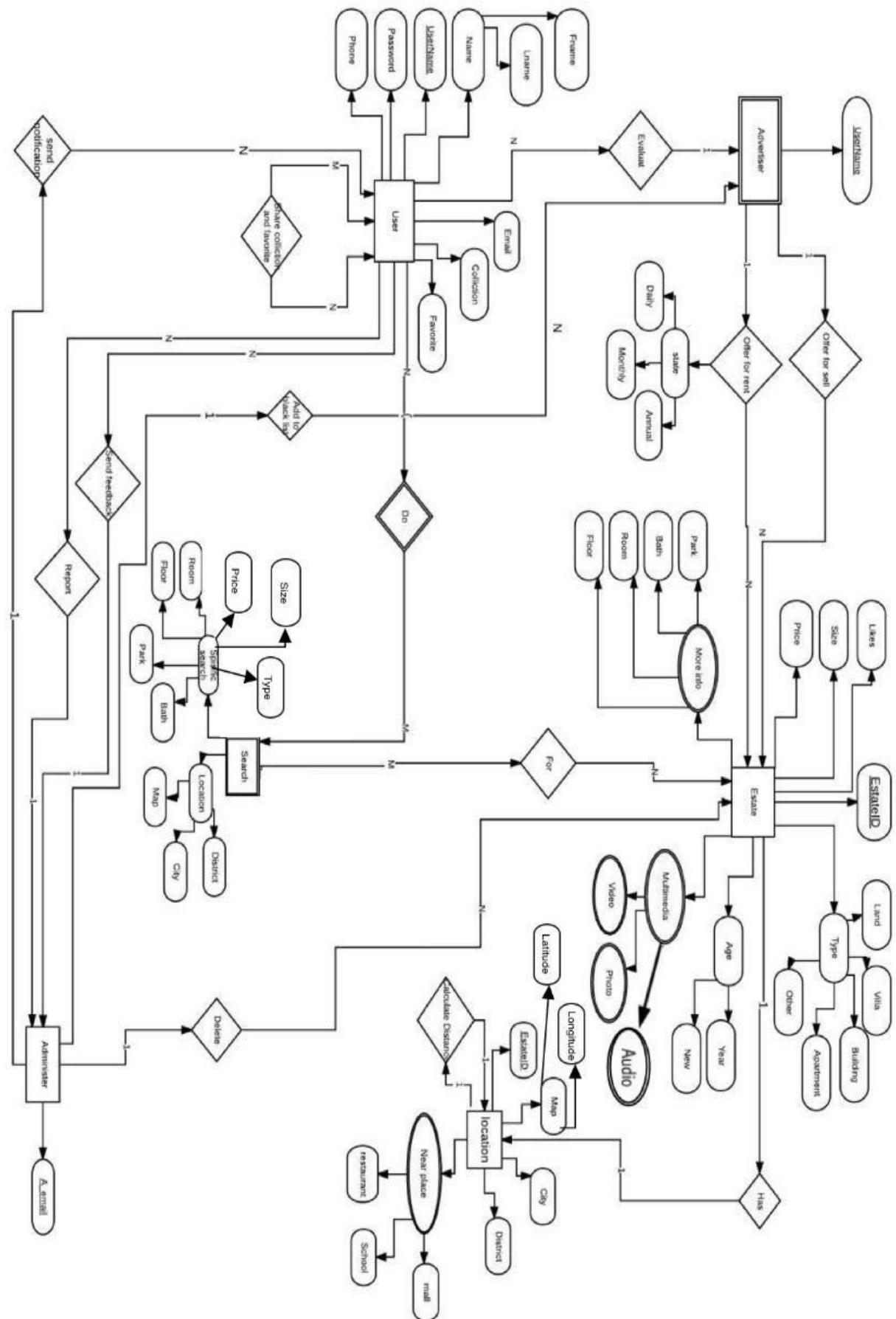


Figure 10: ERD Diagram

Sequence Diagram:

used to show how the objects interact with each other in the system with specifying the order those interactions occur. The Sequence diagram in figure 11 and 12.

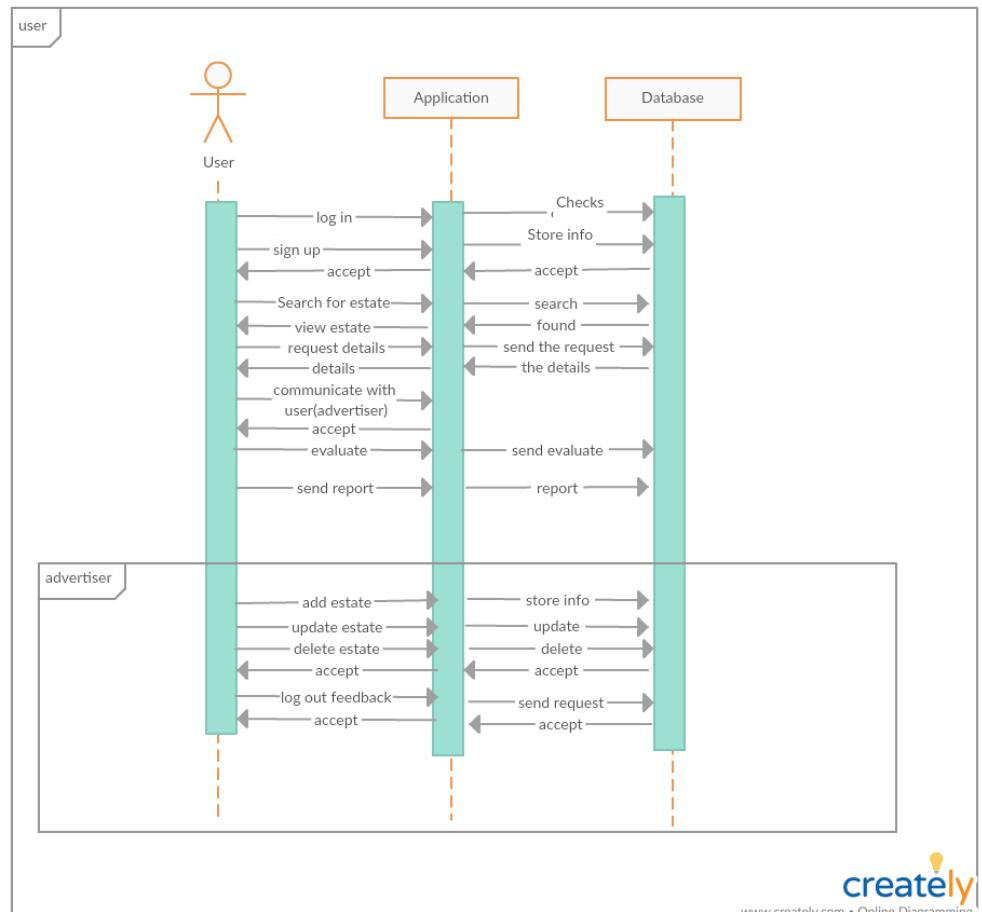


Figure 11: User Sequence Diagram

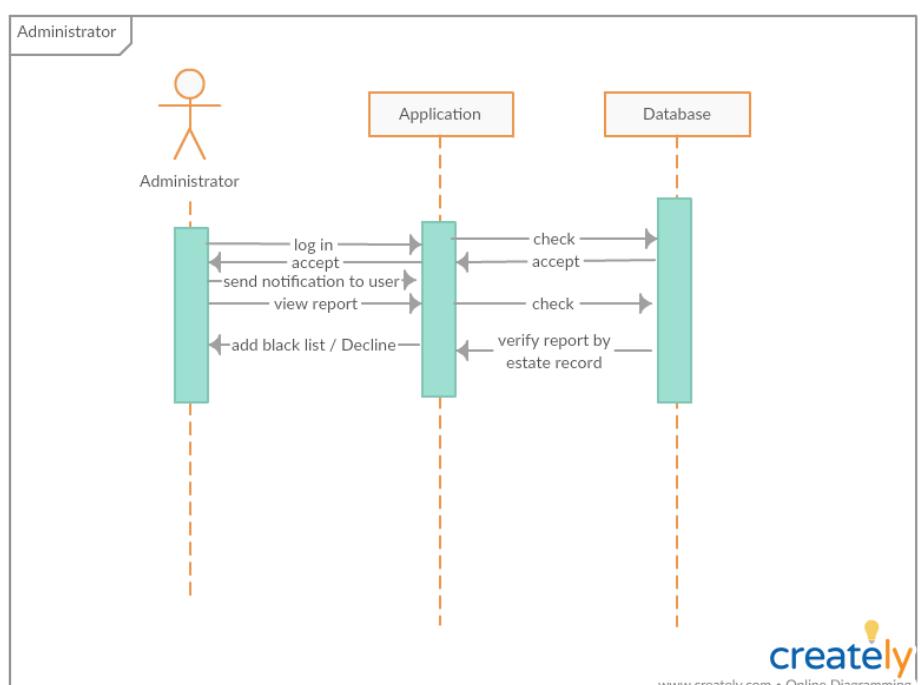


Figure 12: Admin Sequence Diagram

Class Diagram:

The class diagram in the UML is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. The class diagram for our propose will present in Figure13.

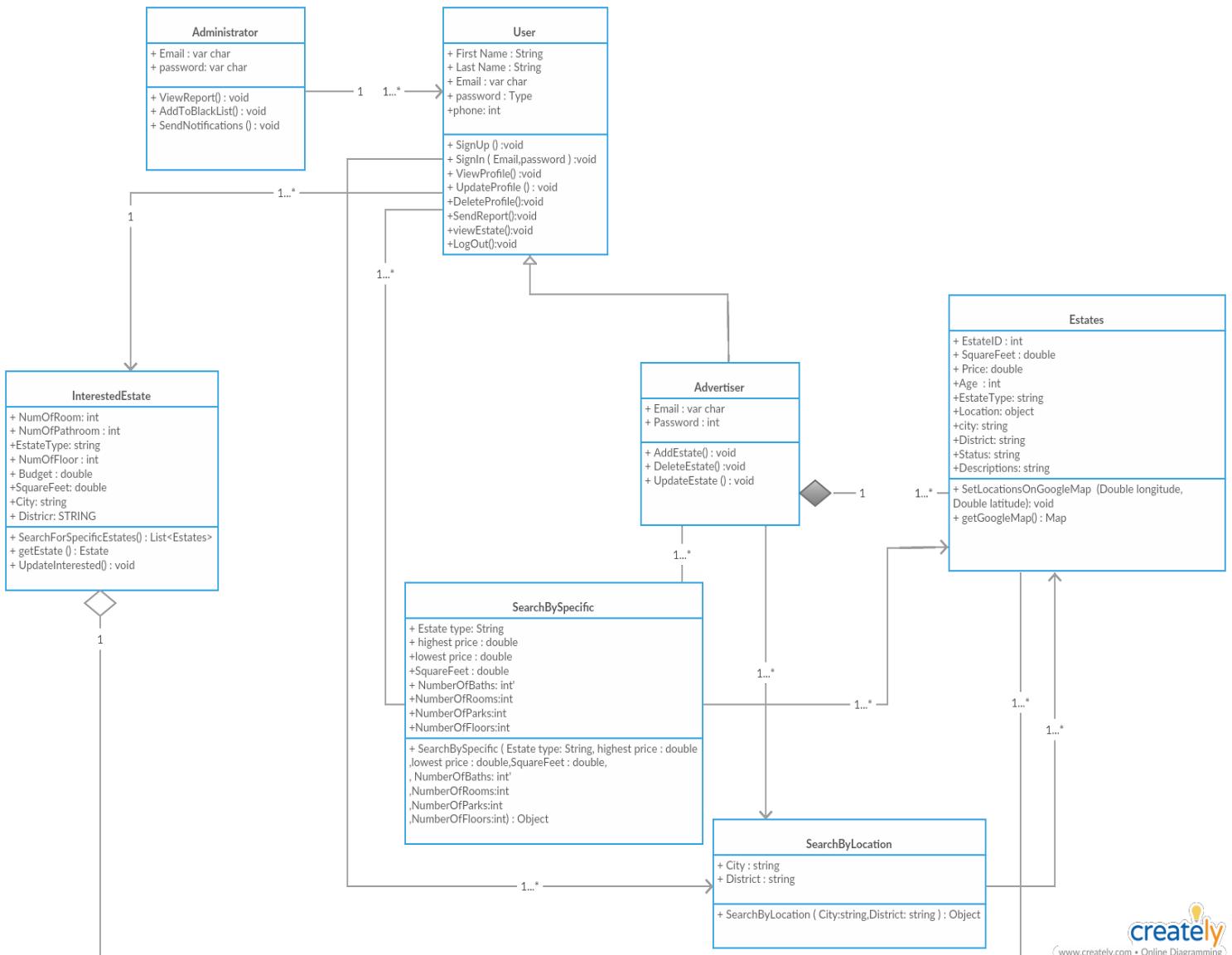


Figure 13: Class Diagram

2.1.4 Proposed Solutions and Alternative Solutions

Proposed Application:

The proposed application, Saudi Estates, is useful and helpful because it will be an appliance that will simplify to the users to browse the estates in more efficient way and provides improvement in the business environment to be better, as we mentioned earlier in chapter 1 the existing systems that specialize in the same area of our project, we noticed they lack some several features and services like doesn't expand to a lot of areas and don't clarify selling process, conversely we will make our best to offer many services and functionalities that would be as convenient as the user wants such as travel flow data on maps to enhance finding locations of estates easily. therefore, saves time and effort. Moreover, rapidly access to the contacts of an owner and schedule an appointment with advertiser to complete the purchase process, and provide estates from around the kingdom and more described in Chapter 1.

Alternative solution:

- The user can either advertise or buy or rent an estate according to his needs.
- The user capable of using travel flow data on maps to find the best location in term of time rather than the advertiser location description.
- Search according to the specification or the last estates added will be displayed in his timeline.



CHAPTER 3:

DESIGN CONSIDERATION

3.1 Database Constraints

3.1.1 Database Schema

3.2 Design Constraints

3.2.1 Hardware and Software environment

3.2.2 End user characteristics

3.3 Architectural Strategies

3.3.1 Algorithm to be used

3.3.2 Project management strategies

3.3.3 Development method

3.3.4 Future enhancements/plans



3.1 DATABASE CONSTRAINTS

This section describes the design of the conceptual model of the database.

3.1.1 Database Schema:

Firebase database is a schema-less database in which the data is stored in JSON format.

Basically, the entire database is a big JSON tree with multiple nodes.

saudi-estates

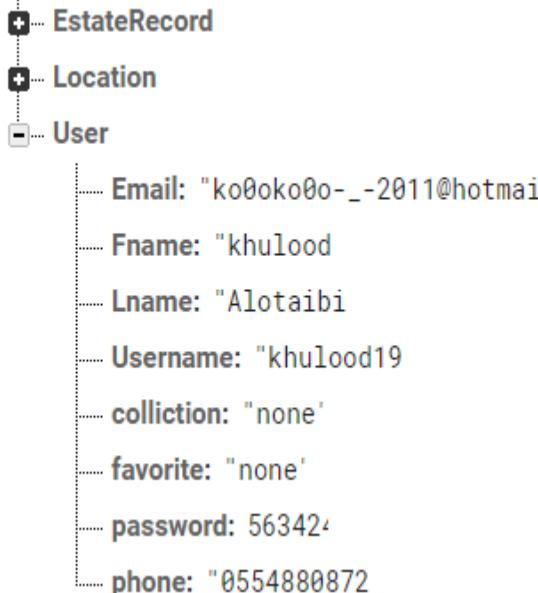


Figure 14: show user database

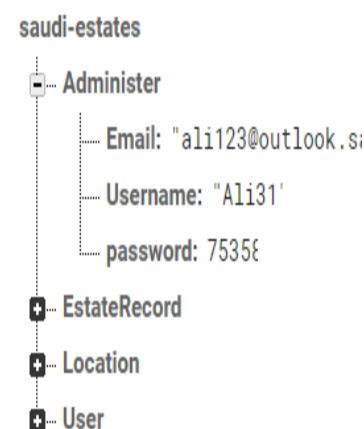


Figure 15: show Administer database

saudi-estates

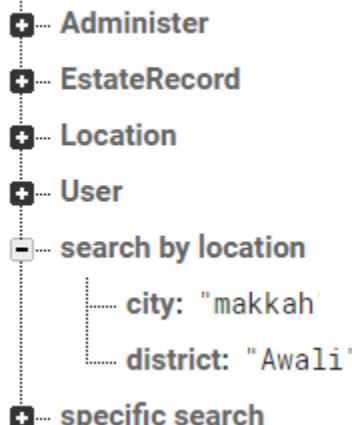


Figure 16: show search by location database

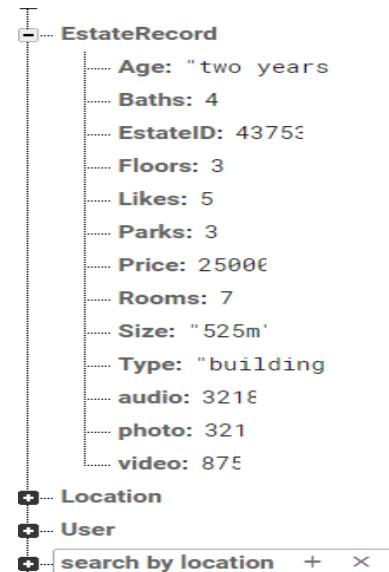


Figure 17: show estate database



Figure 18: show specific search database

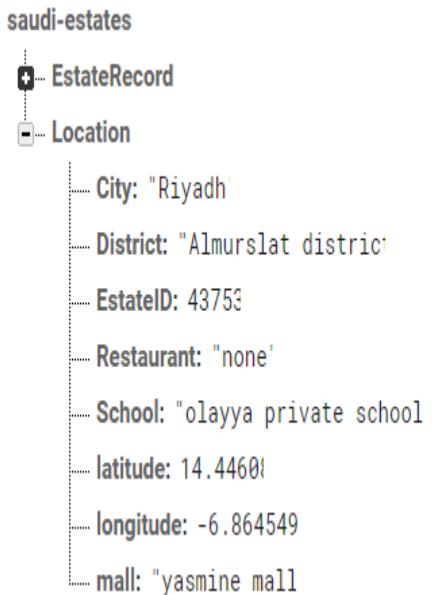


Figure 19: show location database

Name	Type	Description
Email	Varchar	The Admin's email.
Username	Varchar	The User Name of admin which will use it to access the application.
Password	Varchar	The password of account.

Table 20: show admin database details.

Name	Type	Description
City	String	The users can search for estate in a particular city.
District	String	The users can search for estate in a particular city, also in a certain district.

Table 21: show search by location database details.

Name	Type	Description
Baths	Int	The number of bath in estate.
Floors	Int	The number of floors in estate.
Parks	Int	The number of parks in estate.
Price	Int	The price of estate
Size	Double	The size of Estate.
Rooms	Int	The number of room in estate.
Type	String	type of estate (building, villa, land etc.).

Table 22: show specific search database details.

Name	Type	Description
City	String	The city and district where the estate located
District	String	
EstateID	Int	The ID of estate that assigned to this location.
latitude	Double	The coordinates of location
longitude	Double	
Restaurant	String	Public services close to the estate.
School	String	
Mall	String	

Table 23: show location database details.

Name	Type	Description
Age	Int	When the user post estate must determine the age of estate.
Baths	Int	The number of bath in estate.
EstateID	Int	The ID of estate.
Floors	Int	The number of floors in estate.
Parks	Int	The number of parks in estate.
Price	Int	The price of estate
Size	Double	The size of Estate.
Rooms	Int	The number of room in estate.
Type	String	The user must determine the type of estate (building, villa, land etc.).
Like	Int	The total number of likes for estate.
Photo	Int	The ID of photo, audio and video.
audio	Int	
video	Int	

Table 24: show estate database details.

Name	Type	Description
Email	Varchar	The user's email.
Username	Varchar	The UserName of user which will use it to access the application.
Password	Varchar	The password of account.
Fname	String	First Name of users.
Lname	String	Last name of users.
Collection	Int	The number of collection in user account, the default value none.
Favorite	Int	The number of favorite in user account, the default value none.
Phone	Int	The phone number of users.

Table 25: show users database details.

3.2 DESIGN CONSTRAINTS

In this part, we define the hardware and software that we will use to start our proposed system and some of them has been used currently. In addition, characteristics of end users.

3.2.1 Hardware and software environment

- **Hardware environment:**

The application will run on smart phone support Android.

- **Software environment:**

To develop an Android Application, we will use:

- ***Android Studio:***

Android Studio is Android's official IDE. It is purpose built for Android to accelerate your development and help you build the highest-quality apps for every Android device. It offers tools custom-tailored for Android developers, including rich code editing, debugging, testing, and profiling tools.

- ***Firebase:***

Firebase is a mobile platform that helps you quickly develop high-quality apps, it offers a number of different services built-in, including some basic analytics.

Why we choose the firebase?

Firebase frees developers to focus crafting fantastic user experiences. You don't need to manage servers. You don't need to write APIs. Firebase is your server, your API and your data store, all written so generically that you can modify it to suit most needs.

Services gets from firebase:

- Real time database: Firebase provides a free database.
- Works on different operating systems and platforms (Android(Java), iOS(C#), Web(JavaScript)).

- File Storage.
- Less effort for the coder to get synchronized.
- Easy and enjoyable documentation.
- Authentication: Firebase authentication has a built-in email/password authentication system.
- User interaction statistics during the day [9].

