Recommendation

This is certified that Umme Nayema Rownak (CSE 00505225) student of Port City International University under the Department of Computer Science & Engineering had carried out the project on “Real Estate” successfully under my supervision.

Signature of Supervisor

…….

(Mrs. Monisha Biswas)

Lecturer, Dept. of CSE, PCIU

Statement of originality

I, hereby, declare that the work presented in this project is the outcome of the investigation performed by me under the supervision of Mrs. Monisha Biswas, Lecturer, Department of Computer Science and Engineering, Port City International University, South Khulshi, Chittagong, Bangladesh. I also declare that no part of this project and thereof has been or is being submitted elsewhere for the award of any degree or diploma.

……..... ……

Signature of Supervisor Signature of Candidate

Date: Date:

Dedication

I would like to dedicate this project to

Our country,

My Parents

And

My Honorable Teachers.

Acknowledgements

All praises are due to Allah the Almighty, May peace and blessing are upon his servant and Prophet Mohammad (P.B.U.H), his family, companions and those who follow. We are able to complete this project for Allah's kind blessing and mercy.

I deem it a pleasure to acknowledgement my sense of gratitude to my project guide lecturer **Mrs. Monisha Biswas** under whom I have carried out the project work. Her incisive and objective guidance and timely advice encouraged me with constant flow of energy to continue the work. Moreover, she inspired me with his valuable suggestions to successfully complete the project work.

Finally, I must say that no height is ever achieved without some sacrifices made at some end it is here where I own my special debt to my parents and my friends for showing their generous love and care throughout the entire period of time.

Umme Nayema Rownak

ID: CSE 00505225 (Day)

Port City International University

Abstract

Nowadays, one of the common issue people is facing is to find a suitable house or place to live. Or they have the issue of pricing of properties. And more of it they need to paid a good amount to the media. On the other hand if people want to sell or give a rent of their flats or lands they face many difficulties to find a suitable buyer for their flats and lands.

So, I want to develop an application that will provide information about flats, land areas. By this application, anyone can buy and sell their flats, dwell house, lands etc. Here the user can easily verify the properties before buying or take a rent. Buyer can easily communicate with the seller and can negotiate. Above all this application is so reliable to user. Our main concept is giving best & quick result to user.

Table of Contents

List of figures	

CHAPTER-1

INTRODUCTION

1.1 INTRODUCTION

Today our world is digital. Everyone wants to get everything in easy way. Now in this digital world, android application is very familiar and become one of the popular platforms for the people.

For this fast world Real estate business is become common in our society, but however it is not secure in our country. To cope up this problem Our Application is secure for both buyers and sellers. Anyone who is registered can buy or sell their property here easily. Anyone who is registered can easily contact with each other by messaging.

1.2 OBJECTIVE

Main objective of this report is to fulfill the requirement how to develop an android app named "Real Estate" for android operating system. To accomplish the objective, there are several steps in the procedure that have been followed such as:

- User need to register in this app as a buyer or seller.
- Seller can advertise their property for selling.
- Seller can add, manage their properties
- Buyer can buy property from here.
- Buyer can also buy property by searching their desirable area or city.
- Admin can add user, manage user.
- Seller and Buyer can contact with each other by messaging through this app.

1.3 MOTIVATION

I have chosen to develop android app based on real estate properties because of some motivational factors that encouraged me. I understood the importance of gathering knowledge about properties by the different users. For instance, buyer and seller both would get help from this real estate application. In addition, it would be helpful for all general people to know details about their properties including the area the property is situated, the price, the size of the property and also can take loan for the property. So I got motivation to develop an android app to

provide information about properties where people can easily buy or sell their properties from a trustable site without paying extra charge.

1.4 SCOPE OF THE PROJECT

This App is capable of accepting, validating and incorporating data. It enables user to manipulate data in rational time. We have used SQLite in the back end which is a relational database.

It is fully optimized and can handle ambiguous as well as reduced redundancy of data. It is not vulnerable to SQL injection, Brute Force Attack and other security hazards.

- ✓ Scope
- ✓ Importance
- ✓ Timely Helpful
- ✓ Robust

CHAPTER-2

LITERATURE REVIEW

2.1 RELATED WORKS

In Google play store there are many apps available about real estate, like

- Cover art realestate.com.au - Buy, Rent & Sell Property,
- Real Estate,
- Redfin Real Estate,
- Real Estate Dictionary,
- Housing Property Search & Real Estate App
- Realestate
- Real Estate Marketing Ideas 2018
- 99acres Real Estate & Property
- Real Estate Broker CRM Lite
- House.com.mm Property Buy/Rent Lamudi
- Cover art Real Estate sale & rent Trovit,
- Trulia Real Estate: Search Homes for Sale & Rent, and so on.