Firebase analytics features: [10]



Figure 20: firebase analytics

How everything will work:

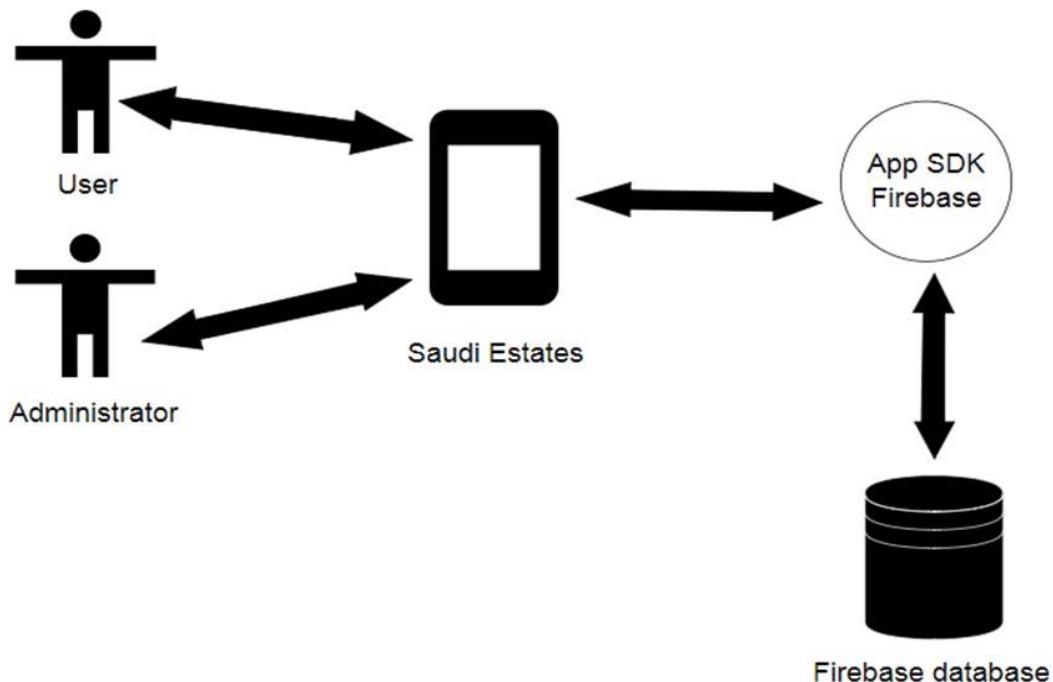


Figure 21: How Everything work

Other Software:

- ***ClickCharts***, ***Flowchar*** software, ***creately*** and ***Lucidchart*** website:

It used for drawing flowcharts such as data flow diagram, Class diagram, Sequence diagram, use cases.. etc..

- ***JustInMind***:

We will use this tool in chapter four to design the prototype of application.

Also, we used Adobe Photoshop CS.

3.2.2 End user characteristics

Our proposed system is used by:

- User:

The users can use this system to search for estates in the kingdom of Saudi Arabia. In addition, the users can Sign-up in our app to benefit from all its services. We display the estates as a list if the user clicks on it, they will

get more information about the estates. And about the search properties, we will provide two ways to search: First by determining specifications estates that he wishes and can also determine the services close to an estate like hospitals, schools, supermarket, and others. Second, search by location by typing the name of the city in the search bar and we will provide property "near me" by this property the user can search for available estate around him. The users can delete, update and add their estates and they can active trigger notifications to send to him notification about newly added estates that conform to predefined specifications .as well, the user can communicate, and share collection & favorite estates with others, and send reports to the admin.

- Admin:

All admin most Sign-in in our app they can do many tasks such as: receive and view reports, send notifications about estates and new in our app to users and others.

3.3 ARCHITECTURAL STRATEGIES

Architecture strategy translates business strategy into objectives for building, enhancing, or replacing business and system capabilities together with an implementation roadmap, all-the-while maintaining resilience to change as a key architectural objective.

3.3.1 Algorithm to be used

Algorithm find pareto front

In this approach, each solution t is compared with every other solution in the population to check if it is dominated by any solution in the population. If the solution i . is found to be dominated by any solution, this means that there exists at least one solution in the population which is better than i in all objectives. Hence the solution cannot belong to the non-dominated set. We mark a flag against the solution i to denote that it does not belong to the non-dominated set. However, if no solution is found to dominate solution i , it is a member of the non-dominated set. This is how any other solution in. the population can be checked to see if it belongs to the non-dominated set. The following approach describes a step-by-step procedure for finding the non-dominated set in a given set P of size N .

Identifying the Non-Dominated Set: Approach 1

Step 1 Set solution counter $i = I$ and create an empty non-dominated set P' .

Step 2 For a solution. $j \in P$ (but $j \neq i$), check if solution j dominates solution i . If yes, go to Step 4.

Step 3 If more solutions are left in P, increment j by one and go to Step 2; otherwise, set $P' = F' \cup \{i\}$.

Step 4 Increment i by one. If $i < N$, go to Step 2; otherwise stop and declare P' as the non-dominated set.

Example:

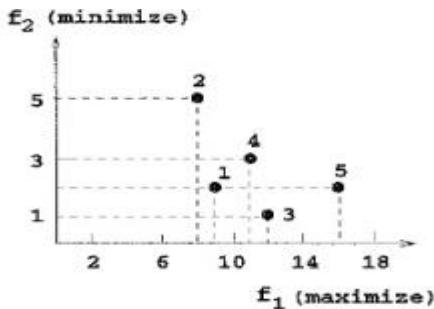


Figure 22: A population of five solutions.

We illustrate the working principle of the above procedure on the same set of five ($N = 5$) solutions, as shown in Figure 14. Ideally, the exact objective vector for each solution will be used in executing the above procedure, but here we use the figure to compare different solutions. We follow the above step-by-step procedure in the following.

Step 1 We set $I = 1$ and $= 0$.

Step 2 We compare solution 1 with all other solutions for domination, starting from solution 2. We observe that solution 2 does not dominate solution 1.

Step 3 However, solution 3 dominates solution 1. Thus, we move to Step 4.

Step 4 Solution 1 does not belong to the non-dominated set and we increment i to 2 and move to Step 2 to check the fate of solution 2.

Step 2 We observe that solution 1 dominates solution 2. We therefore move to Step 4.

Step 4 Thus, solution 2 does not belong to the non-dominated set. Next, we check solution 3.

Step 2 and 3 Starting from solution 1, we observe that neither solution 1 nor 2 dominates solution 3. In fact, solutions 4 and 5 also do not dominate solution 3. Thus, we include solution 3 in the non-dominated set, $P' = \{3\}$.

Step 4 We now check solution 4.

Step 2 Solution 5 dominates solution 4. Thus, it cannot be a member of P' . Step 4 Now we check the final solution (solution 5).

Step 2 We observe that none of the solutions (1 to 4) dominates solution 5.

Step 3 So, solution 5 also belongs to the non-dominated set. Thus, we update $P' = \{3, 5\}$.

Step 4 We have now considered all five solutions and found the non-dominated set = $\{3, 5\}$.

3.3.2 Project management strategies

Project management strategies are the discipline of using established principles, procedures and policies to manage a project from conception through completion. So, in this part, a good management and motivation among different task in the project will be presented.



Initiating:

- in this phase we discussing different ideas with the supervisor and choosing one of them.
- writing a complete project proposal.



Planning:

is the most crucial step in creating a successful system, during this phase we decide exactly what we want to do and the problems we are trying to solve, by;

- Studying the proposed idea with defining the problems and our objectives.
- writing the literature review.
- Studying the advantage and disadvantage of all previous systems.
- Writing the general system idea with all functionality that we want to implement in proposed idea.



Analysis

- Defining different system requirements such as functional and non-functional requirements.
- Studying and Drawing system models.
- Writing proposed and alternative solutions.



Design

- Defining the software and the hardware environment.
- Defining development method.

3.3.3 Development method

As we said before, we will use waterfall model as progress flows in largely one direction ("downwards" like a waterfall) through the phases of conception, initiation, analysis, design, construction, testing, deployment and maintenance.

Advantages of waterfall model:

- This model is simple and easy to understand and use.
- It is easy to manage due to the rigidity of the model, each phase has specific deliverables and a review process.
- In this model phases are processed and completed one at a time. Phases do not overlap.
- Waterfall model works well for smaller projects where requirements are very well understood.

3.3.4 Future enhancements/plans

In future enhancements of the application, we plan to make the application available for the Apple platform to target more user. Activation Virtual Reality technique or 3D view in the application to make user see the estates more clearly and better from pictures. The ability to complete the buying and selling process through the application and make it trustworthy.



CHAPTER 4: SYSTEM DESIGN

4.1 System Architecture and Program Flow

4.1.1 Major modules

4.1.2 Sub modules

4.2 Detailed System Design

4.2.1 Interface description



4.1 System Architecture and Program Flow

In this chapter we will describe the main components of application. In addition, display some interfaces of application with an explanation.

4.1.1 Major modules

Model	Description
User	<p>The user can log in to the application if he wants to search for real estates, either near him or by searching the place, number of rooms, bathrooms ... etc.</p> <p>The user can advertise the real estate by logging in and adding the necessary information to it.</p> <p>He can share and favorite estates. Plus sending reports to the admin.</p>
Admin	Admin takes consideration user's reports to put the real estate blacklisted if it doesn't follow the terms. sends a notification to the user, also delete the real estate automatically that has not been updated.
Database	Database considered an important part of our system because we rely on it in all our operational functions include users, admins registration, estates records, reports, rating record, Google map, blacklist record.
Application	the users can have the application downloaded on their own devices (android), and then optionally can sign up.

Table 26: Major modules.

4.1.2 Sub modules

The system has many sub modules that work together to perform the required functionality.

Model	Description
Log in \ sign up	the user can sign up and the information registered will be stored in the database, then, user can just log in to the system by entering the username and password
Adding an Estate with the necessary information	The user (Advertiser) enters all information about his Estate and submits the information to be stored it in the database.
The possibility of buying\renting an estate	The user can buy\rent the real estate, therefore, communicate with advertiser to complete the purchase process.
Search	Users enter to our application (not necessarily login to the application).and search by location or by the specification.
Notification	Inform the user of real estate.
Sending\receiving reports	Users send a report, then admin verifies the report by Estate ID and decides whether it will be declined or blacklisted.

Table 27: submodules.

4.2 Detailed System Design

4.2.1 Interface description

Interfaces of the welcome page:

This is the first interface in the Saudi Estates when the application is opened. We designed it to welcome the user by simply showing only the application name and logo then a welcome page showing buttons to enter the user login button or sign up.



Figure 23: Interface of the welcome page.

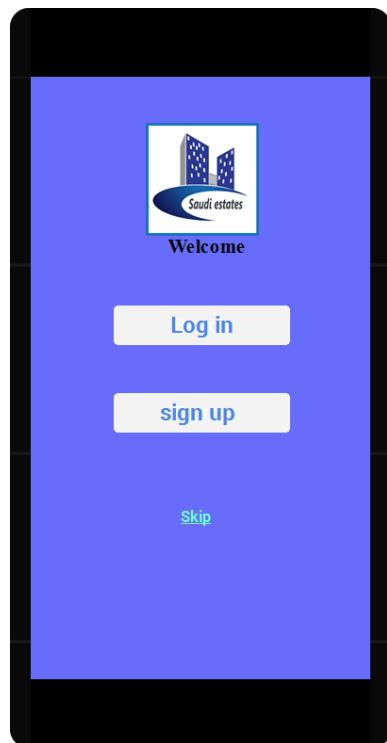


Figure 24: Interface to enter the user.

Interfaces of Login forms:

The form shown will appear to the user. If the user wants to log in through fill out the form and then pressing the log in button.

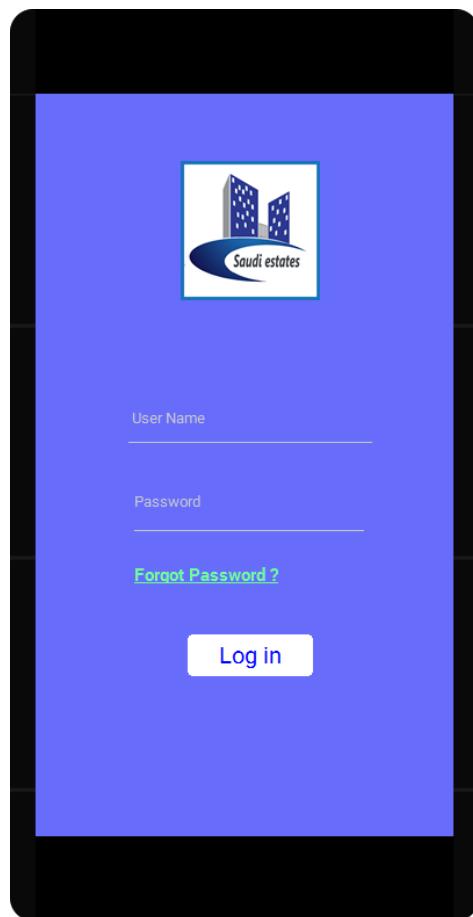


Figure 25: Interfaces of Login form.

Interfaces of sign up forms:

The form shown will appear to the user. If the user wants to sign up for the application through fill out the form and then pressing the sign-up button.

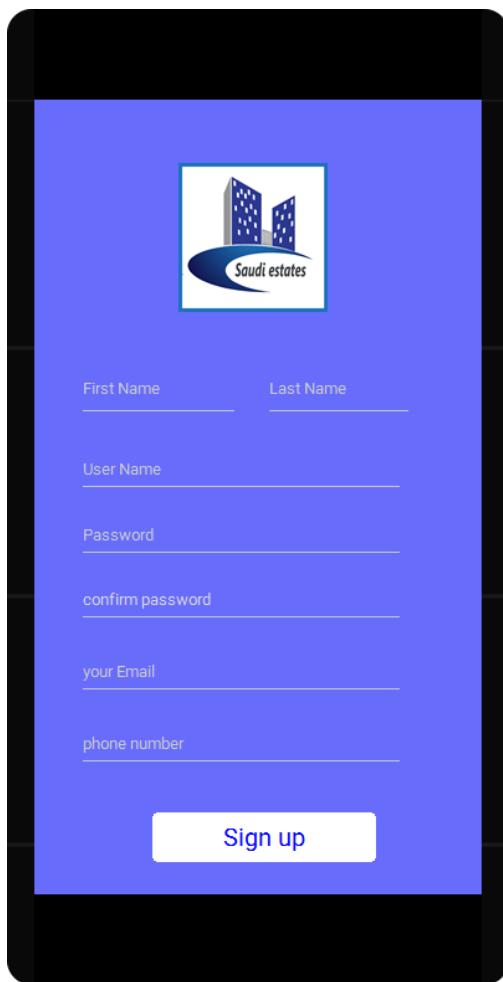


Figure 26: Interfaces of sign up forms.

Interfaces of Home page:

Here screen appears after a user login or pressing skip. It contains search bar, map of Saudi Arabia to select the region he wants to search into the open Google map for what he selected, and the new estates added in the application.

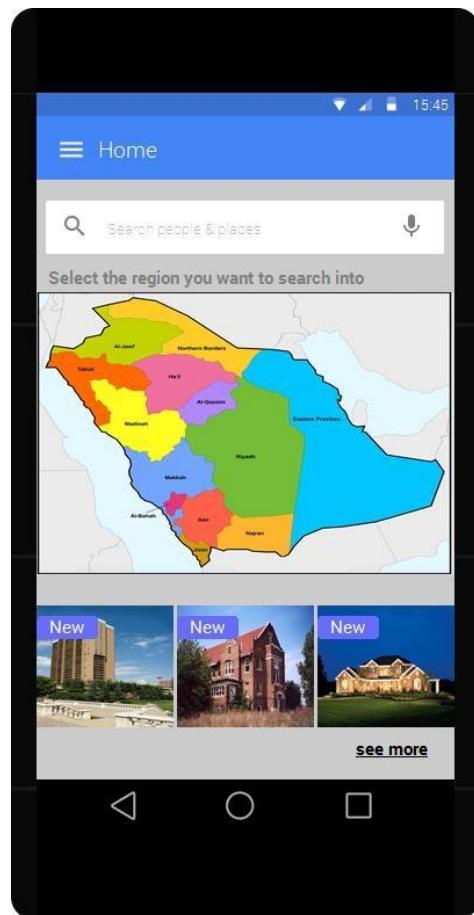


Figure27: Interfaces of Home page.

Interfaces of User profile:

This screen appears when the user logs in and pulls to the right, and it contains list of functions available to the user. When the user doesn't log in some of the functions will not appear in the profile like My Estates, Favorite, Collection, Chat.

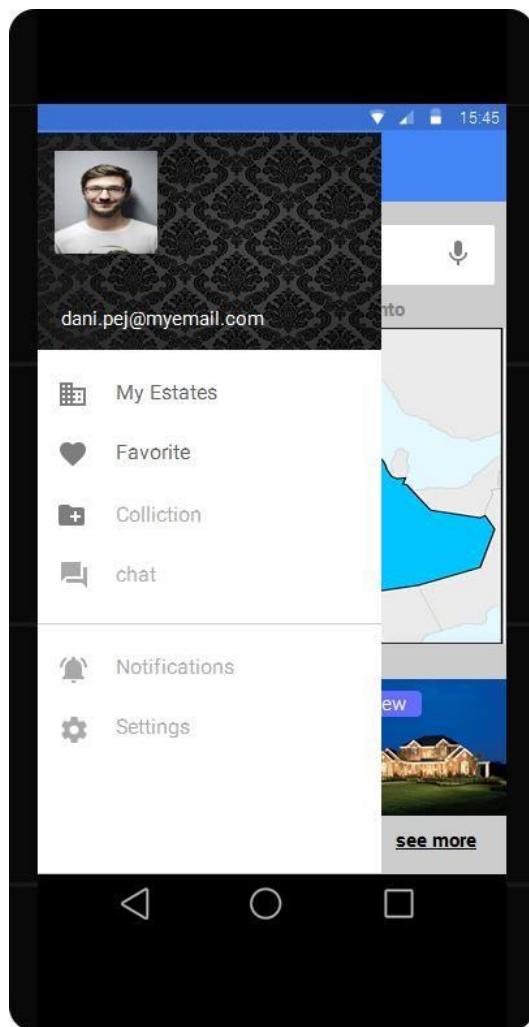


Figure 28: Interfaces of User profile

Interfaces of My Estates:

This screen appears after the user pressing My Estates from his profile, this screen showed the user estate he offers for sale or rent, he can see the details of his estates or delete or update his estate. The user can also add new estates by pressing the plus sign. Ad posting agreement interface will appear to the user with Agree button, the user must read then pressing I Agree to a continuation.



Figure 29: Interfaces of My Estates

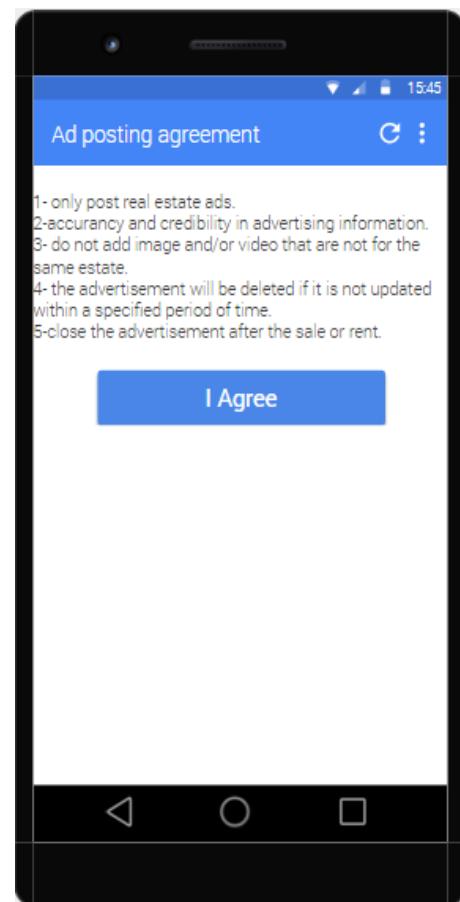


Figure 30: Ad posting agreement interface

Interfaces of Add Estates:

These screens appear after the user pressing I Agree button, the form shown will appear to the user and the user will fill out the form with the details about the estate he will advertising it. For example, he will determine if it is for sale or for rent, and make a name for the advertisement, select the type of estate it is land, villa, house, building or another, then upload images or videos and the location of this estate. The user will fill all form and details that also will appear after pressing Continue button.

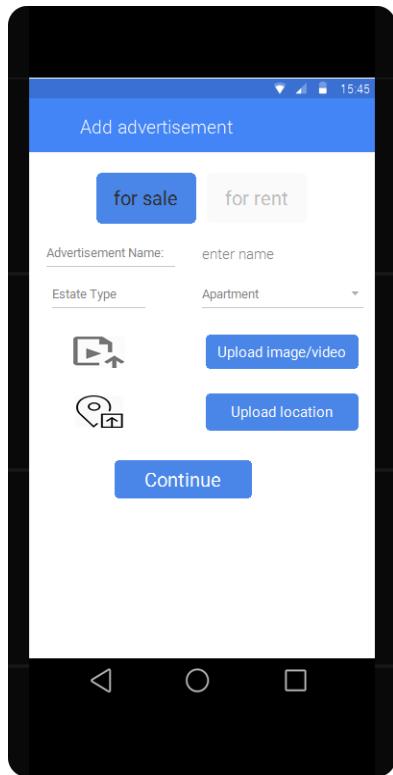


Figure 31: Interfaces of Add Estates

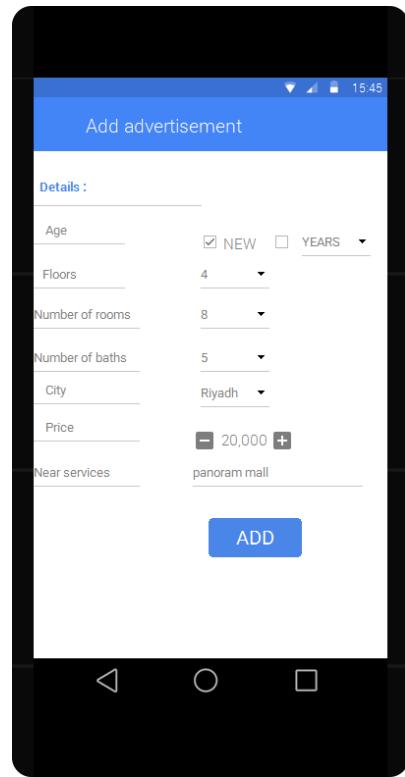


Figure 32: Interfaces of Add Estates

Interfaces of Advertising:

This screen appears after the user pressing on estate advertisement, and after finish from Add Estates forms. It contains the information about the advertiser and his rating, the details about the estate like the location, the price the number of room and floor, near services and more.

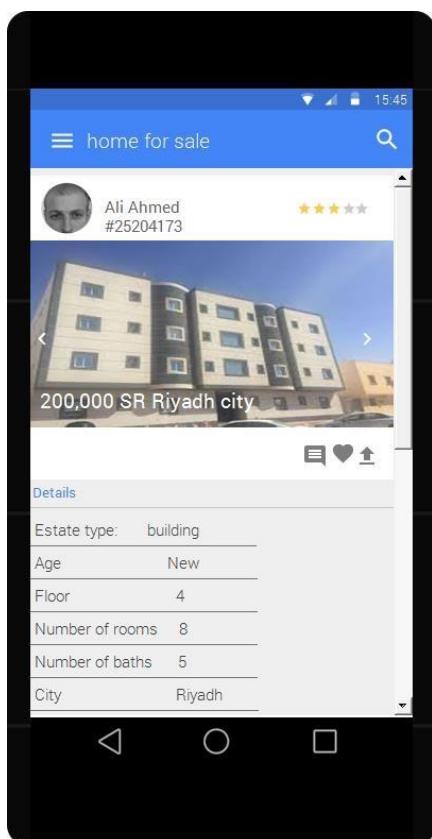


Figure 33: Interfaces of Advertising

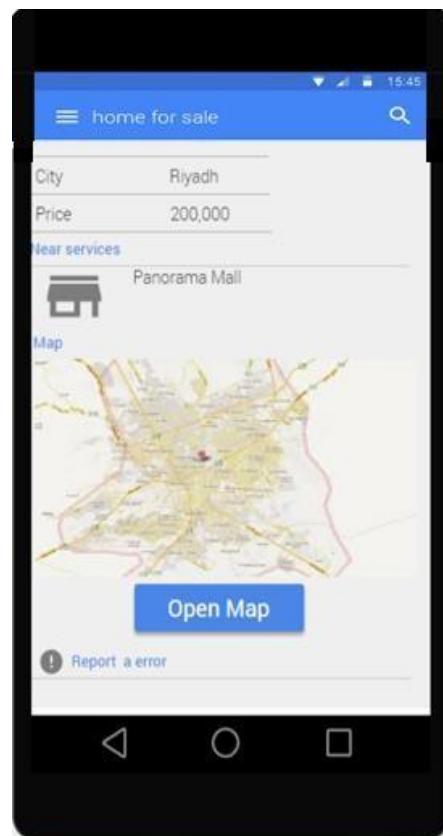


Figure 34: Interfaces of Advertising

Interfaces of Google map and search by location:

This screen appears when the user selects the region from the map on the homepage or when he wants to search by location. Near me button will show the user the estates that near him for sale or rent.

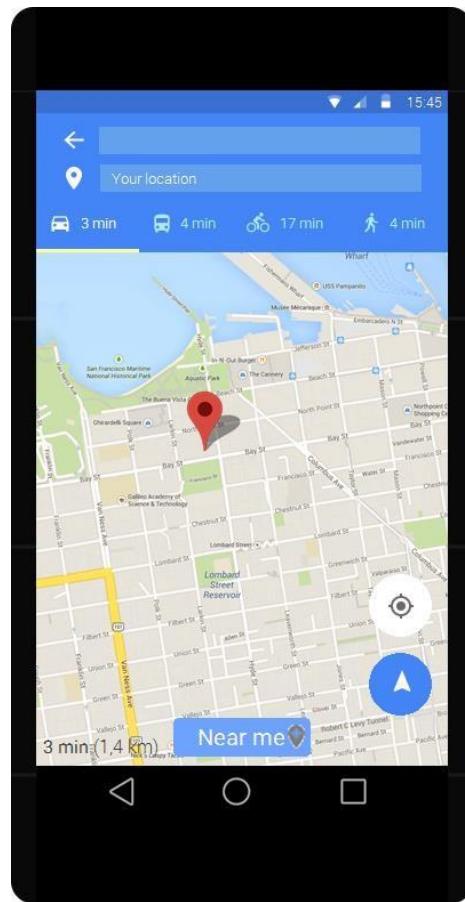


Figure 35: Interfaces of Google map

Interfaces of Settings:

This screen appears when the user clicks on settings, and it contains list of main account setting like edit profile and change password, and other options like contact, languages, help canter, report a problem, privacy policy, and terms. Also, clear search history button and log out

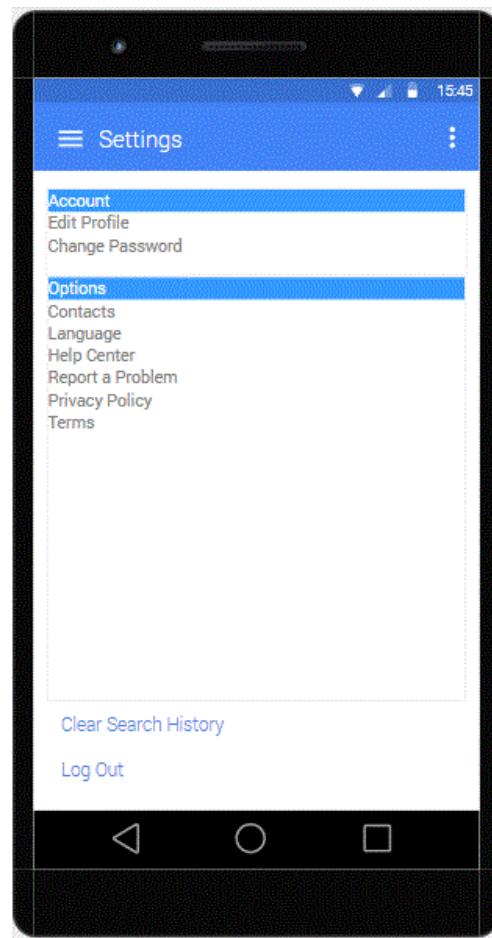


Figure 36: Interface of Settings

Interfaces of Chat:

This screen appears when the user clicks on Chat, and it contains list of other users that the user contact with them.

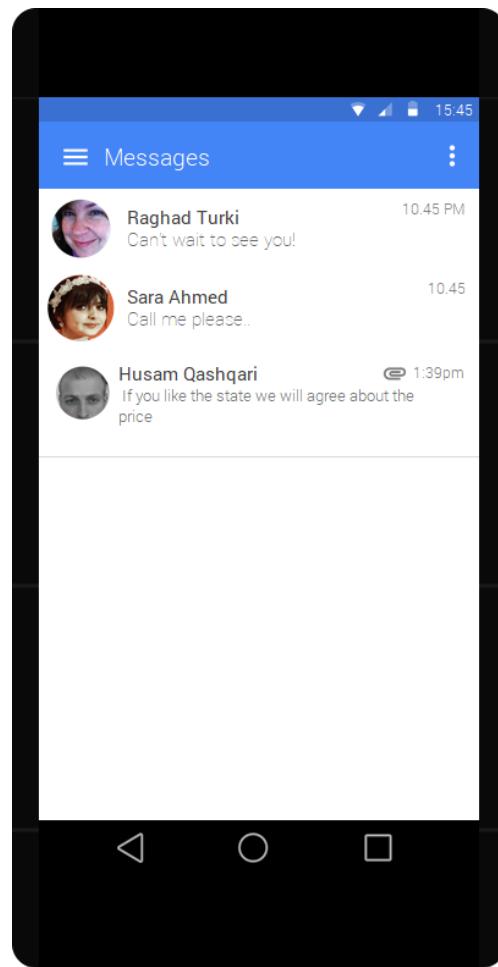


Figure 37: Interfaces of Chat

Interfaces of Admin:

This screen appears for the admin, and it contains list of functions available to the admin, when we receive a report about a user who violate laws or write false information we will add him to the blacklist , and if his name is repeated in the Blacklist 5 times, we will block him and delete his estates and he will not add estates anymore.

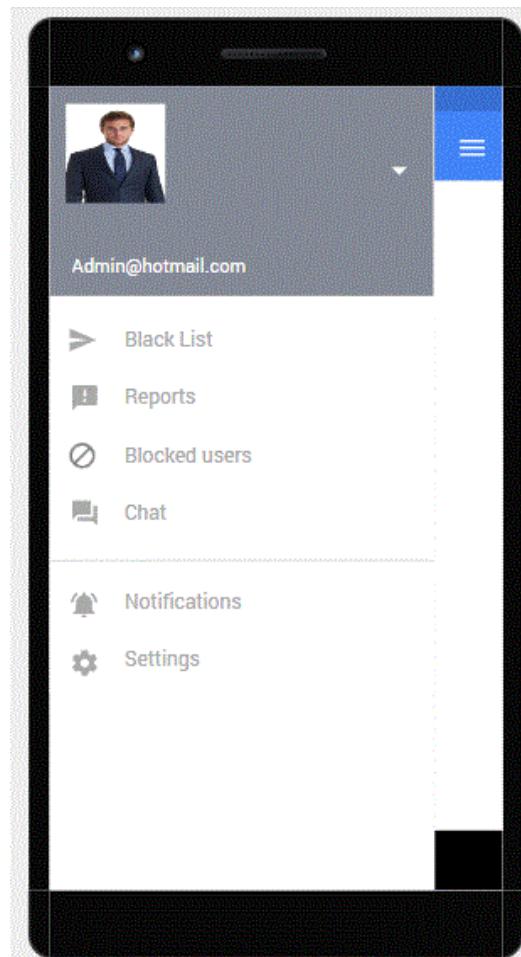


Figure 38: Interfaces of Admin

Interfaces of notifications:

- 1- This is the lock screen where the Saudi estates notifications appear.

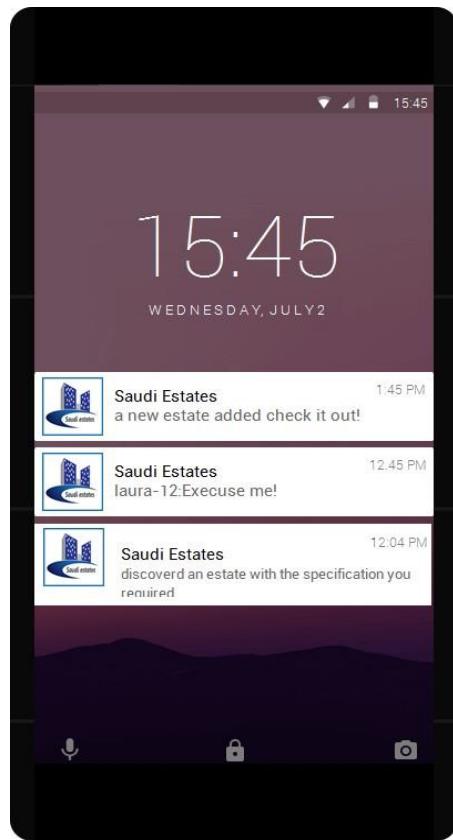


Figure 39: lock screen notification

2-this the interface where you can trigger the notifications and there're Two kinds of notifications.

- General notifications for example (if there's a new estate added to the application, direct messages).
- Special notifications with the specification you choose when you trigger it for example (you want to get notified about a villa with 2 floors and with a specific price and the city you desire).

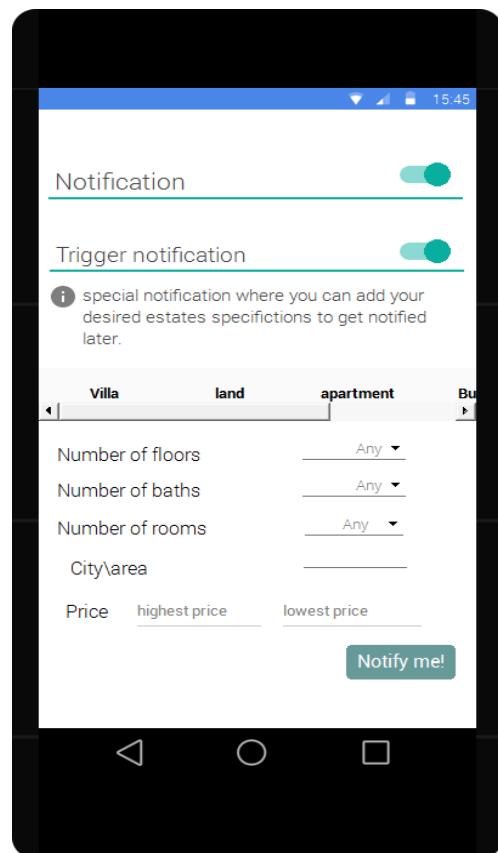


Figure 40: trigger notification

Interface of specification search:

This is the interface where you can filter to a specific search in the search bar with the estate type you choose you can fill in the data you want and press filter to get the results.

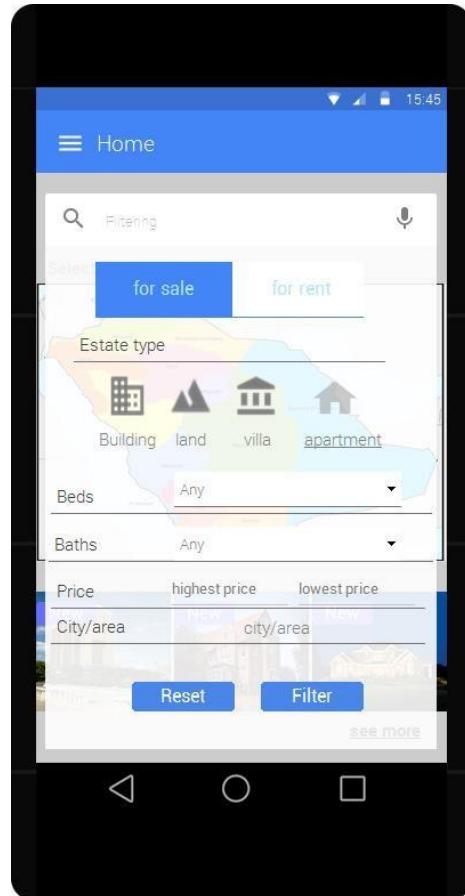


Figure 41: specific search

Interfaces of Log out:

This screen appears after the user pressing on log out from Settings interface. Alert Asking the user if he is sure wants to log out of the application.

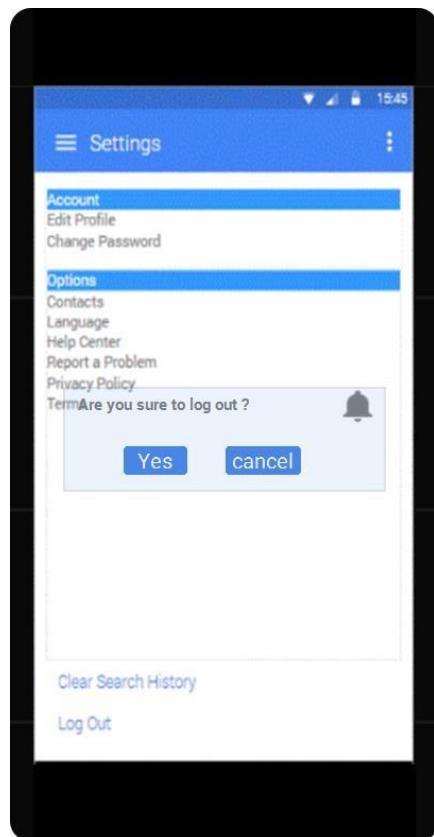
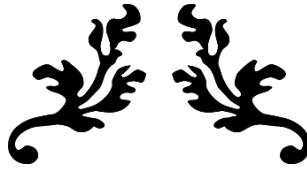


Figure 42: Interfaces of Log out



CHAPTER 5:

IMPLEMENTATION

AND TESTING

5.1 Implementation Phase

5.1.1 Connecting part

5.1.1.1 Connecting the Application with Database

5.1.2 Designing Interface

5.1.2.1 User Interface

5.2 Test

5.2.1 Test Tools and Environment

5.2.2 Testing Scenarios

5.2.2.1 Exception Handling

5.2.2.2 Completed Scenarios

5.3 Challenges

5.4 Conclusion



Introduction

Now, we start to confirm theoretical side that we discussed in the previous chapters to the practical side to make This project. Also, we specify implementation and test process of Saudi Estates Application and we will explain how we implemented this appellation step by step. Furthermore, we'll describes all the app interface (user and admin interface) in details.

5.1 Implementation Phase

5.1.1 Connecting part

After we download Android Studio from ([Download Android Studio\(3.1\) for Windows.](#)) and install it with Software Development Kit (SDK). And then we connect the project to firebase. Firebase real-time database where it does not use tables like(SQL), it stores data as (JSON) object. When you add data to the JSON tree, it becomes a node in the existing JSON structure with an associated key.

5.1.1.1 Connecting the Application with Database

Here we will explain how we connect the app with Firebase Database. Must be realized first of the basic Requirements are: A device running Android 4.0 (Ice Cream Sandwich) or newer, and Google Play services 12.0.1 Or higher and the latest version of Android Studio.

Add Firebase to your app:

- Click Tools > Firebase to open the Assistant window.
- Click to expand one of the listed features (Analytics).
- Click the Connect to Firebase button to connect to Firebase and add the necessary code to your app.

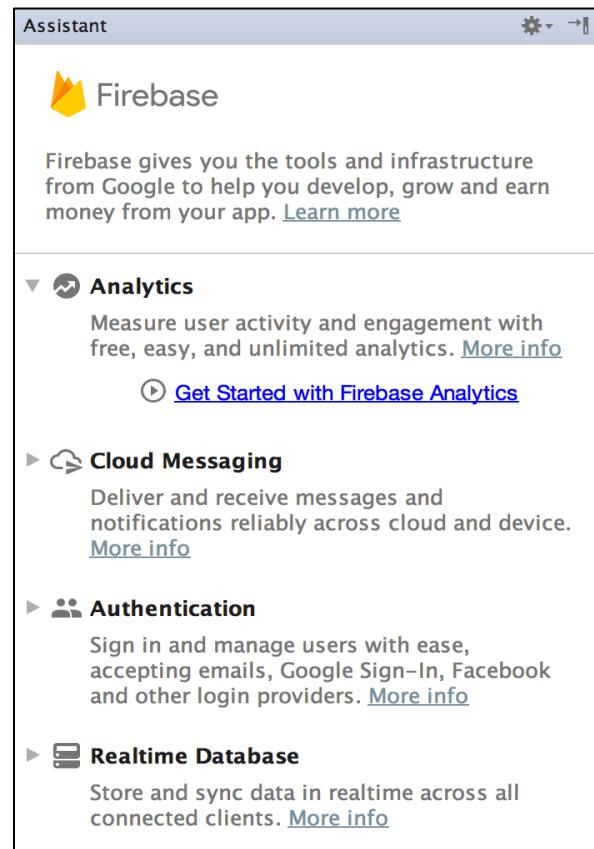


Figure 43: The Assistant tool window in Android Studio.

Create project on the firebase:

After we go to ([Firebase Console](#)) choose add project and fill the required data and then choose the platform That fit to you (we will choose an android), see figure 44 ,45 and follow the steps.

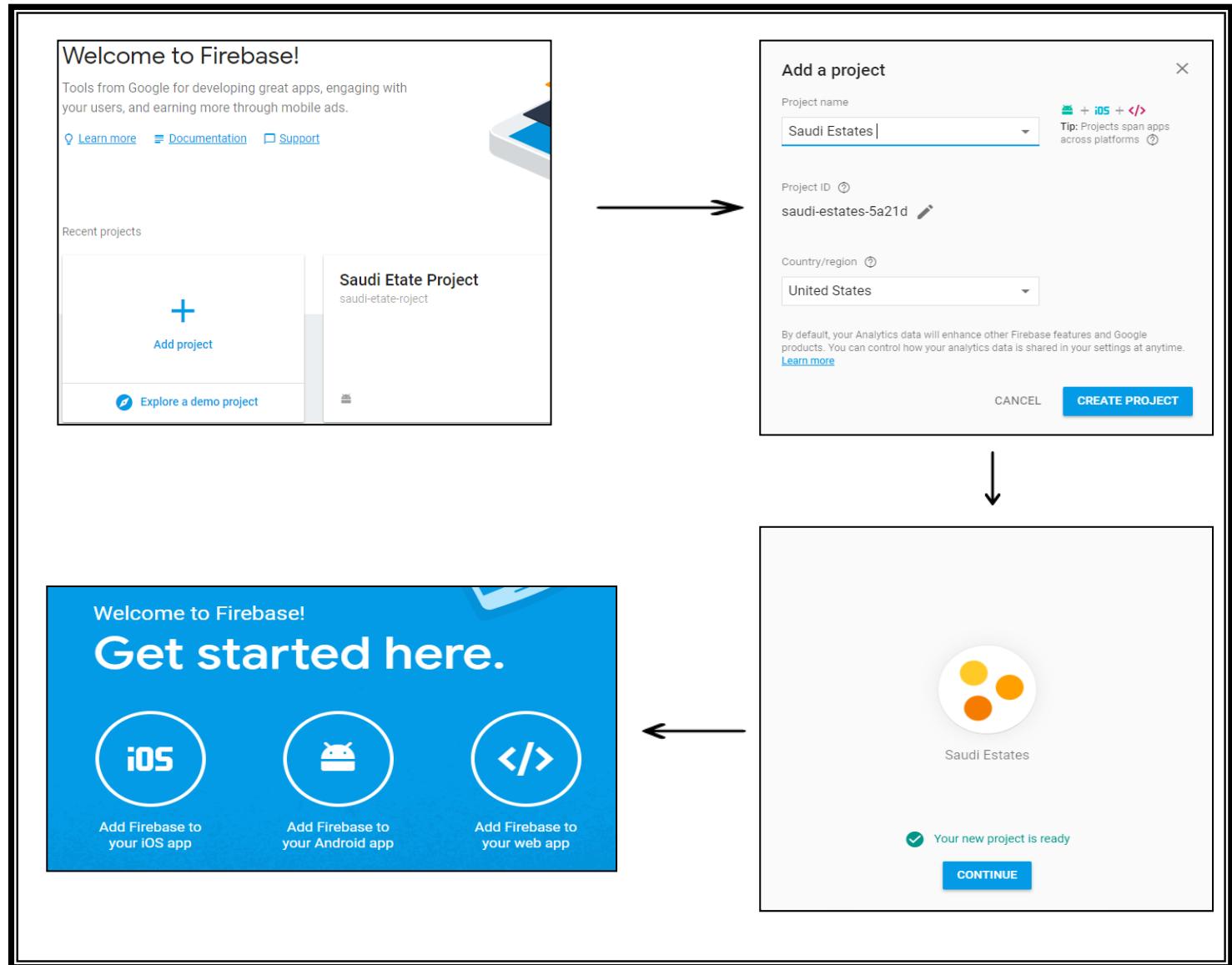


Figure44: Firebase Console part1

After that you must add firebase to your app, this done by doing three steps:

- Register app (by adding package name and SHA-1(you can get SHA-1 from click on Gradle in android project and the choose firebase → android → signing Report)).
- Download the config file.
- Add Firebase SDK.