2.2 LIMITATION OF PREVIOUS WORKS

In many web portals and Google play store there are many apps like my real app but there have lots of limitation also.

- The most important feature of a Real Estate App is having registered users. Many of the app anyone can enter his/her property. Where in our app every user who wants to sell their property need to registered in the app first.
- In most of the app there is no step is taken for the fake buyer or seller. Any fake buyer and seller can register in the app and can enter illegal properties. But in my app if admin found any fake buyer and seller admin can immediately terminate the user.
- Property Image is a key attribute to know about a property. In many app Fake Images has been used. Where in my app if admin found any fake image can immediately delete the property.

- In other apps there is no options kept for the communication between buyer and seller. Some apps given an option to call the seller. But in my app there kept an option for messaging. User just needs to have an internet connection.
- Loan is playing a vital role in my app. Where Other Apps there is no such options.
- There are different types of real estate app in the Google play store. Most of them are working only with the flats or apartments. But in my app I tried to work almost all the field of real estate.

In all the period of my work I always tried to take motivations from other works. I tried to find out their limitations and tried to figure out the problems. Most of the issues I have been trying solved in my app.

CHAPTER-3

PROPOSED APPLICATION

3.1 PROBLEM DEFINITION

As property is plays vital role in human life. Every person wants a decent and comfortable price according their monthly income. But whenever someone wants to buy or sell their property or want to give rent they need to face many difficulties to find a decent buyer or seller or to have all kinds of information's about property. So I got motivation to develop an android app to provide complete information about the property and give a platform where buyer and seller can easily communicate without the fear of having a fake property.

3.2 PROJECT OVERVIEW

Generally, it is an informative application about properties. In this project, it is tried to accumulate almost all information regarding properties used for rent or sale. And it is also tried to keep the app from any kind of illegal works

For example:

- User need to register in the App as a Buyer or Seller. Without registration Seller cannot enter a new Property.
- Admin can see the entire user list. Admin can manage the users. Admin can immediately terminate any fake user if admin found such users.
- Property Type is an important feature to distinguish Property Land or Flat.
- Property Address is a key point of the App. User can select from the dropdown
- Complete Address is the key to found a property. Seller also needed to enter a Complete Address of a Property so that Buyer can easily locate the Property
- Property Size is needed to enter a Property
- Property Price is needed to enter a Property
- Seller need to give complete details about the Property. It could be the about security system, playground details or any important details that need to be included.

- Another important feature is Loan Percentage in the App. Now a days Loan is playing a vital role to buy anything. From differentiate my App from the others Loan is included in the App.
- Image is the key feature of the App. Seller need to use a proper Image for the Property. Otherwise Admin can remove any property that contains fake Image.
- Seller also can manage his Property. Seller can remove any Property after its been sold or Rented
- Communication is another important in the App. Buyer and Seller can communicate with each other. Buyer can message Seller through the App including Desire Property. Seller can reply to the Buyer. The message will kept confidential only Buyer and Seller can see the message.