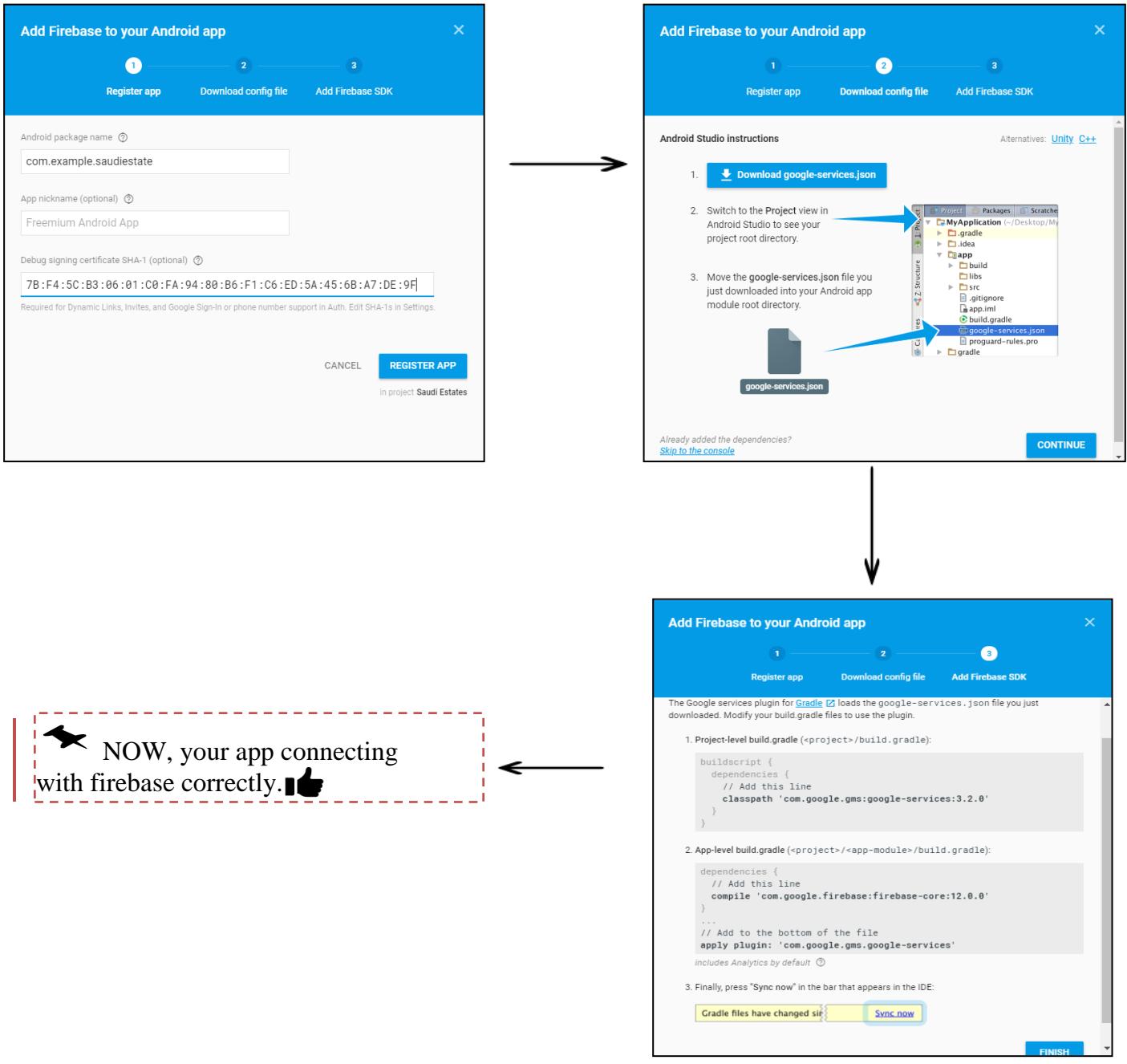


Figure45: Firebase Console part2

5.1.2 Designing Interface

In this part interface for the application will be built. We will display the user and admin interface.

5.1.2.1 User Interface

There are many interfaces for the user:

- **Splash screen:**

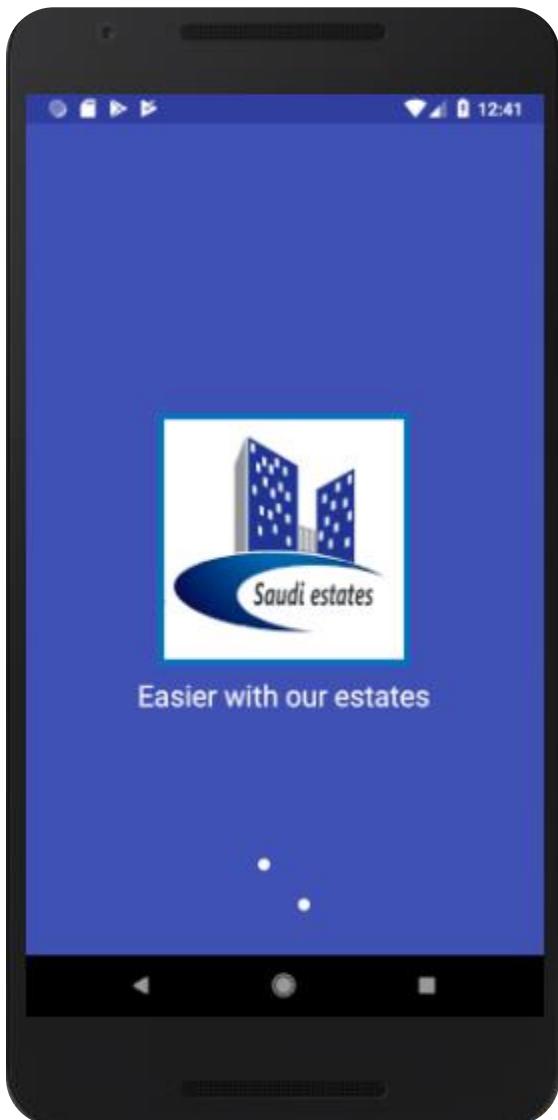


Figure46: splash screen

```
public class welcome extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_welcome);  
        new Handler().postDelayed(new Runnable() {  
            @Override  
            public void run() {  
                Intent i = new Intent( packageContext: welcome.this, welcome2.class);  
                startActivity(i);  
                finish();  
            }  
        }, delayMillis: 3 * 1000);  
    }  
}
```

Splash screen is an activity that will show for a specified period of time when the app is starting. we use handler (class in the Android) and then we call **Handler(). postDelayed**, it will call **run()** method of runnable after set time (in this app we set 3 second) and redirect to main app screen.

- sign-in and sign-up interface:

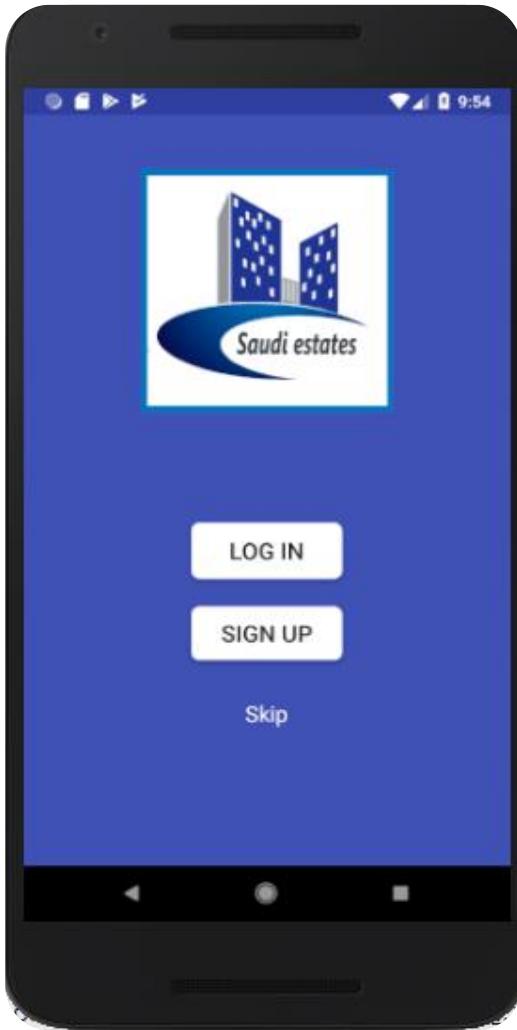


Figure47: login, signup and skip

In the first screen (figure 46) the users could create new account if he/she does not have it or log in directly if he/she has. Also, can skip this step and go to the home page.

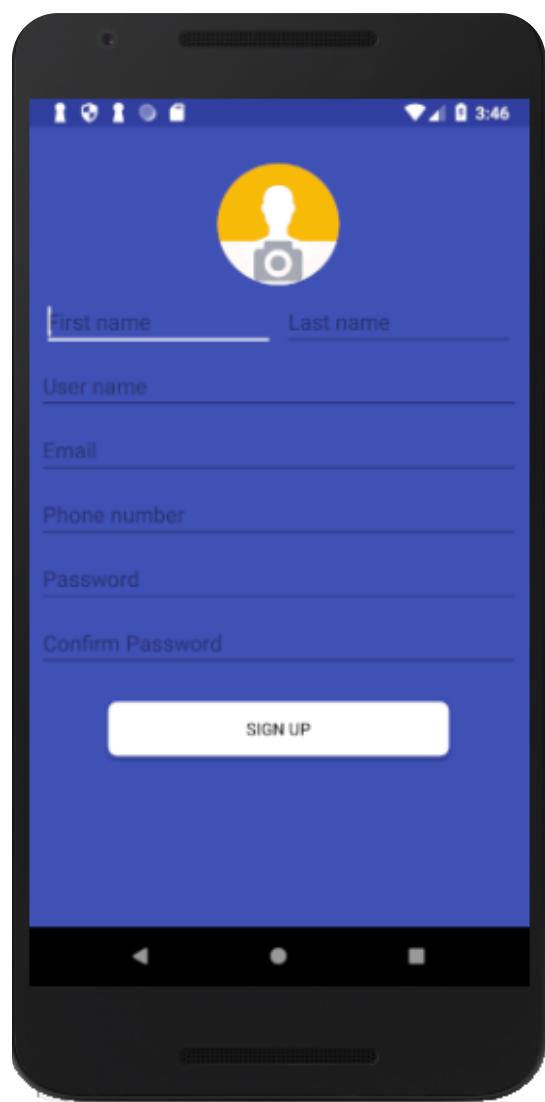
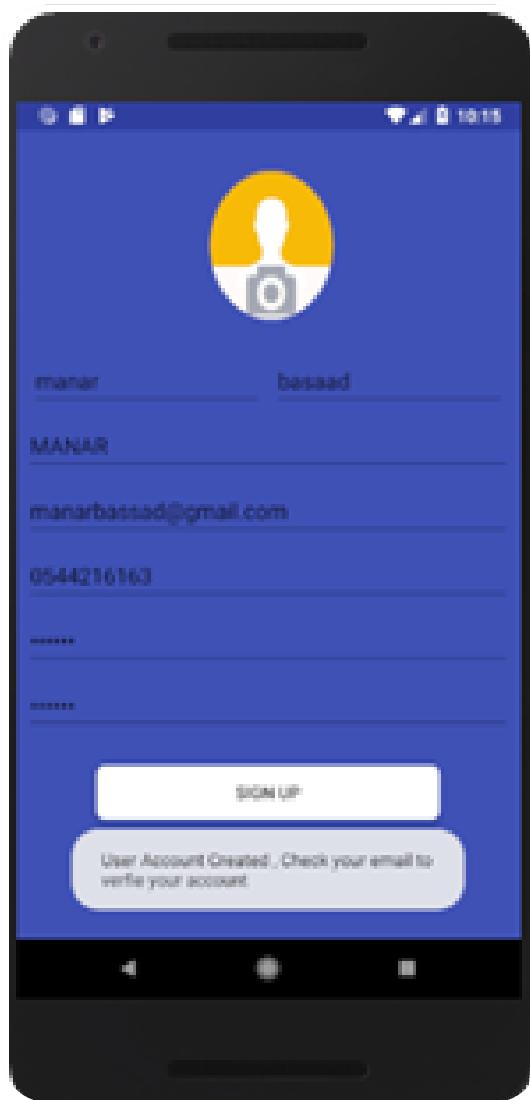


Figure48: sign up



After the user fill the required data an email will be sent to verify their account like in figure 49.



Figure49: create new account

Figure50: verify your email

```

final FirebaseAuthUser user = mAuth.getCurrentUser();
user.sendEmailVerification()
    .addOnCompleteListener( activity: signup.this, (OnCompleteListener) (task) -> {
        if (task.isSuccessful()) {
            Toast.makeText(signup.this,
                "Verification email sent to " + user.getEmail(),
                Toast.LENGTH_LONG).show();
        }
        if (Locale.getDefault().getLanguage().equals("ar")) {
            Toast.makeText( context: signup.this, text: "تم إنشاء حساب جديد . تم إرسال رابط التفعيل إلى بريدك",
                Toast.LENGTH_LONG).show();
        } else {
            Toast.makeText( context: signup.this, text: "User Account Created , " +
                "Check your email to verify your account", Toast.LENGTH_LONG).show();
        }
    } else {
        Log.e( tag: "ConfirmEmail", msg: "sendEmailVerification", task.getException());
        Toast.makeText( context: signup.this,
            text: "Failed to send verification email.",
            Toast.LENGTH_LONG).show();
    }
});
```

Email Verification:

to implement this function just call **sendEmailVerification()** method by using the object "user" is object from FirebaseAuthUser and then add an **OnCompleteListener** to the Task.
If the task **isSuccessful()** the email will be sent.



Figure51: email has been verified.

```

private void checkUserValidation(DataSnapshot dataSnapshot, String emailForVer) {
    if (user.isEmailVerified()) {
        Iterator iterator = dataSnapshot.getChildren().iterator();
        while (iterator.hasNext()) {
            DataSnapshot dataUser = (DataSnapshot) iterator.next();
            if (dataUser.child("Email").getValue().toString().equals(emailForVer)) {
                Intent in = new Intent( packageContext: login.this, Main2Activity.class);
                in.putExtra( name: "USER_KEY", dataUser.child("userKey").getValue().toString());
                startActivity(in);
                SharedPreferences.Editor editor = getSharedPreferences( name: "USERS", MODE_PRIVATE).edit();
                editor.putString( $: "userKey", dataUser.child("userKey").getValue().toString());
                editor.putString( $: "userName", dataUser.child("Username").getValue().toString());
                editor.putString( $: "userEmail", dataUser.child("Email").getValue().toString());
                editor.apply();
            }
        }
    }
}

```

Shared Preferences:

The benefit we got from using it, is store user data even if the application is closed. It acts as a temporary storage of user data it will stored (**userkey**, **userName** and **userEmail**).

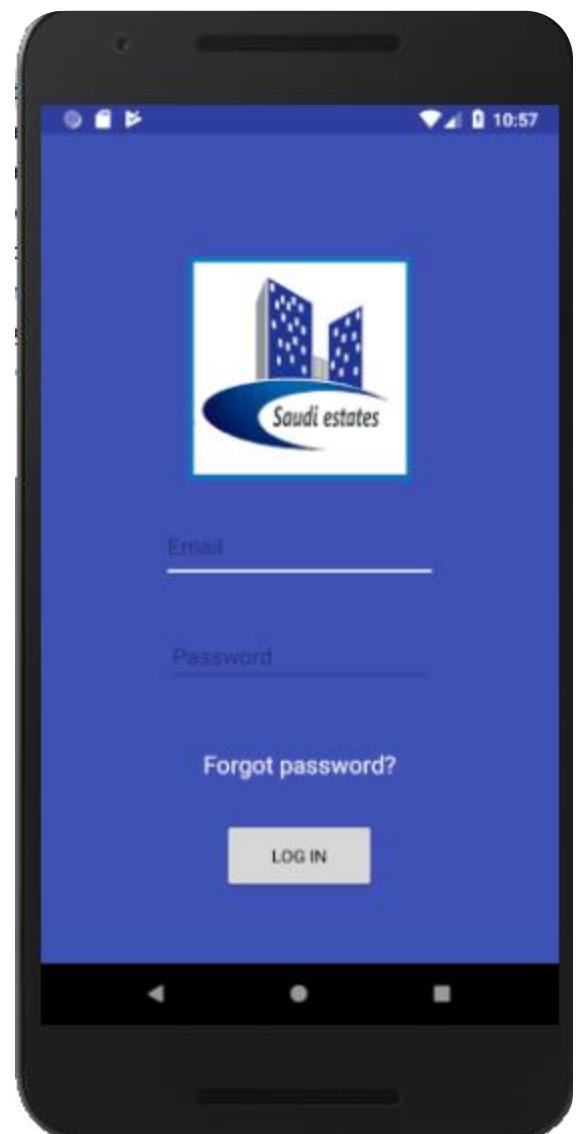


Figure 52: log in.

Home:

-Map fragment:

Maps are represented in the API by the GoogleMap and MapFragment classes.

OnMapReadyCallback() Once an instance of this interface is set on a MapFragment object, the **onMapReady(GoogleMap)** method is triggered when the map is ready to be used. **onMyLocationChange (Location location)** called to access to the location of the user.

```
mapfragment.getMapAsync(new OnMapReadyCallback() {
    @Override
    public void onMapReady(GoogleMap googleMap) {
        if (googleMap != null) {
            mMap = googleMap;
            mMap.clear();
            GetEstates();
            mMap.setOnMyLocationChangeListener(new GoogleMap.OnMyLocationChangeListener() {
                @Override
                public void onMyLocationChange(Location location) {
                    lat = location.getLatitude() + "";
                    lan = location.getLongitude() + "";
                    if (open == false) {
                        lat = location.getLatitude() + "";
                        lan = location.getLongitude() + "";
                        open = true;
                    }
                    LatLng latLng = new LatLng(location.getLatitude(), location.getLongitude());
                    MarkerOptions markerOptions = new MarkerOptions();
                    markerOptions.position(latLng);
                    mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));
                    CameraPosition cameraPosition = new CameraPosition.Builder().target(latLng).zoom(15).build();
                    mMap.animateCamera(CameraUpdateFactory.newCameraPosition(cameraPosition));//
                }
            });
        }
    }
});
```

also, we Permission from user to access to access his location through use the following code:

```
private void checkLocationPermission() {
    if (ContextCompat.checkSelfPermission(getActivity(), Manifest.permission.ACCESS_FINE_LOCATION)
        != PackageManager.PERMISSION_GRANTED) {
        if (ActivityCompat.shouldShowRequestPermissionRationale(getActivity(),
            Manifest.permission.ACCESS_FINE_LOCATION)) {
            new AlertDialog.Builder(getActivity())
                .setTitle("Location Permission Needed")
                .setMessage("This app needs the Location permission, please accept to use location functionality")
                .setPositiveButton(text: "OK", (dialogInterface, i) -> {
                    //Prompt the user once explanation has been shown
                    ActivityCompat.requestPermissions(getActivity(),
                        new String[]{Manifest.permission.ACCESS_FINE_LOCATION,
                            MY_PERMISSIONS_REQUEST_LOCATION});
                })
                .create()
                .show();
        } else
            ActivityCompat.requestPermissions(getActivity(),
                new String[]{Manifest.permission.ACCESS_FINE_LOCATION,
                    MY_PERMISSIONS_REQUEST_LOCATION});
    }
}
```

```

public void GetEstates() {
    mDatabase.addChildEventListener(new ChildEventListener() {
        @Override
        public void onChildAdded(DataSnapshot dataSnapshot, String s) {
            Log.e( tag: "Response :::", dataSnapshot.getValue().toString());
            if (list.size() < 5) {
                list.add(dataSnapshot.getValue(Estates.class));
                for (int i = 0; i < list.size(); i++) {
                    double lat = Double.valueOf(list.get(i).Lat);
                    double lan = Double.valueOf(list.get(i).Lng);
                    DecimalFormat df = new DecimalFormat( pattern: "#.##");
                    df.format(lat);
                    df.format(lan);
                    Marker marker = mMap.addMarker(new MarkerOptions().position(new LatLng(lat, lan)).title(list.get(i).Name));
                }
            }
        }
    });
}

```

By call **GetEstates()** this method fetch all estates are added in app from firebase and then add the marker on map this marker takes **lat, lang and estate name**.

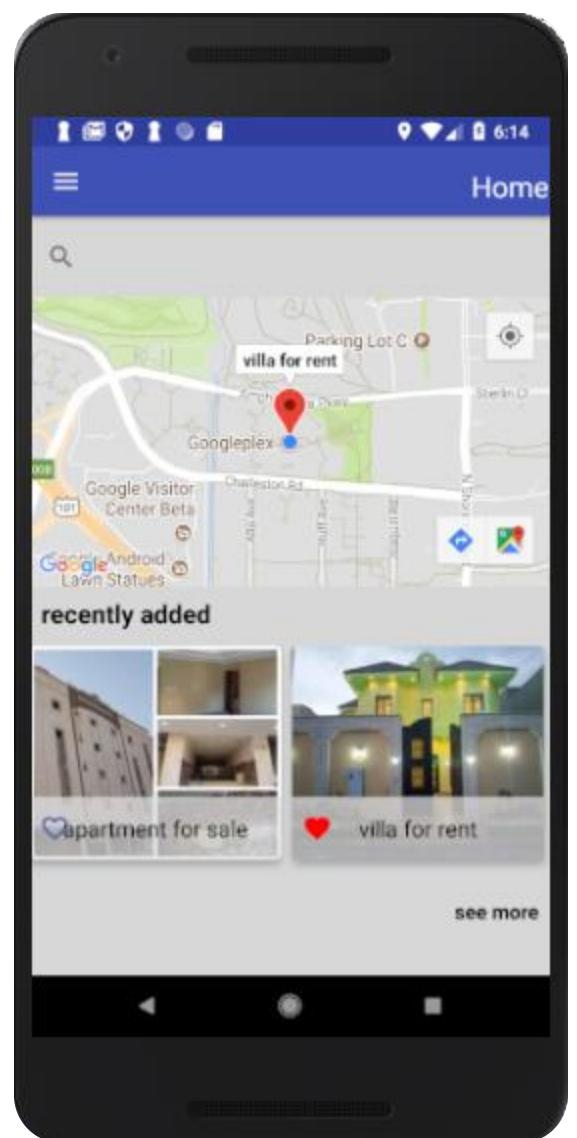


Figure 53:home interface.