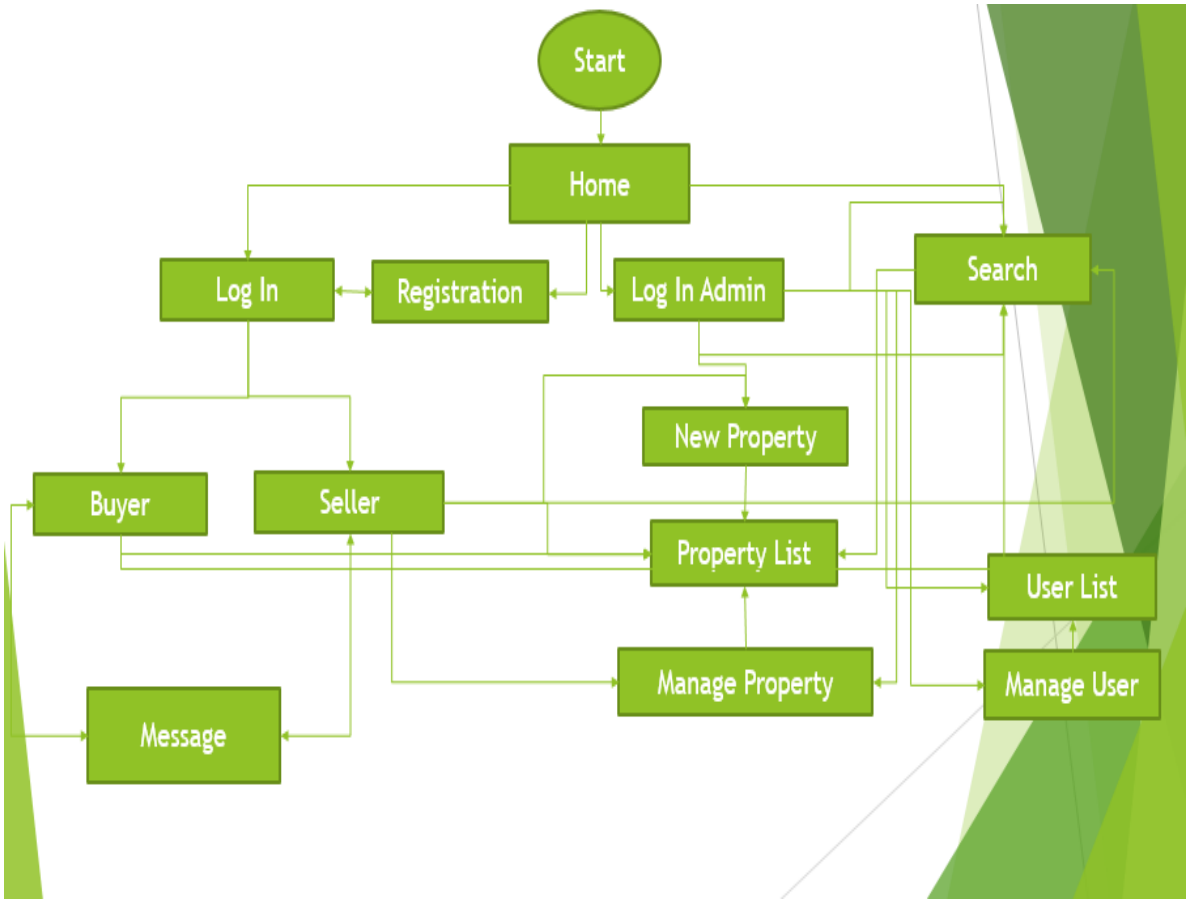


Fig. 3.1 Real Estate Project Overview

3.3 DEVELOPED SYSTEM

There are three Users for the App

- Admin
- Buyer and
- Seller

The Complete Process

- User need to register in the App as Buyer or Seller
- App has a Admin
- The attributes that are used for Enter a New Property are:
    - ✓ Property Address
    - ✓ Property Complete Address
    - ✓ Property Size
    - ✓ Property Type
    - ✓ Property Price
    - ✓ The Reason Property Used for Rent/ Sell
    - ✓ Loan Percentage
    - ✓ Property Image
- All of the fields should contain correct Information with a real Image.
- Admin will always have a close look on the given Properties. If Admin found any Fake Property Admin can remove it immediately.
- Seller can Add & Manage his/her property. If any changes need to make Seller can easily do that. Seller can change the Image also if needed. Or Property is being already Sold Seller can remove it.
- Admin will see the entire User List
- Admin can remove any unwanted or fake user.
- After Log in Buyer and Seller can communicate with other. If Buyer any Property interesting Buyer can message Seller and if Buyer feels interested he/she can reply to the Buyer. Entire Message will be confidential. Only Buyer or Seller can see the Message.
- Search option is also developed in the App. User Need to select some options that are given in the UI and can have the desirable search Property

3.4 E-R Diagram

An entity relationship model, also called an entity-relationship (ER) diagram, is a graphical representation of entities and their relationships to each other.

The diagram below shows that, there have many entities like Property, User, and Message. Each of these entities has many attributes. Every entity has primary key that showed here by underlining the name of attribute. However, here userid is used as a foreign key for property, message entities and property id for message entity.

Each User entity that contains buyer usertype can have many properties and their relationship is one-to-many. By the same way Message entity related to other 2 entity User and Property and their relationship is many-to-many relationships.