-search function:

```
mDatabase.addChildEventListener(new ChildEventListener() {  
    @Override  
    public void onChildAdded(DataSnapshot dataSnapshot, String s) {  
        Log.e( tag: "Response :::", dataSnapshot.getValue().toString());  
        if (dataSnapshot.getValue(Estates.class).Age.contains(getIntent().getExtras().getString( key: "age")) ||  
            dataSnapshot.getValue(Estates.class).City.contains(getIntent().getExtras().getString( key: "city")) ||  
            dataSnapshot.getValue(Estates.class).EstateType.contains(getIntent().getExtras().getString( key: "tag")) ||  
            dataSnapshot.getValue(Estates.class).Floors.contains(getIntent().getExtras().getString( key: "floors")) ||  
            dataSnapshot.getValue(Estates.class).NearServices.contains(getIntent().getExtras().getString( key: "service")) ||  
            dataSnapshot.getValue(Estates.class).NoOfpaths.contains(getIntent().getExtras().getString( key: "paths")) ||  
            dataSnapshot.getValue(Estates.class).NoOfpaths.contains(getIntent().getExtras().getString( key: "paths")) ||  
            dataSnapshot.getValue(Estates.class).NoOfRooms.contains(getIntent().getExtras().getString( key: "rooms")) ||  
            dataSnapshot.getValue(Estates.class).NoOfRooms.contains(getIntent().getExtras().getString( key: "price")) ) {  
                list.add(dataSnapshot.getValue(Estates.class));  
  
                estatesAdapter.notifyDataSetChanged();  
            }  
    }  
}
```

The search feature provides the user to search for estates according to the specifications that he wishes , he can for example search for a house with the following details (age:25, floors:2, number of rooms:7 ,city:Riyadh,price:65000,..) and the results according to the specifications will appear.

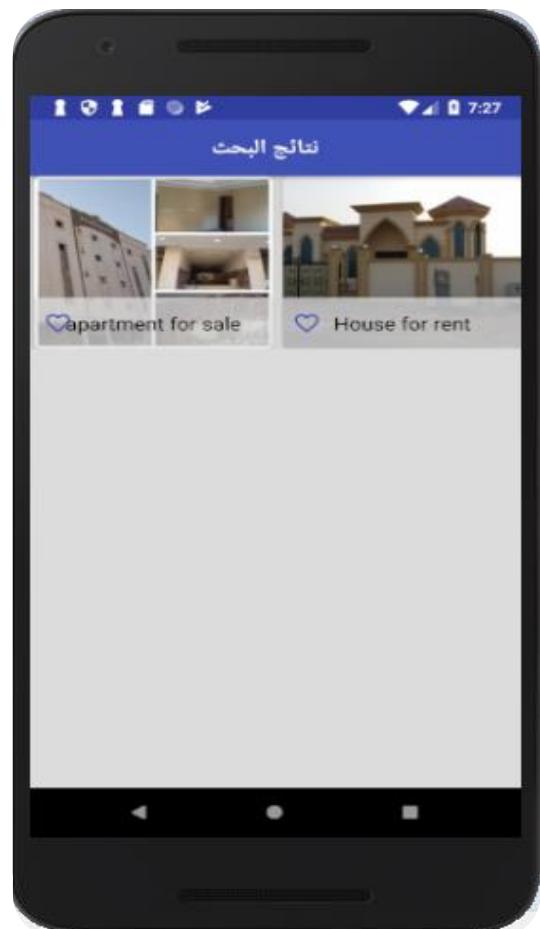


Figure 54: search results

Navigation Drawer:

The navigation drawer is a UI panel that shows your app's main navigation menu. It is hidden when not in use, but appears when the user swipes a finger from the left edge of the screen or, when at the top level of the app, the user touches the drawer icon in the app bar.

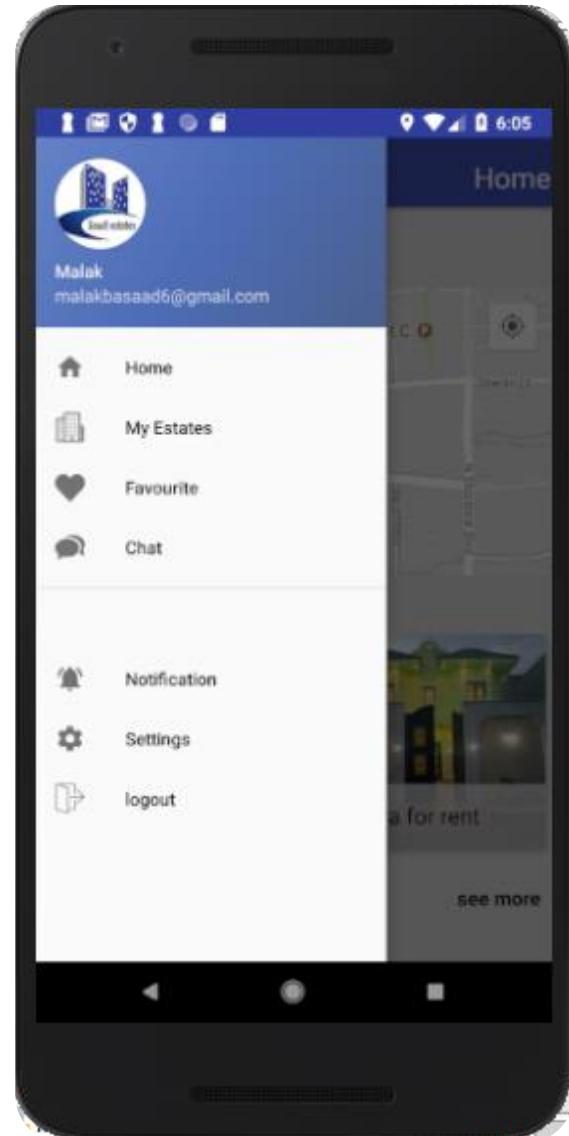


Figure 55: Navigation Drawer.

- **Creating Navigation Drawer:**

- **Toolbar:** Object is created and used in setSupportActionBar(toolbar) method. To use The created toolbar in the app_bar_main2.xml as an action bar in an app instead of the default Action bar.
- **DrawerLayout:** Object is created to allow the navigation bar to be pulled out from the edges of the window.
- **getSharedPreferences:** In order to store data to the SharedPreferences you need to first instantiate an instance of the SharedPreferences like so.

```
preferences = getSharedPreferences ("USERS", Context.MODE_PRIVATE);
```

The string `Users` is the name of the preference file you wish to access. If it does not exist, it will be created.

- **ActionBarDrawerToggle:** object is created to link the Activity that will contain the navigation bar and DrawerLayout and the ActionBar to implement navigation bar. this object has constructor which takes the Activity and DrawerLayout, Toolbar, String that describe "open navigation drawers ", String that describe " close navigation drawers "
- The **addDrawerListener(toggle);** called with ActionBarDrawerToggle object to handle events in drawers.
- **syncState:** it will synchronize the changed icon's state, which depends on actions of DrawerLayout. If you ever tried to remove it, the icon of arrow won't rotate any more.
- **getSupportFragmentManager:** Return the FragmentManager for interacting with fragments associated with this activity.
- Finally, we declare **navigationView** item by using **findViewById ()** to handle events on navigation items by calling **setNavigationItemSelectedListener ()**;
- **onNavigationItemSelectedListener ()**: Called when an item in the navigation menu is selected. And, the selected item. Returns boolean, true to display the item as the selected item. It checks if the id is equal to the clicked item id. for example, the id of the My Estate item is equal to action_my_estates id.

My Estates:

It's the activity contains the user's Estates that he added.

- in this activity we used an **Adapter**: are the link between a set of data and the AdapterView that displays the data.

AdapterView are ViewGroups that display child views given to it by an adapter.

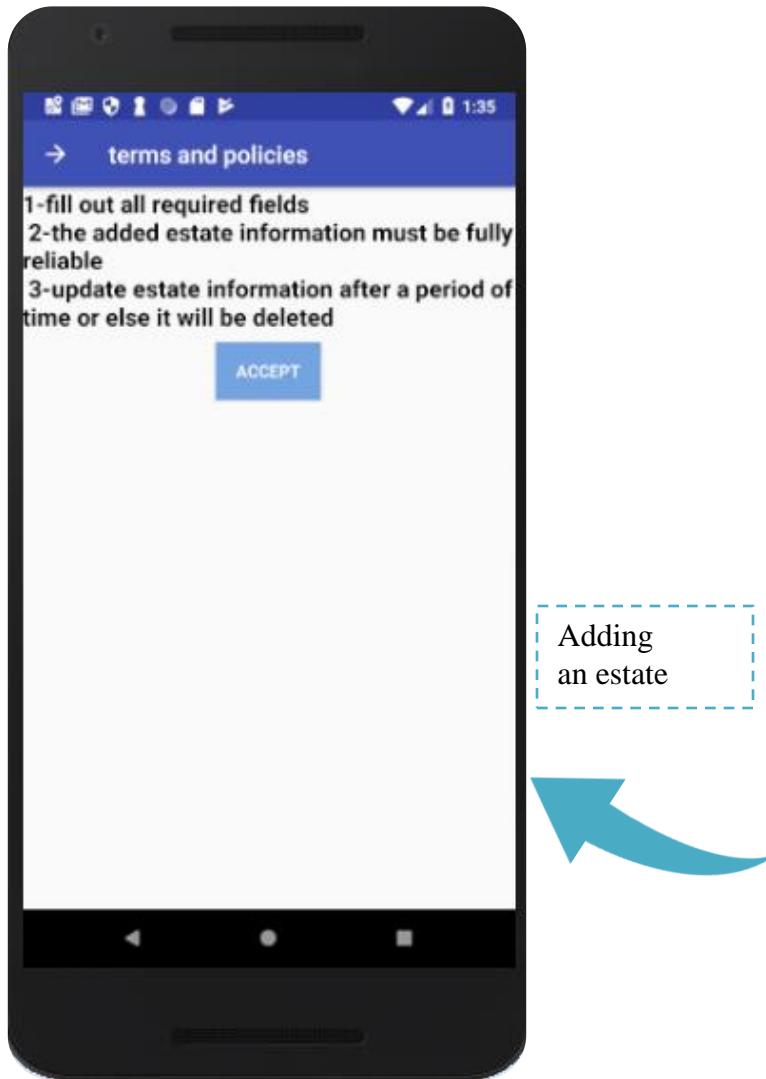


Figure 57: Terms and policies.

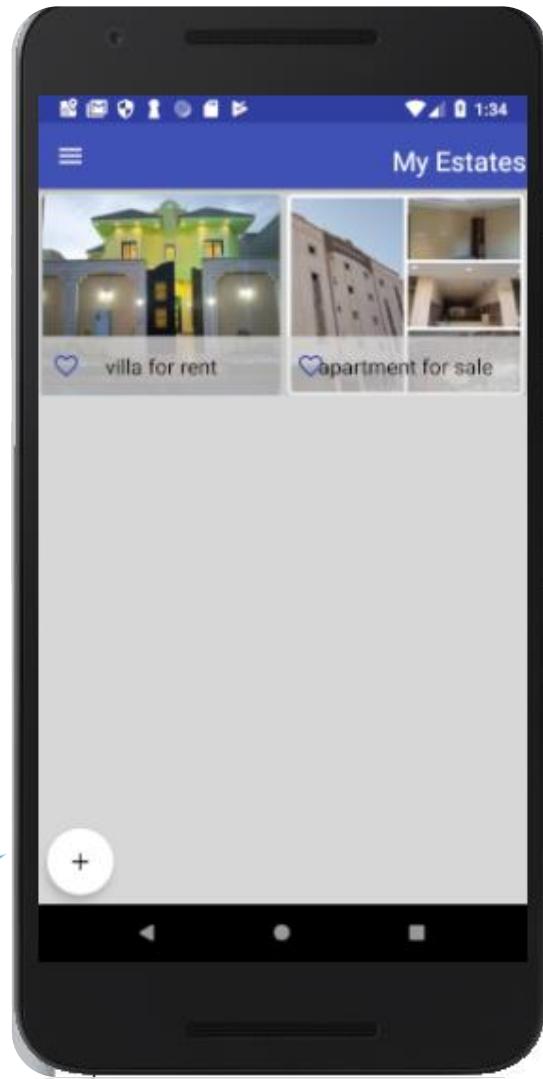


Figure 56: My Estate.

- Add an Estate:

- Upload image:

```
findViewById(R.id.btn_upload_img).setOnClickListener((view) -> {  
  
    if (ActivityCompat.checkSelfPermission(context: AddAdsActivity.this, android.Manifest.permission.READ_EXTERNAL_STORAGE)  
        != PackageManager.PERMISSION_GRANTED) {  
        checkCallPermission();  
    } else {  
        Intent intent = new Intent(Intent.ACTION_GET_CONTENT);  
        intent.setType("image/*");  
        startActivityForResult(intent, requestCode: 1);  
  
    }  
  
});
```

checkSelfPermission : check if you have a permission and If the app has the permission, the method returns PERMISSION_GRANTED.

checkCallPermission(): This *method* will be invoked when user allows or deny's a permission from the permission dialog so take actions accordingly.

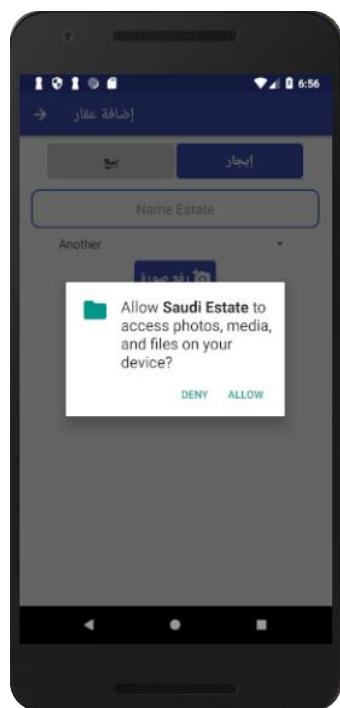


Figure 59: access permission.

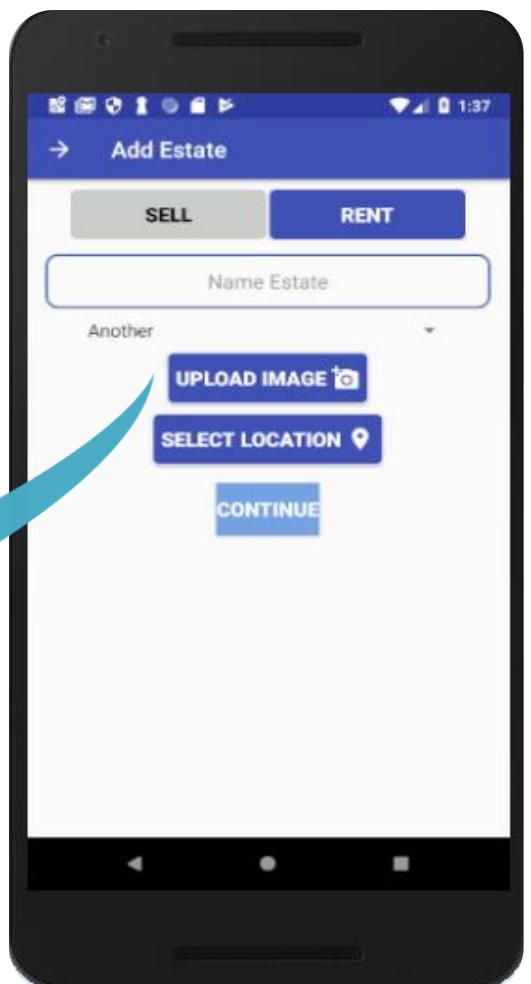


Figure 58: Add Estate.

-Select location:

```
findViewById(R.id.btn_upload_location).setOnClickListener((view) -> {
    PlacePicker.IntentBuilder builder = new PlacePicker.IntentBuilder();
    try {
        startActivityForResult(builder.build(activity: AddAdsActivity.this), PLACE_PICKER_REQUEST);
    } catch (GooglePlayServicesRepairableException | GooglePlayServicesNotAvailableException e) {
        e.printStackTrace();
    }
});
```

PlacePicker: provides a UI dialog that displays an interactive map and a list of nearby places, including places corresponding to geographical addresses and local businesses. Users can choose a place, and your app can then retrieve the details of the selected place.

GooglePlayServicesRepairableExceptions: are special instances of UserRecoverableExceptions which are thrown when Google Play Services is not installed, up-to-date, or enabled.

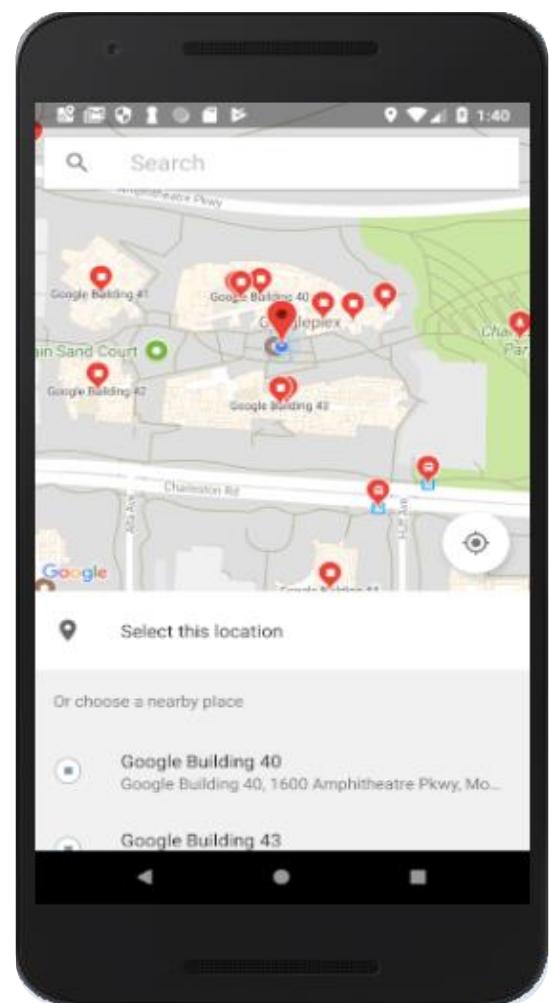


Figure 60: select location.

-Estate Details:

Add details of the estate (Age, Floors, Number of rooms, Number of baths, Price, City, Near services)

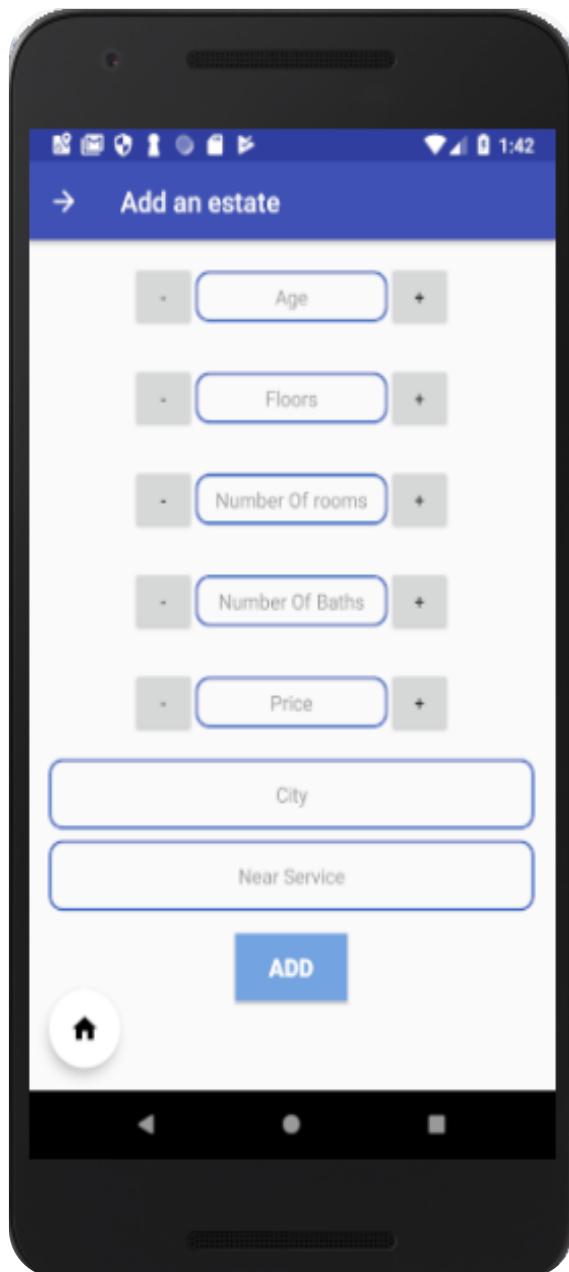
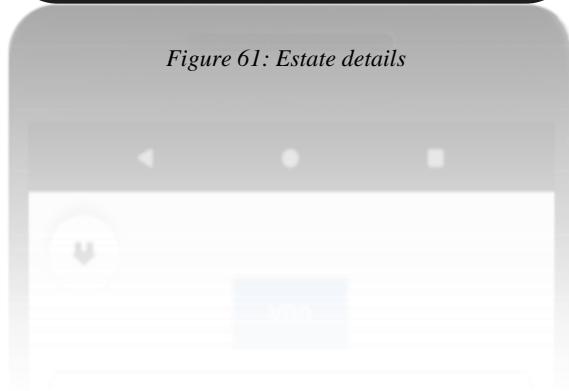


Figure 61: Estate details



-share button:

The user can share the estate to other application like (What's app, SMS messages, Emails, and other applications). [13]

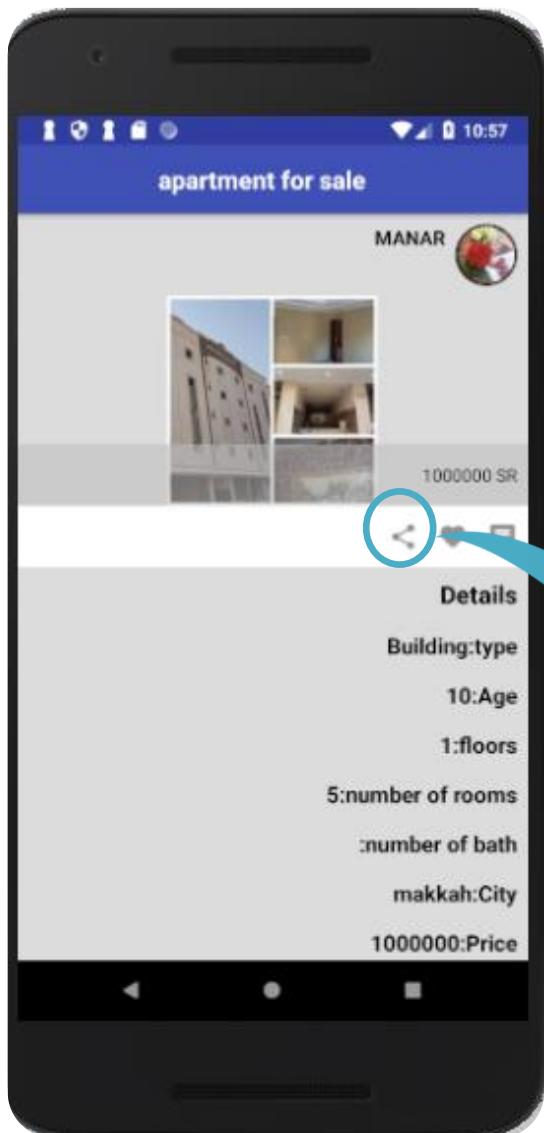


Figure 62: Estates details

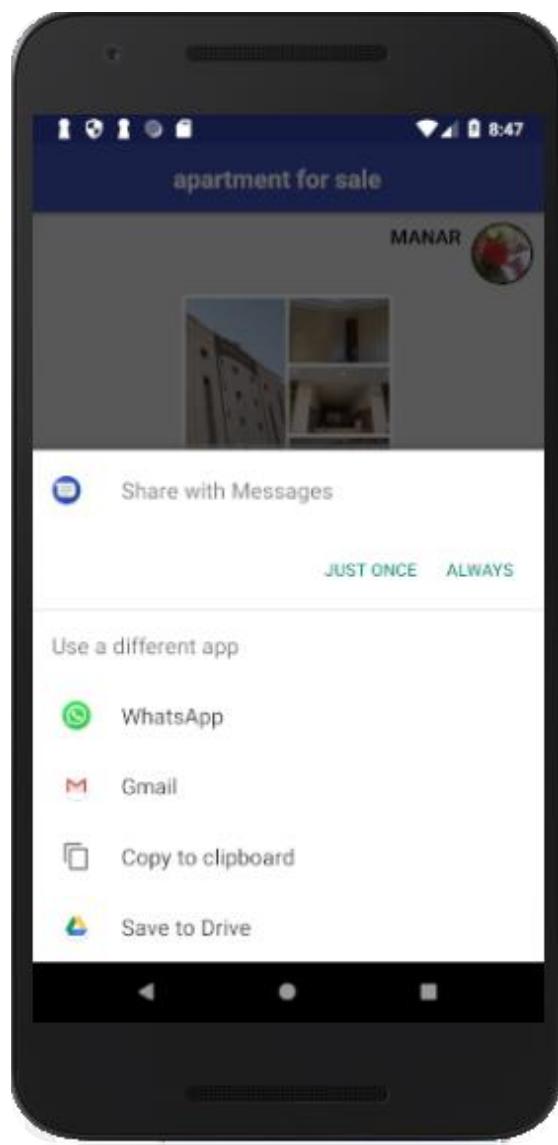


Figure 63: share button

```
findViewById(R.id.share_btn).setOnClickListener((v) -> {
    Intent sendIntent = new Intent();
    sendIntent.setAction(Intent.ACTION_SEND);
    sendIntent.putExtra(Intent.EXTRA_TEXT,
        value: "there's a new estate added in saudi estates check it out in the link below : " +
        "https://play.google.com/store/apps/details?id=com.example.saudiestate");
    sendIntent.setType("text/plain");
    startActivity(sendIntent);
});
```

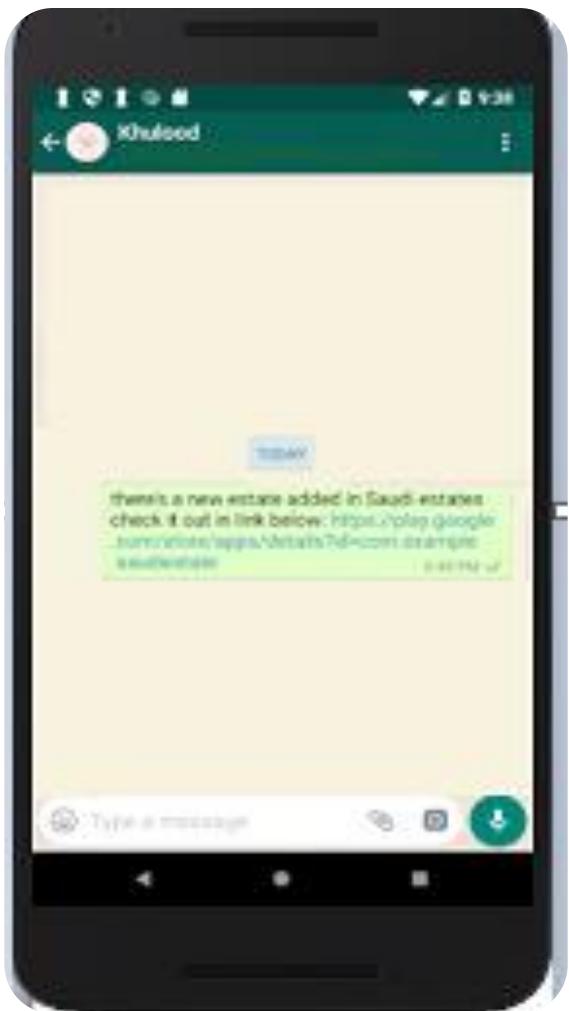


Figure 64 share to what's up

sendIntent.setAction(Intent.ACTION_SEND);
we specified the action we want the intent to "trigger." Android defines several actions, including ACTION_SEND. Which the intent will send data from one activity to another, **sendIntent.setType("text/plain");**
To send data to another activity, you need to specify the data and its type, the system will identify compatible receiving activities and display them to the user.

-Report a problem:

The user will be able to report about a certain estate if the estate

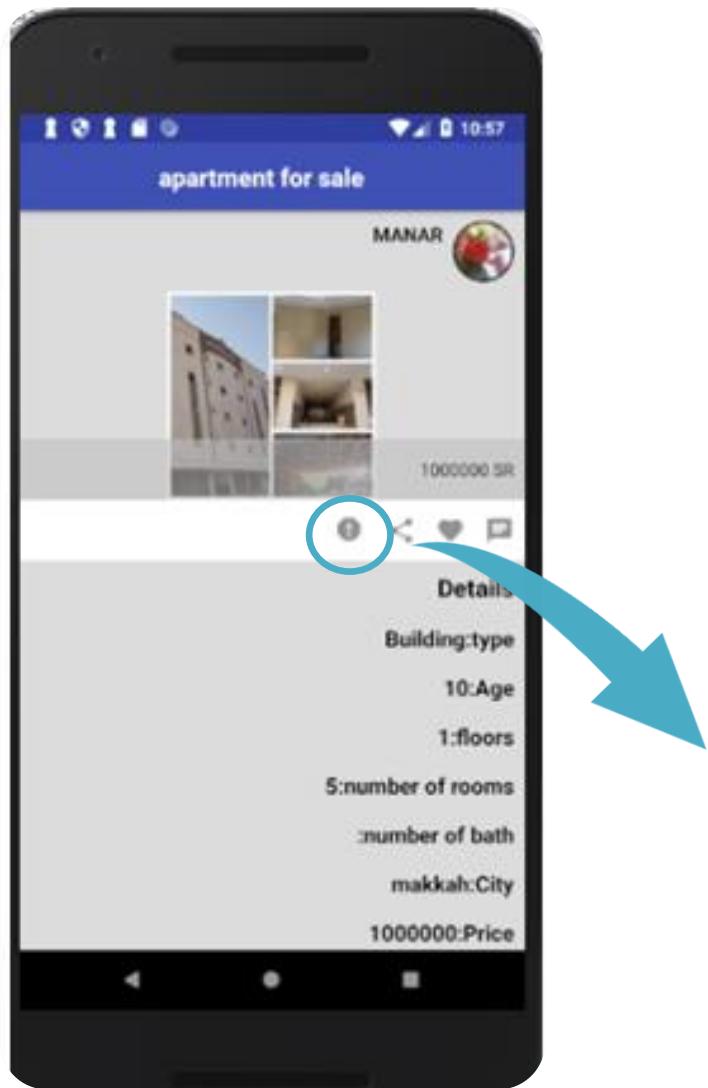


figure 65: estate details

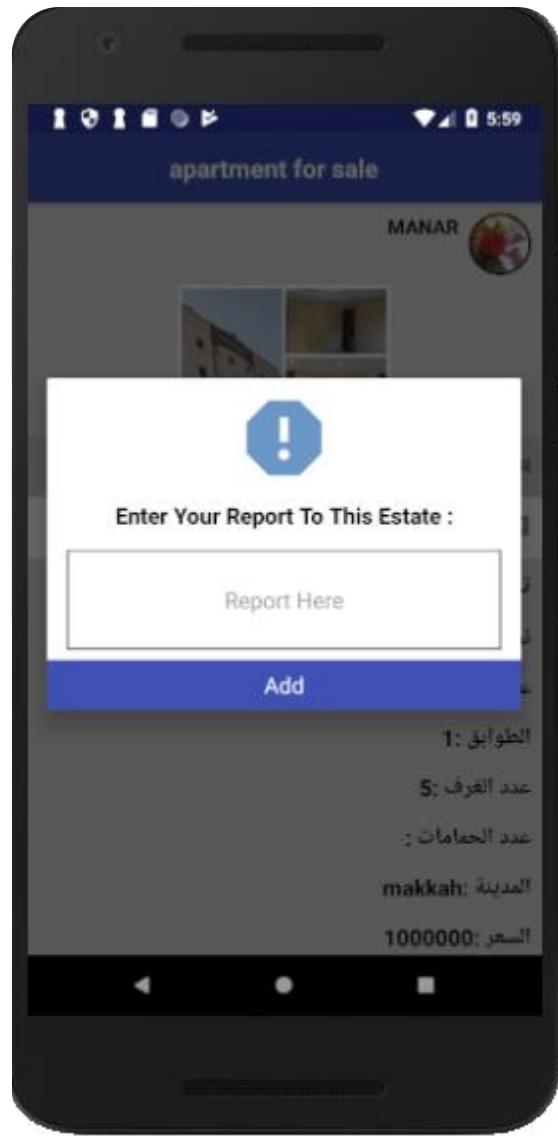


figure 66: Report a problem.

- **Delete Estate:**

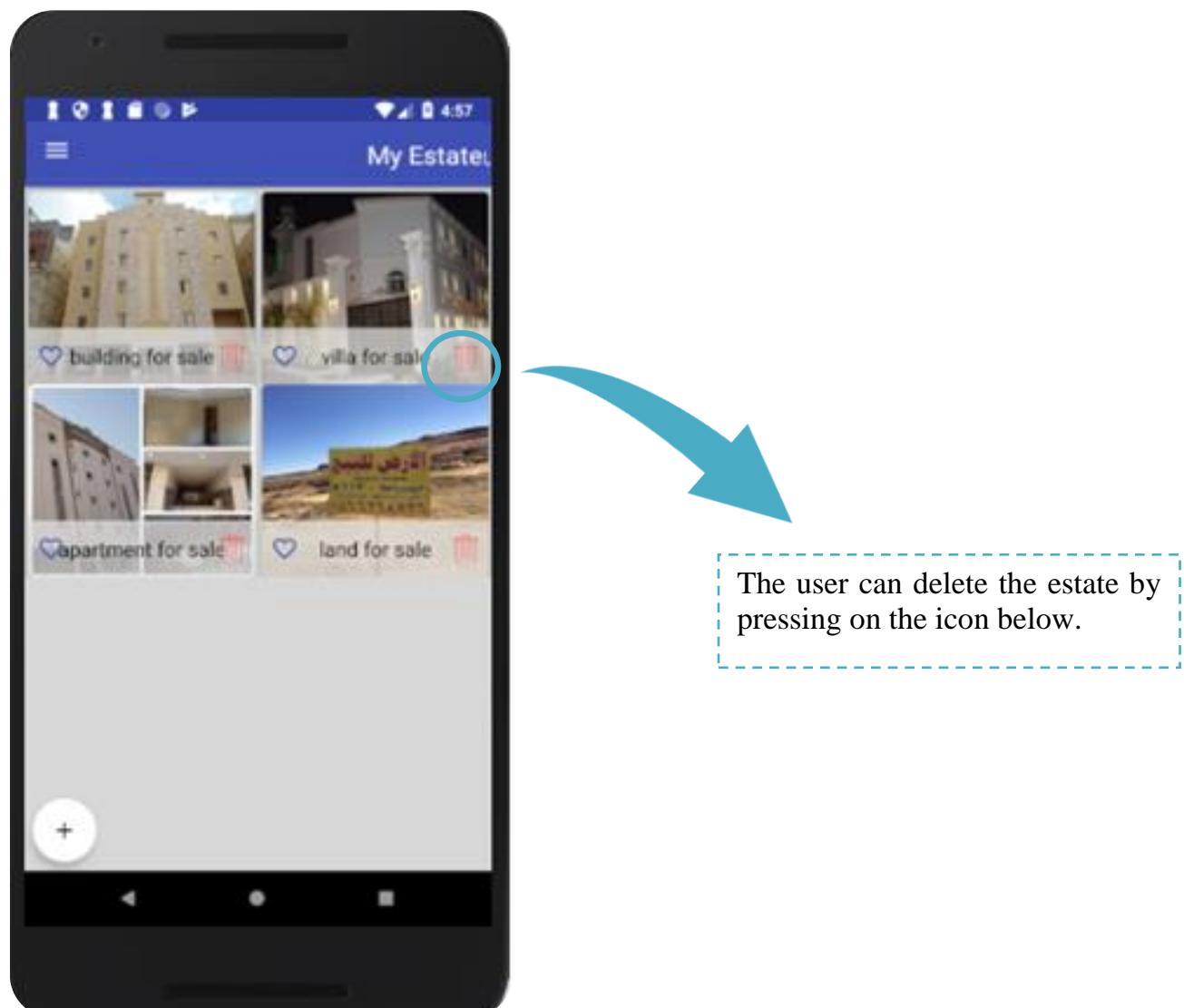


Figure 67: Delete button



- Favorite list:

FavouriteDB it is helper class extend from **SQLiteOpenHelper**. **SQLiteOpenHelper** It is very Lightweight database that comes with Android OS. We use it to store favorite list of users.

```
public class FavouriteDB extends SQLiteOpenHelper {  
    public static String DB_Name = "FavouriteDB.db";  
    public static int version = 1;  
  
    public FavouriteDB(Context context) { super(context, DB_Name, null, version); }
```

And must do override:

1-void *onCreate()*:

this method is override when the database is created for the first time. In this method we created the table (fav). The parameter **SQLiteDatabase db** it is database.

```
@Override  
public void onCreate(SQLiteDatabase db) {  
    db.execSQL("create table fav( Id INTEGER PRIMARY KEY AUTOINCREMENT , KeyEstate TEXT , Type TEXT, Name TEXT, " +  
              "EstateType TEXT , Image TEXT , Age TEXT , Floors TEXT , NoOfRooms TEXT , NoOfpaths TEXT , City TEXT , Price TEXT , NearSer  
    )
```

2-void *onUpgrade()*:

Called when the database needs to be upgraded. we use this method to drop tables, add tables, or do anything else it needs to upgrade. The parameters **SQLiteDatabase db** its database, **int oldVersion**: its old database version and **int newVersion** its new database version.

```
@Override  
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
    db.execSQL("DROP TABLE IF EXISTS fav ");  
    onCreate(db);  
}
```

```

public boolean insert_fav(String KeyEstate, String Type, String Name, String EstateType, String Image, String Age, String Floors,
String NoOfRooms, String NoOfpaths, String City, String Price, String NearServices, String Lat, String Lng, String KeyUser, String Collection)
{
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues contentValues = new ContentValues();
    contentValues.put("KeyEstate", KeyEstate);
    contentValues.put("Type", Type);
    contentValues.put("Name", Name);
    contentValues.put("EstateType", EstateType);
    contentValues.put("Image", Image);
    contentValues.put("Age", Age);
    contentValues.put("Floors", Floors);
    contentValues.put("NoOfRooms", NoOfRooms);
    contentValues.put("NoOfpaths", NoOfpaths);
    contentValues.put("City", City);
    contentValues.put("Price", Price);
    contentValues.put("NearServices", NearServices);
    contentValues.put("Lat", Lat);
    contentValues.put("Lng", Lng);
    contentValues.put("KeyUser", KeyUser);
    contentValues.put("Collection", Collection);
    long result = db.insert( table: "fav", nullColumnHack: null, contentValues);
    if (result == -1) {
        return false;
    } else {
        return true;
    }
}

```

And then create **Insert_fav()** to inserting data by getting writable instance (**getWritableDatabase()**) to write to database.

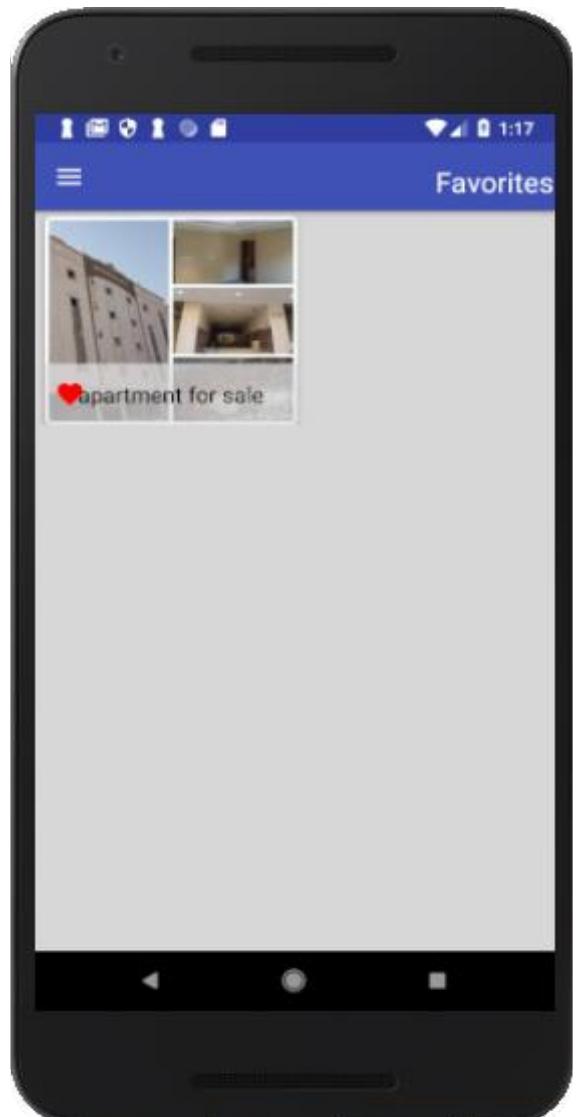


figure 68: favorite list

-Chat:

```
private void sendMessage() {
    String message = mChatMessageView.getText().toString();
    if(!TextUtils.isEmpty(message)) {
        String current_user_ref = "messages/" + mCurrentUserId + "/" + mChatUser;
        String chat_user_ref = "messages/" + mChatUser + "/" + mCurrentUserId;
        DatabaseReference user_message_push = mRootRef.child("messages")
            .child(mCurrentUserId).child(mChatUser).push();
        String push_id = user_message_push.getKey();
        Map messageMap = new HashMap();
        messageMap.put( k: "message", message);
        messageMap.put( k: "seen", v: false);
        messageMap.put( k: "type", v: "text");
        messageMap.put( k: "time", ServerValue.TIMESTAMP);
        messageMap.put( k: "from", mCurrentUserId);
        Map messageUserMap = new HashMap();
        messageUserMap.put( k: current_user_ref + "/" + push_id, messageMap);
        messageUserMap.put( k: chat_user_ref + "/" + push_id, messageMap);
        mChatMessageView.setText("");
        mRootRef.child("Chat").child(mCurrentUserId).child(mChatUser).child("seen").setValue(true);
        mRootRef.child("Chat").child(mCurrentUserId).child(mChatUser).child("timestamp").setValue(ServerValue.TIMESTAMP);
        mRootRef.child("Chat").child(mChatUser).child(mCurrentUserId).child("seen").setValue(false);
        mRootRef.child("Chat").child(mChatUser).child(mCurrentUserId).child("timestamp").setValue(ServerValue.TIMESTAMP);
        mRootRef.updateChildren(messageUserMap, (databaseError, databaseReference) -> {
            if(databaseError != null){
                Log.d( tag: "CHAT_LOG", databaseError.getMessage().toString());
            }
        });
    };
}
```

chat: The purpose of the chat in the app is to make it easier for the user to communicate with the advertiser and make agreements towards the estates he added.