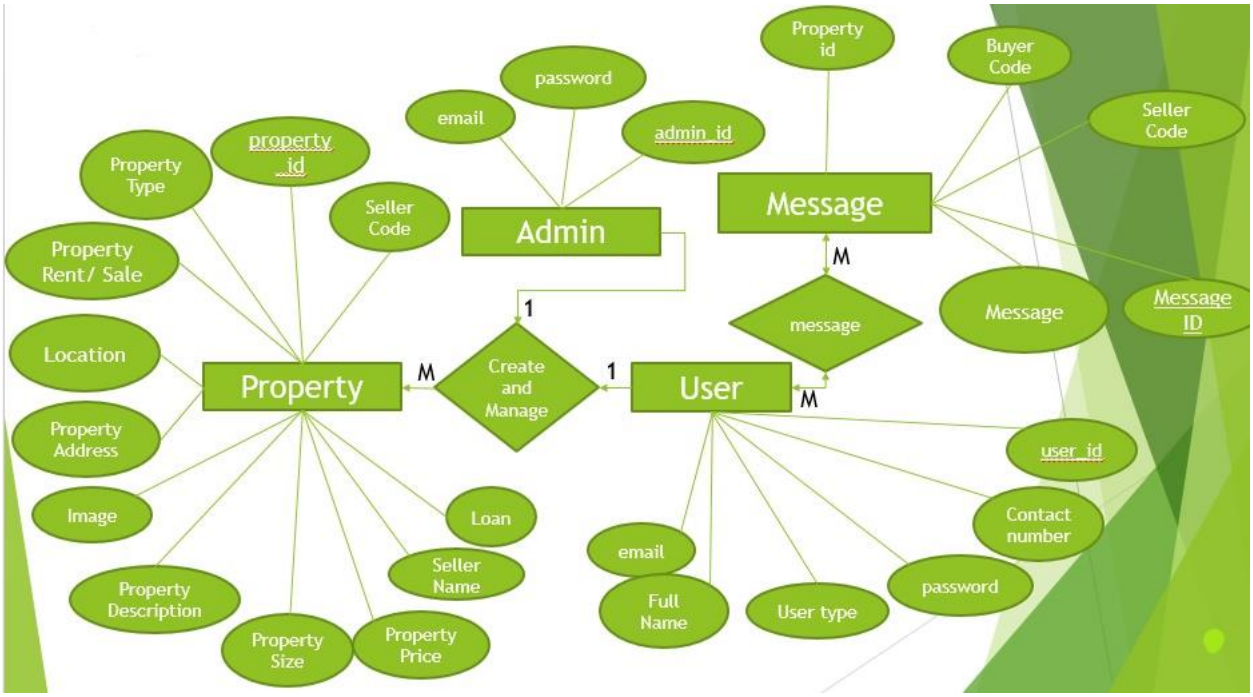


Fig. 3.1 E-R Diagram

CHAPTER-4

SOFTWARE REQUIREMENT SPECIFICATION

4.1 SOFTWARE REQUIREMENT SPECIFICTION

A software requirements specification (SRS) is a description of a software system to be developed. SRS (also known as a Software Requirements Specification) is a document or set of documentation that describes the features and behavior of a system or software application.

- **Property List:** The most informative part of the App is the Property List where anyone can traverse thousands of Properties. And if he/she found the desired one he/she need to register in the App and can be the owner of desired Property.
- **Register in the App**: Anyone can see the property but for further activities like communicate with seller user need to register in the app and authorized by admin
- **User List:** User List is a confidential List. Only Admin can see the entire List. Admin can have a closed look on the User who are using the App and can terminate any Fake User.
- **Effective search option:** My App has an effective search option where user can easily search through property address, price, size, loan percentage. User Need to select the options that are given in the UI and can have the desirable search Property.
- **Adding & Managing Property**: Adding a Property is the main task done by the Seller. Seller can add any type of Property. Seller will also manage his Property. Seller can remove any Property after its been sold or Rented. Some Important Features that's been used for Property are:
  - ✓ **Image of property:** Image is the key feature of the App. Seller need to use a proper Image for the Property. Otherwise Admin can remove any property that contains fake Image Every property image is been shown in property list and property detail page
  - ✓ **Loan:** Another important feature is Loan Percentage in the App. Buyer must have select loan percentage from the dropdown from the UI of New Property
  - ✓ **Property Description:** All property must have property details
  - ✓ **Property Price and Size:** Property Price and size must be included

- ✓ **Property Type:** Property Type is an important feature to distinguish Property Land or Flat. Seller need to select the Property Type before submitting a New Property.
- ✓ **Property Address:** Complete Address is the key to found a property. Seller also needed to enter a Complete Address of a Property so that Buyer can easily locate the Property

- **Communicate:** After Log in Buyer and Seller can communicate with other. If Buyer any Property interesting Buyer can message Seller and if Buyer feels interested he/she can reply to the Buyer. Entire Message will be confidential. Only Buyer or Seller can see the Message.