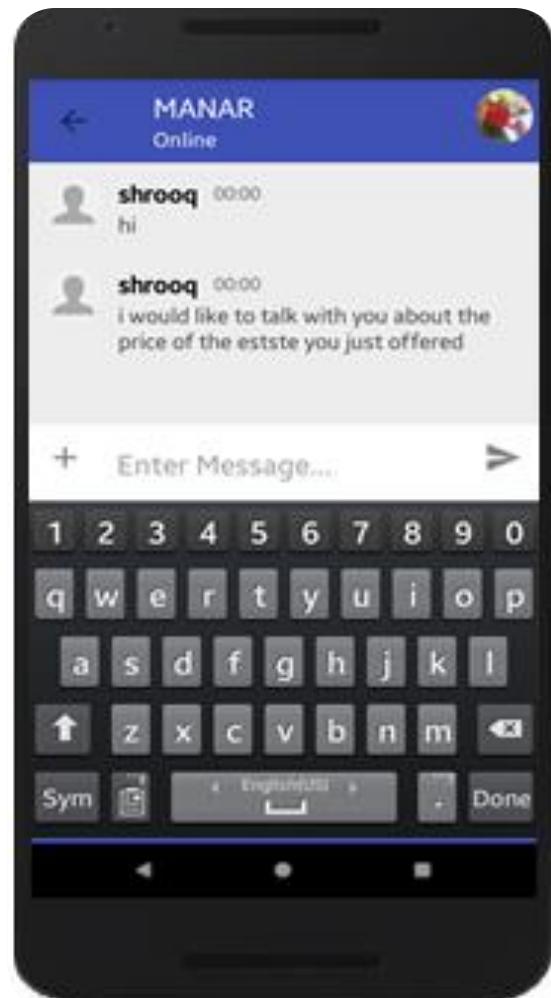


Figure69: chat

```

public static String getTimeAgo(long time, Context ctx) {
    if (time < 1000000000000L) {
        // if timestamp given in seconds, convert to millis
        time *= 1000;
    }

    long now = System.currentTimeMillis();
    if (time > now || time <= 0) {
        return null;
    }

    // TODO: localize
    final long diff = now - time;
    if (diff < MINUTE_MILLIS) {
        return "just now";
    } else if (diff < 2 * MINUTE_MILLIS) {
        return "a minute ago";
    } else if (diff < 50 * MINUTE_MILLIS) {
        return diff / MINUTE_MILLIS + " minutes ago";
    } else if (diff < 90 * MINUTE_MILLIS) {
        return "an hour ago";
    } else if (diff < 24 * HOUR_MILLIS) {
        return diff / HOUR_MILLIS + " hours ago";
    } else if (diff < 48 * HOUR_MILLIS) {
        return "yesterday";
    } else {
        return diff / DAY_MILLIS + " days ago";
    }
}

```

This code above for the chat function illustrate the time of the chat arrival for the sender and receiver .and, the period of time in between.[15]

Settings:

Contains the following pages (Edit profile, change password, Languages, Help center, Report a problem, Terms).

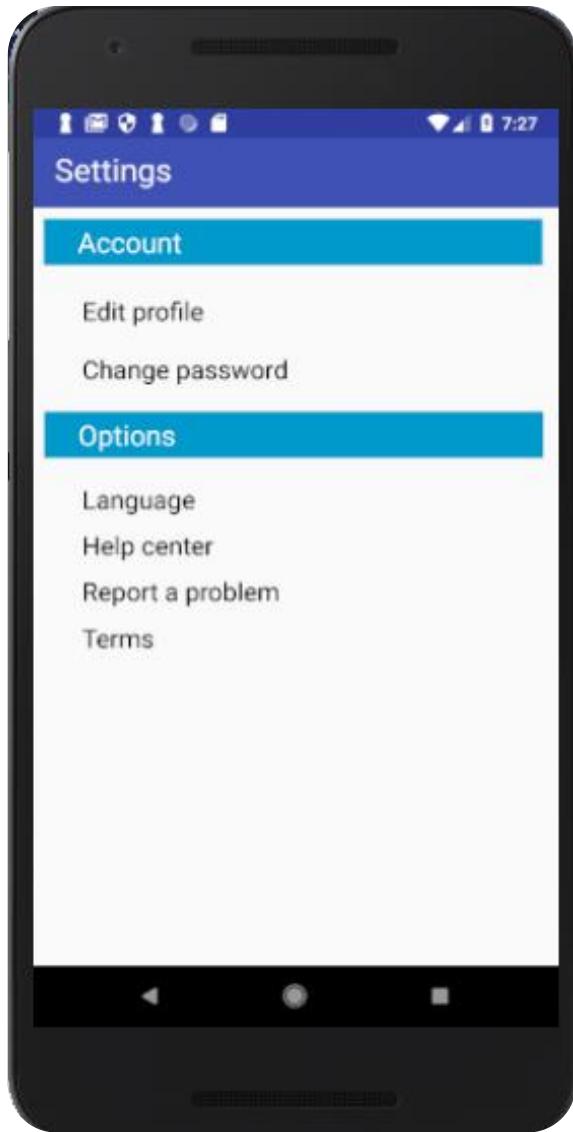


Figure 70: Settings

-Change password:



Figure 71: change password

The user will enter his registered email and click on change password and email will be sent to reset his password

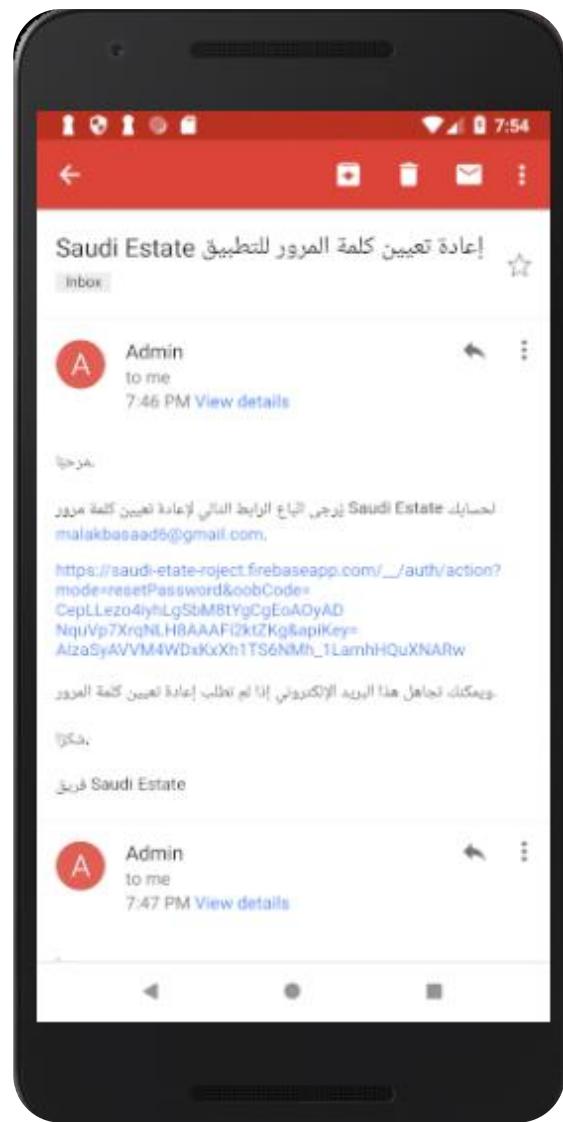


Figure 72: change password email

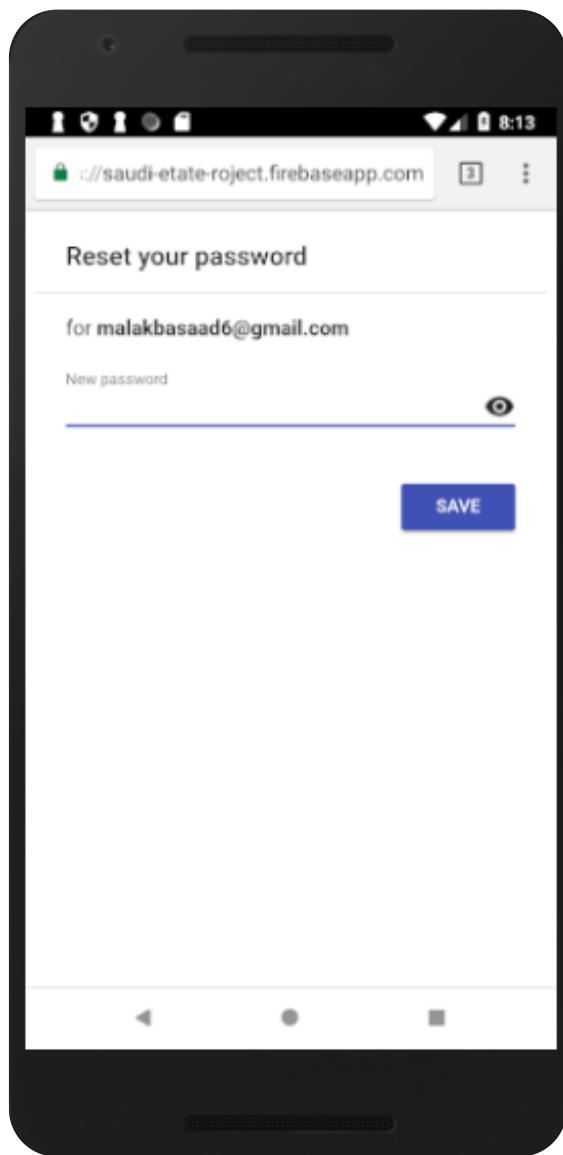


Figure 73: reset password



The user will enter a new password and click on save and the password will change. therefore, he can sign in with the new password.

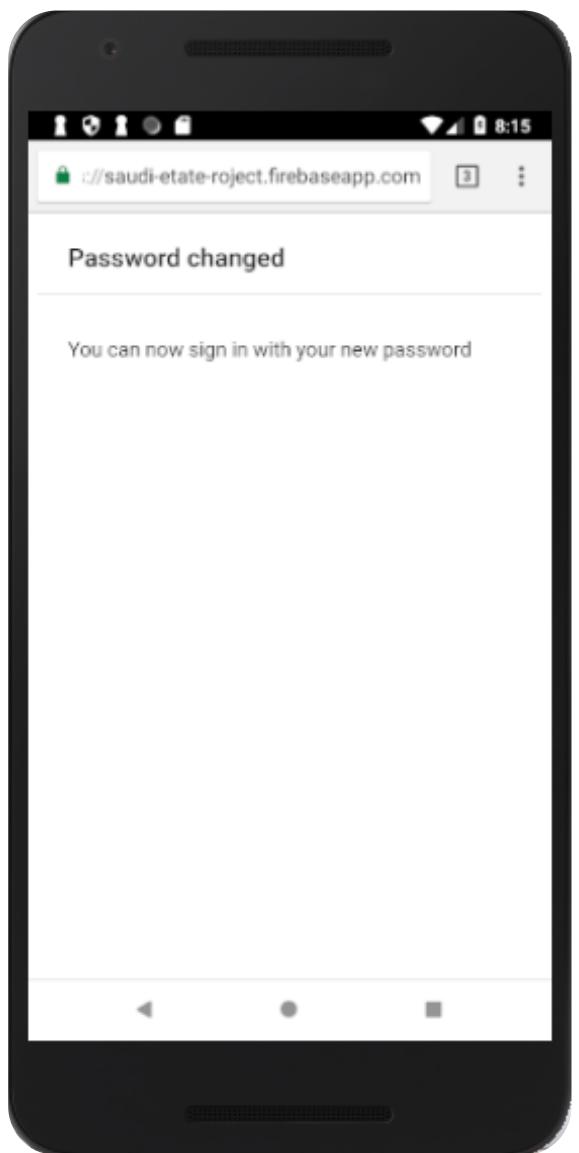


Figure 74: password changes

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_change_password);
    PasswordEmail = (EditText)findViewById(R.id.editText4);
    ChangePassword = (Button)findViewById(R.id.button3);
    firebaseAuth = FirebaseAuth.getInstance();
    ChangePassword.setOnClickListener((view) -> {
        String useremail = PasswordEmail.getText().toString().trim();
        if (useremail.equals("")){
            Toast.makeText(context: changePassword.this, text: "please enter your registered email",Toast.LENGTH_LONG).show();
        }else{firebaseAuth.sendPasswordResetEmail(useremail).addOnCompleteListener((task) -> {
            if(task.isSuccessful()){Toast.makeText(context: changePassword.this, text: "Password change email sent"
                ,Toast.LENGTH_SHORT).show();
                finish();
            }else{Toast.makeText(context: changePassword.this, text: "Error in sending Password change email "
                ,Toast.LENGTH_SHORT).show();
            }
        });
    });
}

```

firebaseAuth.sendPasswordResetEmail(useremail): firebase will check whether the entered email is registered the database or not.

then, **if else statement with the following:** if the task went successfully a reset email will be sent to the registered email else if he entered an unregistered email a message shown that there's error in sending password change email.

-Edit profile:

```
findViewById(R.id.btn_register).setOnClickListener((v) -> {
    progressDialog.show();
    if (uri!=null){
        StorageReference file_path = mStorage.child("Photo_Profile").child(uri.getLastPathSegment());
        file_path.putFile(uri).addOnSuccessListener((OnSuccessListener) (taskSnapshot) -> {
            final Uri do_uri = taskSnapshot.getDownloadUrl();