CHAPTER-5
SYSTEM ANALYSIS

5.1 INTRODUCTION

System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. System design is a process of planning a new business system or replacing an existing system by defining its components or modules to satisfy the specific requirements. Analysis specifies what the system should do. System Design focuses on how to accomplish the objective of the system.

5.2 PROPERTY LIST

In system, there should be property list which is most important task on the application. In “real estate” there should have a list of property and which will be using for finding properties systematically by descending order. There have to be added good number of properties where users can find his/her desired property.

5.3 EFFECTIVE SEARCH FUNCTION

Search function is essential for this Real Estate app. User can search a property by its price, size, loan percentage, address. If one of the data matches with the property details all of them shown as a list. User can choose his desire property from the search option. This is very effective and user will be satisfied with this search function.

5.4 MESSAGING FEATURE

Messaging is one of the important features of the app. If buyer choose a property and want more details or want to negotiate about the price he/she can easily communicate with the seller. Entire Message will be confidential. Only Buyer or Seller can see the Message.

5.5 IMAGE OF THE PROPERTY

It is very important to know how the property is look like. A property can be a flat, land or dwell house. If image of the property added in the application, it will be great. Before buying or taking rent a property, users can easily watch how the property is look like.

5.6 PROPERTY DETAILS

Property details can be included as a feature like areas, property size, property price, loan percentage and property details etc. Most of the people want to know this type of information about the property.

5.7 USER LIST

Only admin can see all the users whom are registered in the app. If admin found something wrong about the user admin terminate the user.

5.8 ADDING & MANAGING PROPERTIES

Seller can add new properties and manage the properties. If there is some changes occur seller can easily change them. And after property is bought by buyer seller can remove the property.

CHAPTER-6

SOFTWARE DESCRIPTION

6.1 ANDROID STUDIO

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Instant Run to push changes to your running app without building a new APK
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems
- C++ and NDK support
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine

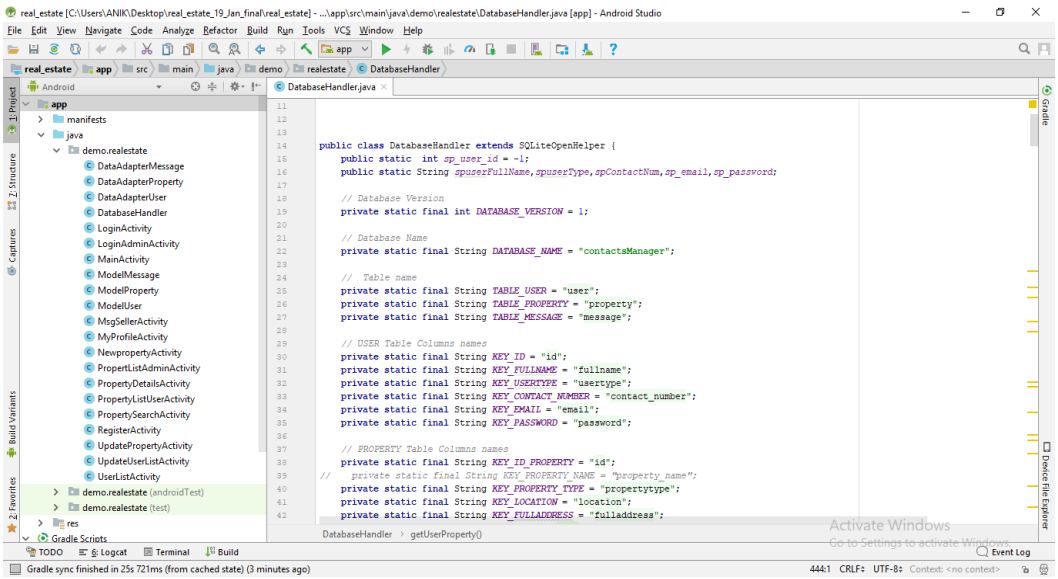


Fig. 6.1 Android Studio IDE

6.2 DEVELOPMENT TOOLS AND TECHNOLOGIES

XML

XML stands for Extensible Markup Language and is a text-based markup language derived from Standard Generalized Markup Language (SGML).

XML is a markup language much like HTML used to describe data. XML tags are not predefined in XML. We must define our own Tags. Xml as itself is well readable both by human and machine. Also, it is scalable and simple to develop.

In Android we use xml for designing our layouts because xml is lightweight language so it doesn't make our layout heavy.

In this article we will go through the basic concepts of xml in Android and different XML files used for different purpose in Android. This will help you in writing a UI code to design your desired user interface.

JAVA

Java is a high-level programming language originally developed by Sun Microsystems and released in 1995. Java runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX.

There are lots of applications and websites that will not work unless you have Java installed, and more are created every day. Java is fast, secure, and reliable. From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!

Java platform is a collection of programs that help to develop and run programs written in the Java programming language. Java platform includes an execution engine, a compiler, and a set of libraries. JAVA is platform-independent language. It is not specific to any processor or operating system.

JAVASCRIPT

JavaScript is a programming language that allows you to implement complex things on web pages, every time a web page does more than just sit there and display static information for you to look at displaying timely content updates, or interactive maps, or animated 2D/3D graphics, or scrolling video jukeboxes, etc.

Anyone can bet that JavaScript is probably involved. It is the third layer of the layer cake of standard web technologies, two of which (HTML and CSS) we have covered in much more detail in other parts of the Learning area.

Why JavaScript?

JavaScript is most commonly used as a client side scripting language. This means that JavaScript code is written into an HTML page. When a user requests an HTML page with JavaScript in it, the script is sent to the browser and it's up to the browser to do something with it.

6.3 ANDROID DATABASE SQLITE AND API VERSION

SQLite is an in-process library that implements a self-contained, server less, zero configurations, transactional SQL database engine. The code for SQLite is in the public domain and is thus free for use for any purpose, commercial or private. SQLite is a compact library.

IT is an open source SQL database that stores data to a text file on a device. Android comes in with built in SQLite database implementation. SQLite supports the entire relational database feature

Android API	SQLite Version
API 27	3.19
API 26	3.18
API 24	3.9
API 21	3.8
API 11	3.7
API 8	3.6
API 3	3.5
API 1	3.4

Fig. 6.2 SQLite and API Version

CHAPTER-7
IMPLEMENTATION

Project implementation (or project execution) is the phase where visions and plans become reality. This app can be used in Real Estate Company. Company can use it. It will be helpful for real estate broker also.

This app is fully informative. So, if any one wants, he or she can keep up the information about real estate property. A user will know where he/she can found the desire home he/she is looking for by this.

SQLite database is used for this app's data store. It stores data in structured manner. SQLite has higher performance. It can also be queried and the data retrieval is much more robust. The application only has to load as much data as it needs, rather than reading the entire application file and holding a complete parse in memory. Startup time and memory consumption are reduced.

7.1 OPENING APP

I choose the icon of the application as a real estate icon because this app is representing a real estate world. It will work offline; no active internet connection is required.

This app is totally informative.

- After installing successfully, this is the ICON of REAL ESTATE APP and by a single tap; users would be able to open the app.

7.2 HOME SCREEN

The information of the property is taken from websites that are reviewed by the brokers..

In Home screen app divided into 2 main parts:

- Property List
- Menu containing the item
  - ✓ Registration
  - ✓ Log in
  - ✓ Log in as Admin
  - ✓ Search Property

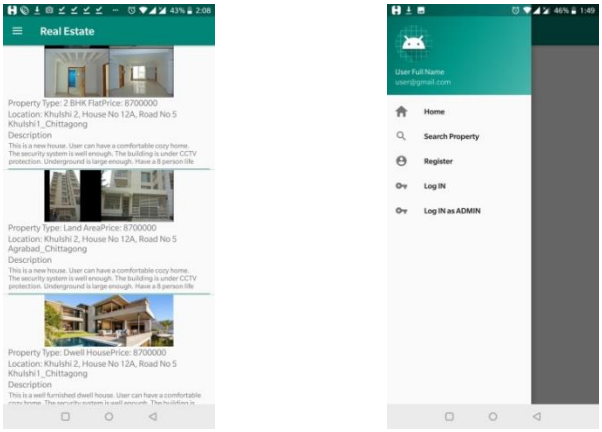


Fig. 7.1 Home Screen and Menu before Log IN

7.3 PROPERTY DETAILS

When user taps one of the properties from the property list, property details page will occur. In the details page all the information regarding that property which is given by the seller will show.