            progressDialog.dismiss();
            mUserCheckData.child("Fname").setValue(((EditText) findViewById(R.id.et_firstname)).getText().toString());
            mUserCheckData.child("Lname").setValue(((EditText) findViewById(R.id.et_lastname)).getText().toString());
            mUserCheckData.child("phone").setValue(((EditText) findViewById(R.id.et_phone)).getText().toString());
            mUserCheckData.child("image").setValue(do_uri);
            mUserCheckData.child("userName").setValue(((EditText) findViewById(R.id.et_username)).getText().toString());
            Toast.makeText(context: EditProfileActivity.this, text: "Saved Succesfully", Toast.LENGTH_SHORT).show();
        });
    }
});
```



Figure 75: Edit profile

The user can Edit and make changes in his profile by editing (image ,name , last name, Username, phone Number).

-Terms:

The terms of using the application the users must follow and accept.

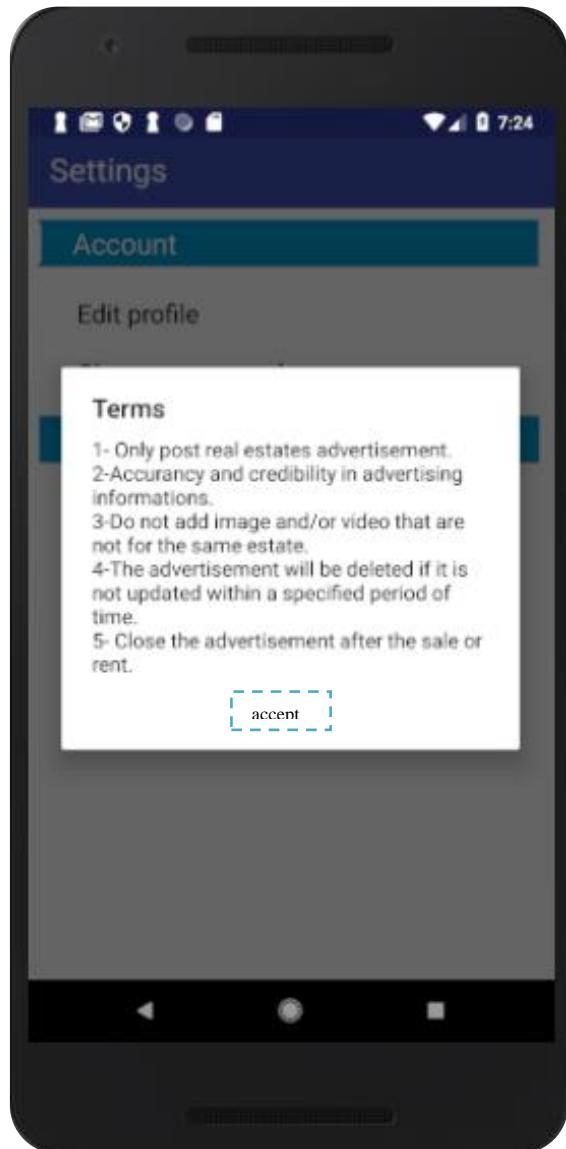


Figure 76: terms

5.2 Test

5.2.1 Test Tools and Environment

5.2.2 Testing Scenarios

5.2.2.1 Exception Handling

The user must fill out all required data if he has an account in our application.

If he does not activate his email via the link , this message will appear.

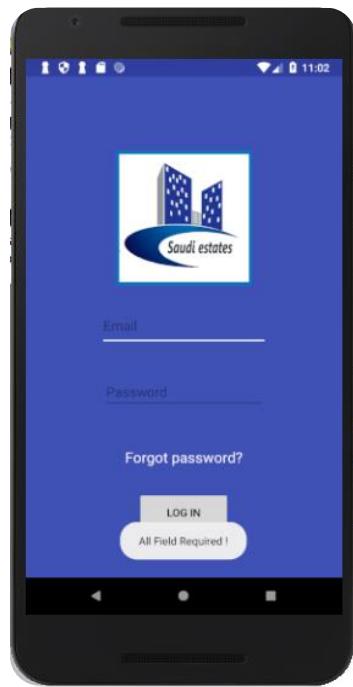


Figure 77: message1



Figure 78: message2

If user does not have account in our application.



Figure 79: message3

The user must fill out all the required data.

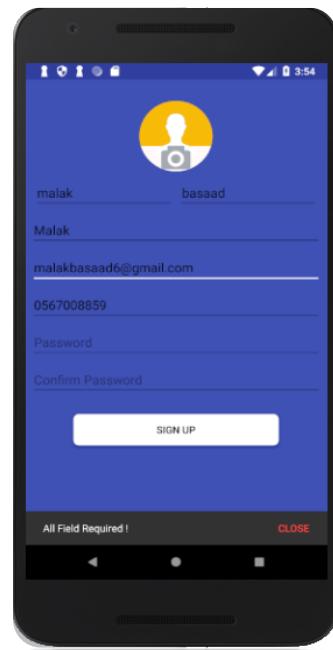


Figure 80: message4

If the confirmation password didn't match the entered password , this message will appear.

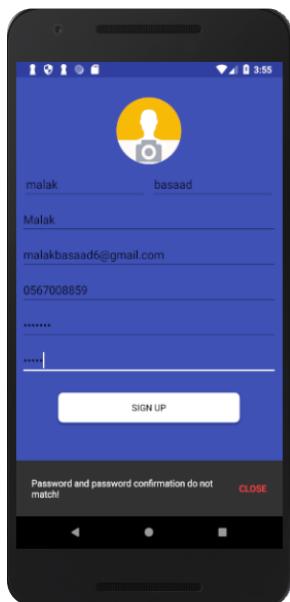


Figure 81: message5

The user must enter a registered email.

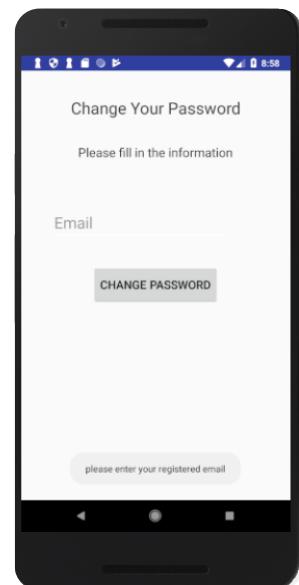


Figure 82: message6

This error message will appear if the user entered an unregistered email .

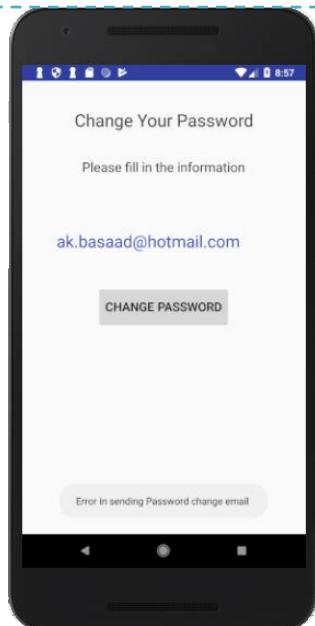


Figure 83: message7

Shows the advertiser to write the name of the Estate.

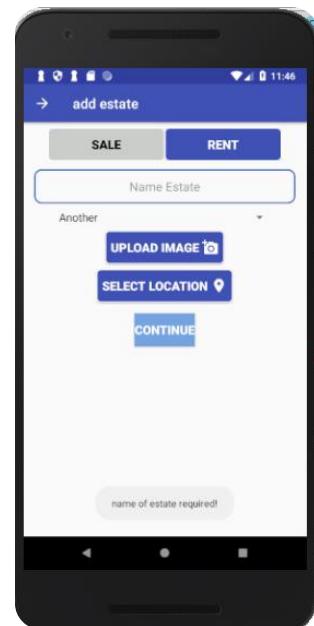


Figure 84: message8

The three messages appear to us unless they are filled in as required below.

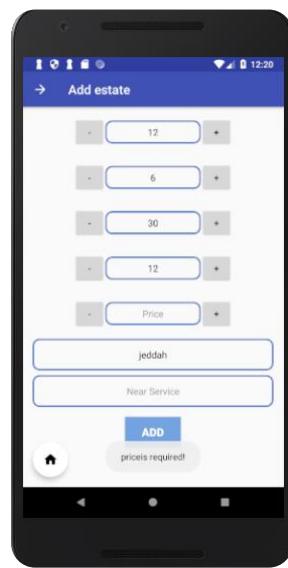


Figure 85: message9

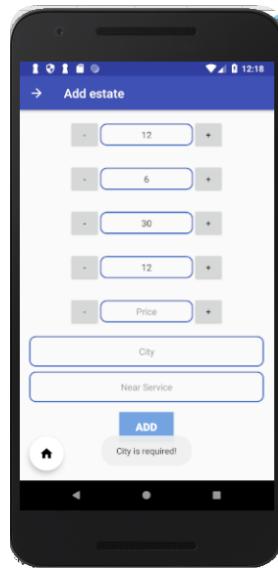


Figure 86: message10

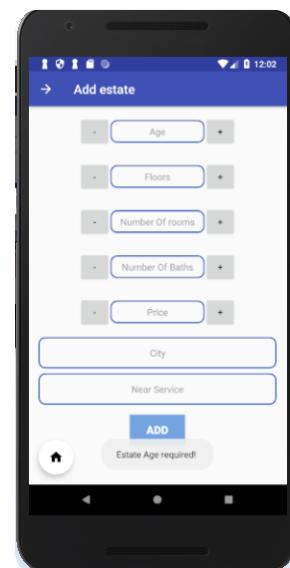


Figure 87: message 11

This message appears if the advertiser has uploaded an image of his Estate.

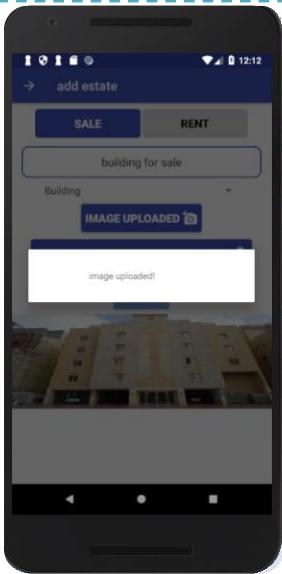


Figure88: message12

The advertiser must select the location of the Estate.

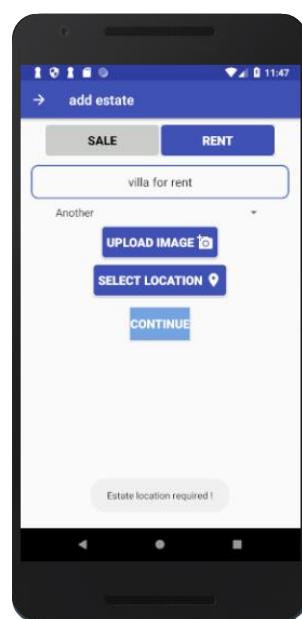


Figure89: message13

This message appears if the advertiser click on add button and has added all the required fields.

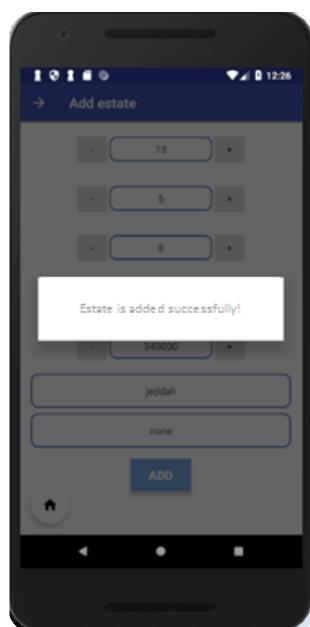


Figure90: message14

These messages appear when the estate is added to the Favorites by clicking on the heart marker and becomes red. Also, when you delete the estate from your favorites, click on the label I itself to become white.

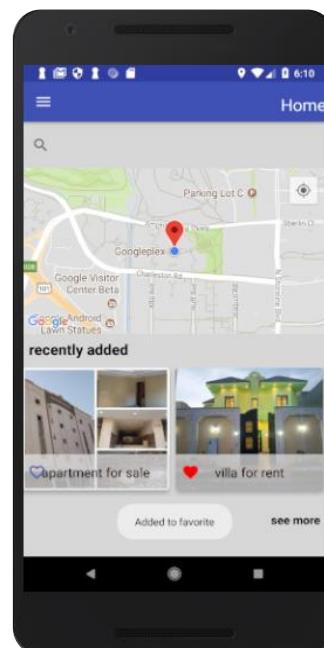


Figure91: message15

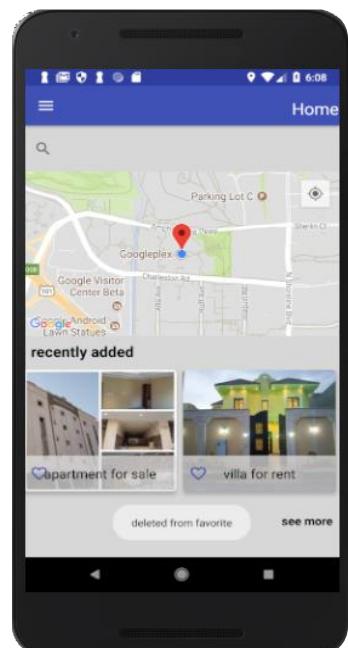


Figure92: message16

This message will appear when the user wants to delete an estate.

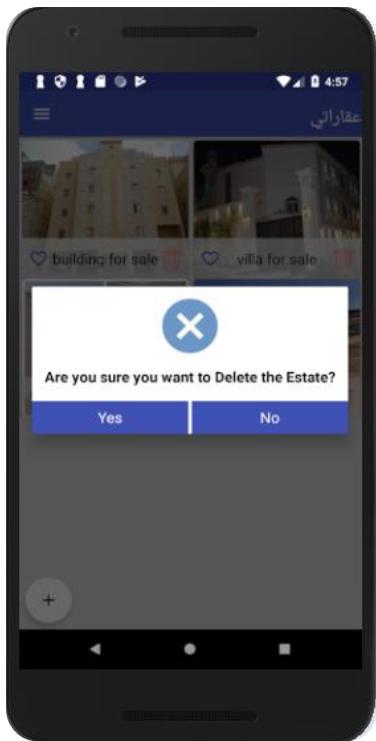


Figure93: message17

This message will appear when the user click on save button on edit profile to save his changes.

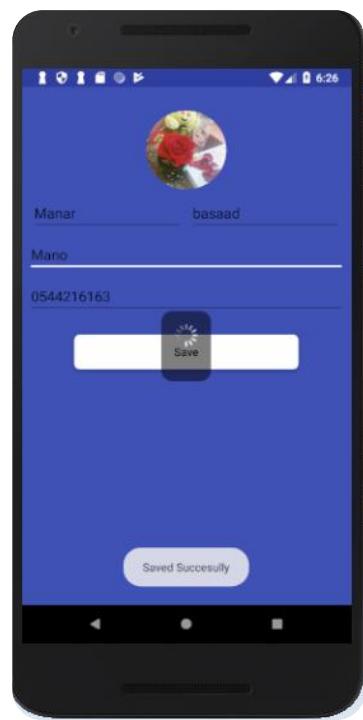


Figure94: message18

5.2.2.2 Completed Scenarios

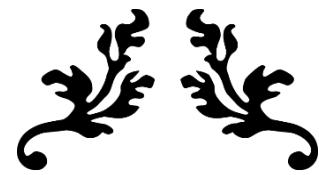
- Before anything the user must download the application, Then he can sign up or search for estates without sign up.
- After that the user can search for estates.
- if he doesn't find what he wants then he can sign up (if he didn't) then turn on notifications.
- if he found an estate then he can show (information, see it on map, show best traffic data and calculate distance)
- if this estate is suitable for him then he can contact advertiser (the user who offer his estate for rent or sell).
- if the deal is done then the estate will be deleted.
- if the deal is not done and if there a problem like (The advertiser is not honest and put the wrong information's) then the user can send a report, then the advertiser will put into a blacklist (we will count until 5 times he puts there then we will prevent him from offering any other announcement).
- also, user after sign in he can offer his estate for sell or rent.
the user can also make a favorite list which other users can see his likes under the estate.

5.3 Challenges

- During the implementation many challenges have been faced like:
- Run our application in appropriate emulator.
- Determine the location of estates.
- How handle with adapter and recyclerview (for estate and favorite list) and fragment.
- Retrieve data from firebase.
- Create simple database from SQLite.
- How to get message form sender and receiver in chat.

5.4 Conclusion

Most of the applications in the present time contribute to providing comfort to the users so the users Can browse, buy and sell what he/ she want at any time. We hope that our app is useful for everyone who uses it and we hope to develop it to support IOS.



Appendix A



Details tasks:

Roles	Responsibilities
Malak Muhammad basaad.	<p>CHAPTER ONE:</p> <ul style="list-style-type: none"> • literature review. • expected outcome. <p>CHAPTER TWO:</p> <ul style="list-style-type: none"> • data analysis. • data flow diagrams. • Proposed solutions and alternative solutions. <p>CHAPTER THREE:</p> <ul style="list-style-type: none"> • gather information about firebase. <p>CHAPTER FOUR:</p> <ul style="list-style-type: none"> • design interfaces with description. <p>CHAPTER FIVE:</p> <ul style="list-style-type: none"> • explain user and admin interface in report. • implemented user's registration, home page and navigation drawer.

Khulood daifallah Al-Otaibi.

CHAPTER ONE:

- introduction.
- literature review.
- overview of the document.

CHAPTER TWO:

- data analysis.
- class diagram.

CHAPTER THREE:

- Database constraints.
- Hardware and software environment.
- project management strategies.

	<p>CHAPTER FOUR:</p> <ul style="list-style-type: none"> • design interfaces with description. <p>CHAPTER FIVE:</p> <ul style="list-style-type: none"> • explain connection part in report. • explain user and admin interface in report. • connect the app with firebase. • make the app support both languages (English and Arabic). • Implement "My Estate" and notifications.
Shoroog balgaith Al-bargi.	<p>CHAPTER ONE:</p> <ul style="list-style-type: none"> • purpose of this document. • objective of the project. <p>CHAPTER TWO:</p> <ul style="list-style-type: none"> • Use Case Diagram(UCD) and details. <p>CHAPTER THREE:</p> <ul style="list-style-type: none"> • future enhancements/plans. <p>CHAPTER FOUR:</p> <ul style="list-style-type: none"> • design interfaces with description. <p>CHAPTER FIVE:</p> <ul style="list-style-type: none"> • explain user and admin interface in report. • implement setting and navigation drawer.

Doa'a Turkistani.	<p>CHAPTER ONE:</p> <ul style="list-style-type: none"> • purpose of the project. • method/approach. <p>CHAPTER TWO:</p> <ul style="list-style-type: none"> • questionnaire. • activity diagram. • ERD diagram. <p>CHAPTER THREE:</p> <ul style="list-style-type: none"> • development method. <p>CHAPTER FOUR:</p> <ul style="list-style-type: none"> • design interfaces with description. <p>CHAPTER FIVE:</p> <ul style="list-style-type: none"> • explain completed scenario in report. • implement chat and favorite list.
Nebras talat sindi.	<p>CHAPTER ONE:</p> <ul style="list-style-type: none"> • project aims.

	<p>CHAPTER TWO:</p> <ul style="list-style-type: none">• data analysis.• sequence diagram. <p>CHAPTER THREE:</p> <ul style="list-style-type: none">• Algorithm to be used. <p>CHAPTER FOUR:</p> <ul style="list-style-type: none">• major modules, sub modules. <p>CHAPTER FIVE:</p> <ul style="list-style-type: none">• explain exception handling in report.• implement "My Estate" and search.
--	--

Table28: Details tasks

Gantt chart (Semester 1):

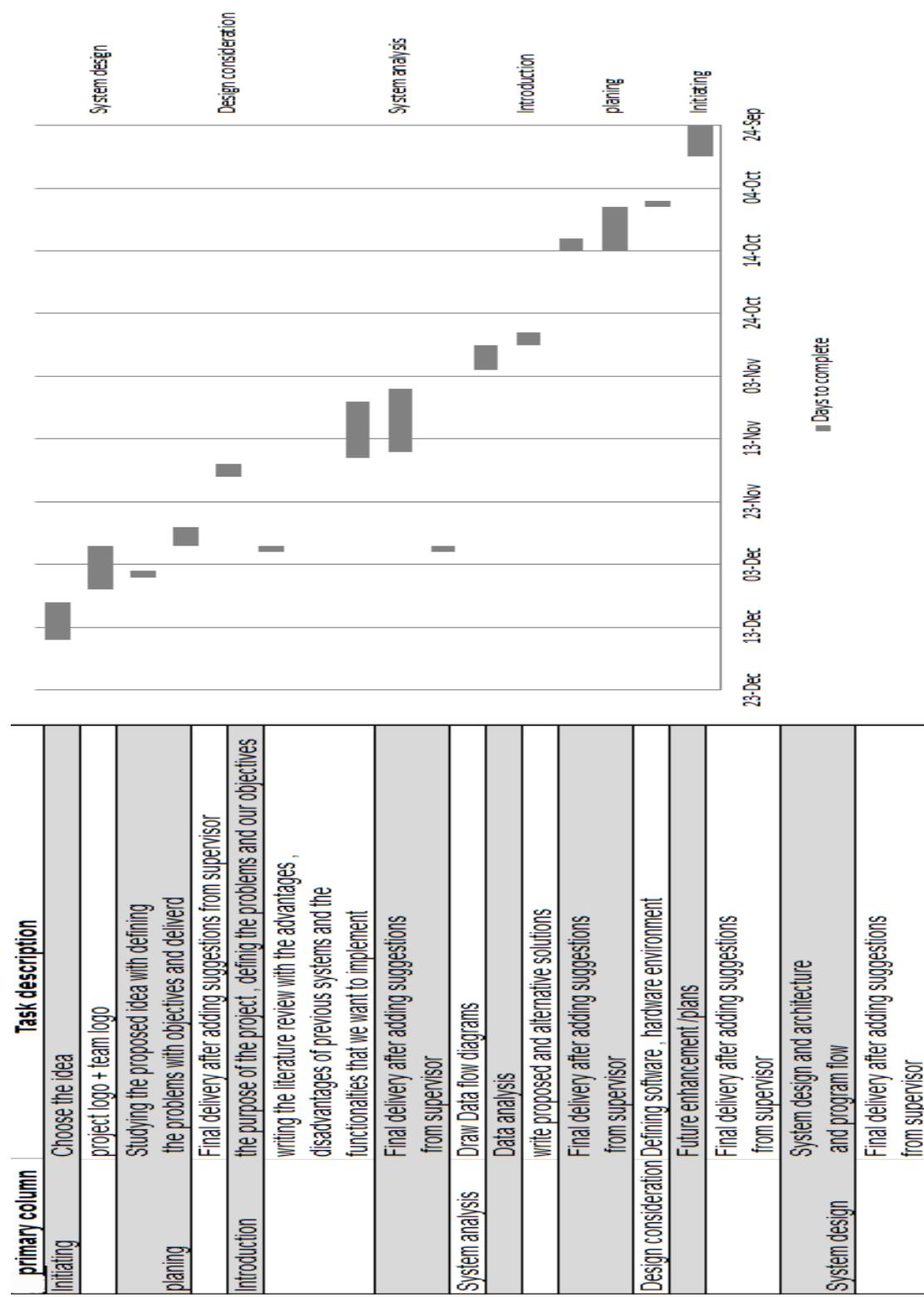


Table29: project timeline1



Appendix B



Gantt chart (Semester 2):

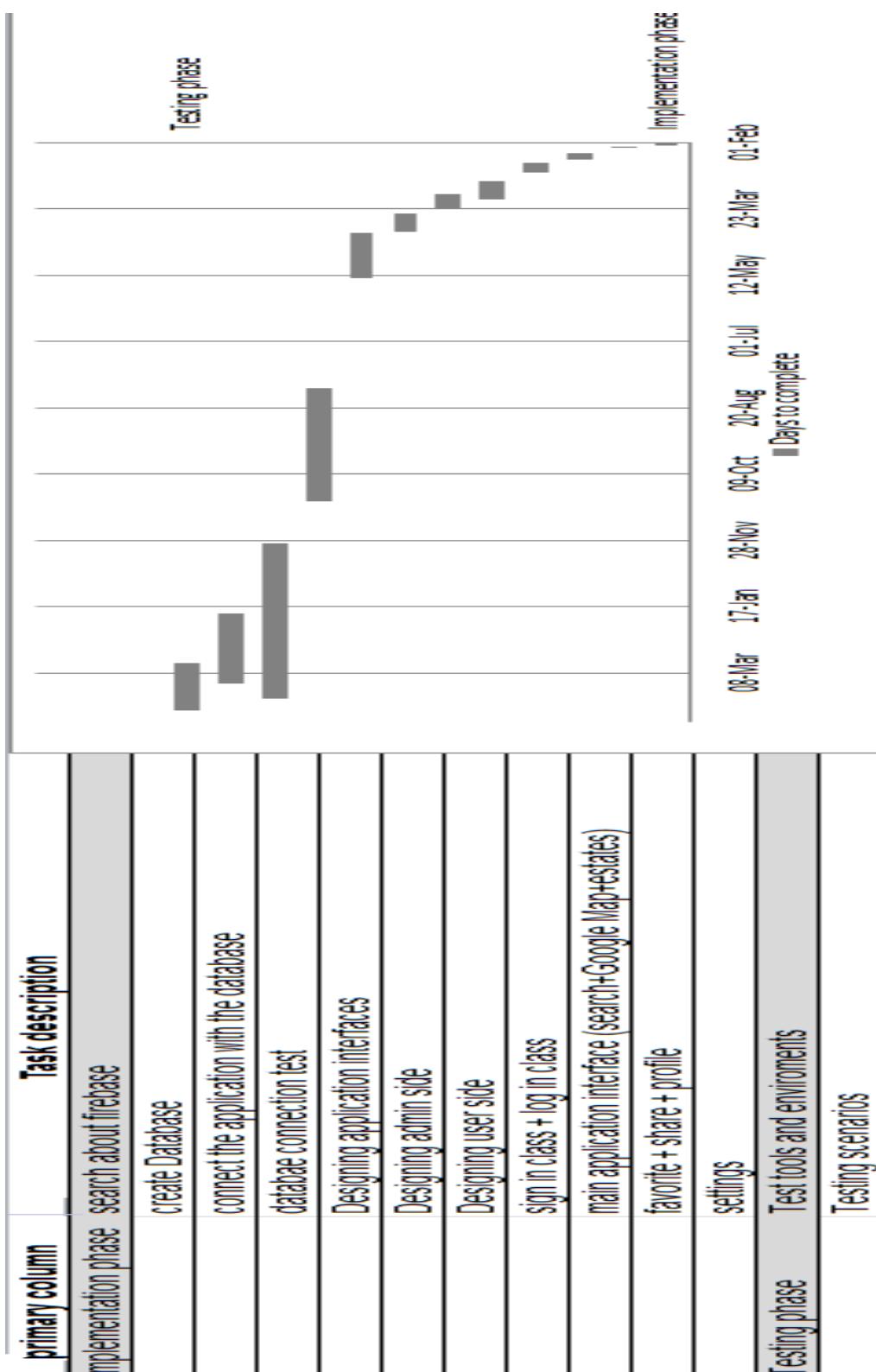


Table30: project timeline 2



Appendix C



Saudi estates

Malak M.Basaad – Khulood D.al-Otaibi – Shoroog
 B.Al-Bargi – Doa'a M.Turkistani – Nebras T.Sendi



Supervisor: Dr. Muhammed Arif

INTRODUCTION

Today Real Estate Industry has been growing rapidly and mobile app plays a very important role in the real estate company at this time.

Our project falls under the part: information system, it is an application so we will use the programming and database design. Thus, we aim to develop an application able to create desired interactivity to encourage users to explore the best possible results out of their searches with minimum efforts in very less time.

The very first use of these real estate's mobile apps is that we give users a direct channel to reach them and the users will be able to use these apps at any time right the clock. In addition, feature visualization of the estate with our interactive maps and the users will become informed on the estate and area demographics and most of the real estate applications display their estates attached with pictures, video, and description.

PROBLEM

Difficult people resort to real estate offices.

Waste time and effort.

The cost of some real estate offices.

Difficult in informing or receiving real estates offers and finding the right real estate in the right place.

CONCLUSION

Most of the applications in the present time contribute to providing comfort to the users so the users can browser , buy and sell what he / she want at any time . We hope that our app is useful for everyone who uses it and we hope to develop it to support IOS.

FEATURES

There are some of the main features of our project:

- Search based on specifications for the estate such as specific city, type of estate (like land, villa, apartment, and building), cost of the estate, number of floors, rooms, baths..etc.).
- Search by location with near me feature that enables the user to discover real estates around him.
- Support travel flow data from Google Map to find the best location.
- Allow the user to add an estate and offer it for sale or rent attached with the announced estate picture, video, and audio of the description and a map of the estate location.
- The app has a notification feature, it will send notifications about the recent estates.
- The administrator will receive the report from the user and start verifying it by taking estate ID and decide whether it followed the terms of our application or not, to take the proper action for the advertiser like add him to black list or decline the user's report.
- The application will delete the estate automatically when the advertiser does not update the estate after passing period amount of time.
- Share feature that allows a user to share the estate with social media applications (for example: WhatsApp, Facebook and twitter ...etc).
- Allow users to make a list of favorite and collection estates.
- The application supports both Arabic and English languages according to the user's choice.

OUTCOME

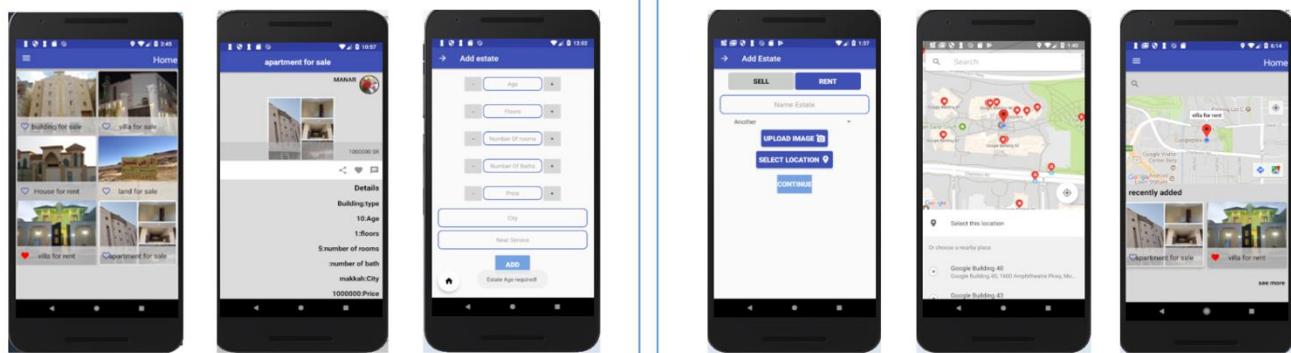


Figure95 : Project Poster

References:

- [1] <https://fstoppers.com/apps/matterport-new-age-real-estate-137707>
- [2] Homesnap Real Estate& Rentals, created by Homesnap, available on google play and app store.
- [3] Lamudi available on app store.
- [4] Housing-Real Estate & Property available on Google play.
- [5] Redfin available on Google play and app store.
- [6] Houses.com available on app store &Android.
- [7] https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm
- [8] <https://courses.telegraph.co.uk/article-details/86/waterfall-project-management-explained/>
- [9] <https://howtofirebase.com/what-is-firebase-fcb8614ba442>
- [10]<https://www.google.com.sa/search?:whatisfirebase>
- [11] <https://firebase.google.com/docs/?authuser=1>
- [12] [we learn some basics android programming from this PDF](#)
- [13] <https://developers.google.com/maps/documentation/android-api/map-with-marker>
- [14] <https://developer.android.com/training/sharing/send.html>
- [15] <https://stackoverflow.com/questions/13018550/time-since-ago-library-for-android-java>