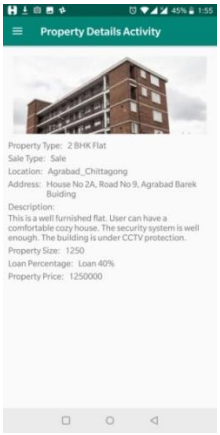


Fig. 7.2 Property Details

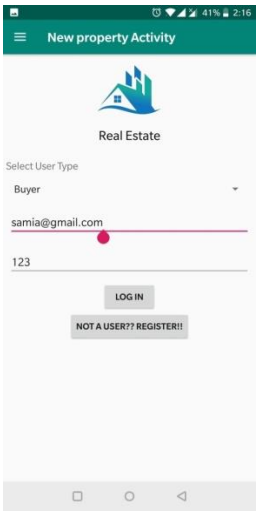
7.4 REGISTRATION

If user uses this app for the first time user need to register in the app as a buyer or seller. Seller can sale or rent a property where buyer can buy the property.

Fig. 7.3 Registration

7.5 LOG IN AS BUYER

After registration user can log in into the app. If User is a buyer he/she can see the property list. Buyer can see the property details. If Buyer want to communicate with the seller can easily communicate through messaging



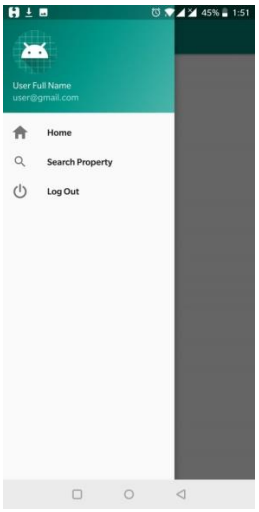


Fig. 7.4 Buyer Log IN and Menu

7.6 LOG IN AS SELLER

After registration user can log in into the app as a seller. If User is a seller he/she can see the property list. Seller can see the property details. If Seller want to communicate with other seller can easily communicate through messaging. Seller can add new property. He/she can manage his/her property. If there needs any changes seller can easily do this. Or if the property is being sold or rented seller can delete the property

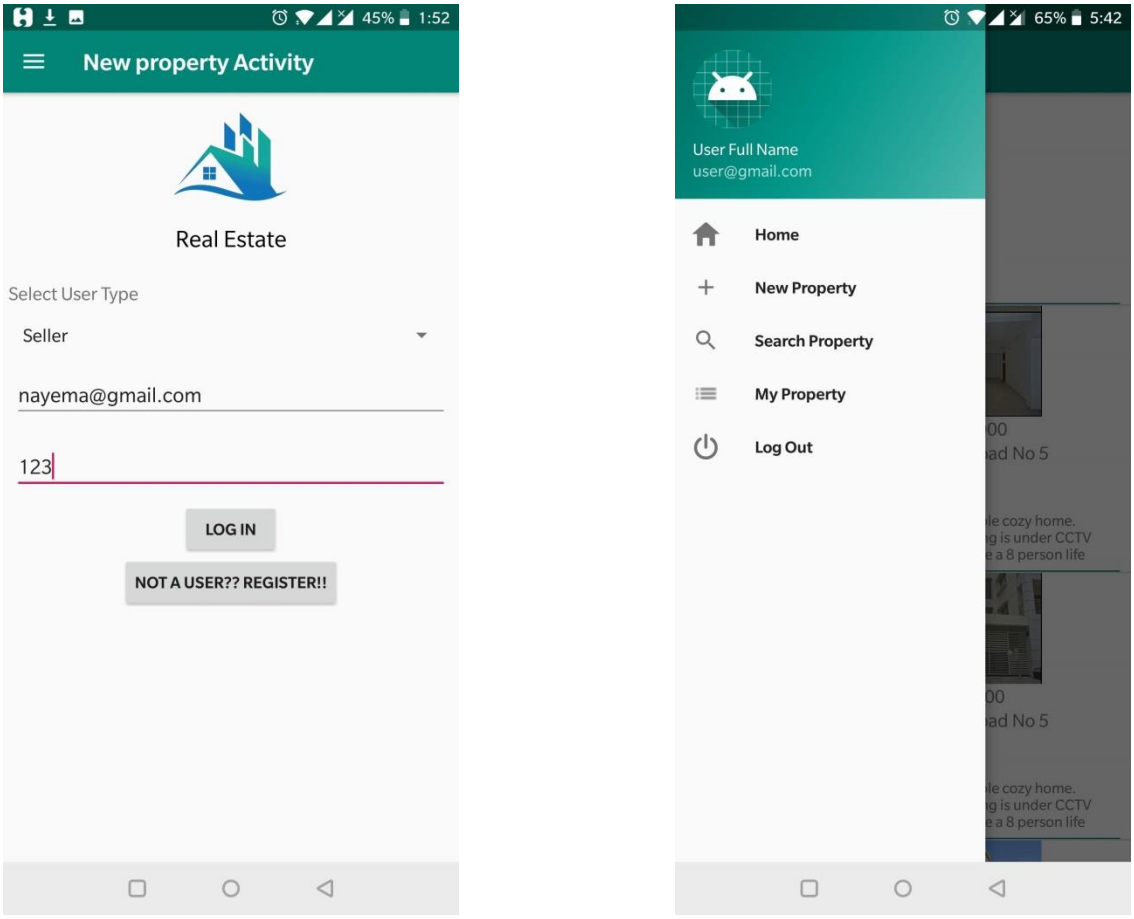


Fig. 7.5 Seller Log IN and Menu

7.7 LOG IN AS ADMIN

This app has an admin who has all the authorities. Admin can add new property. He/she can manage the property. If there needs any changes admin can easily do this. Or if the property is being sold or rented admin can delete the property. Admin has the authority to terminate any user if he found any user corrupted. Admin can delete any property if he found the property corrupted.

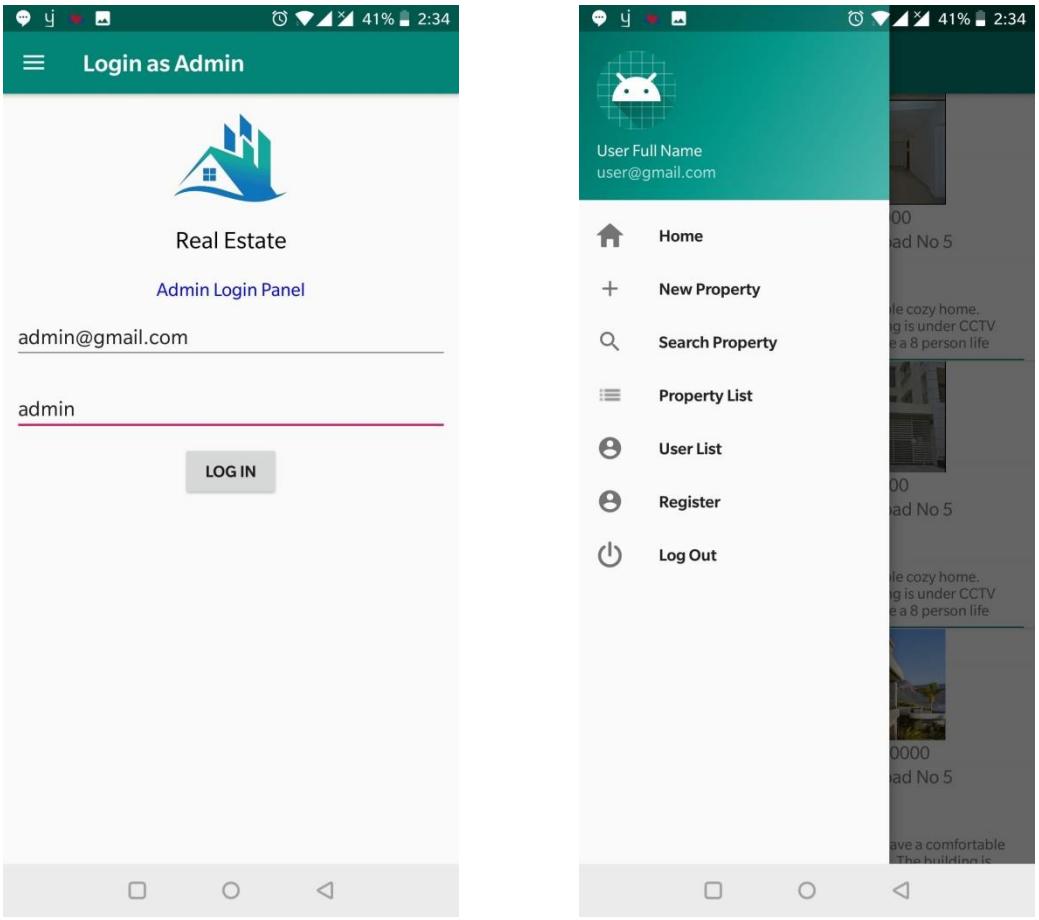
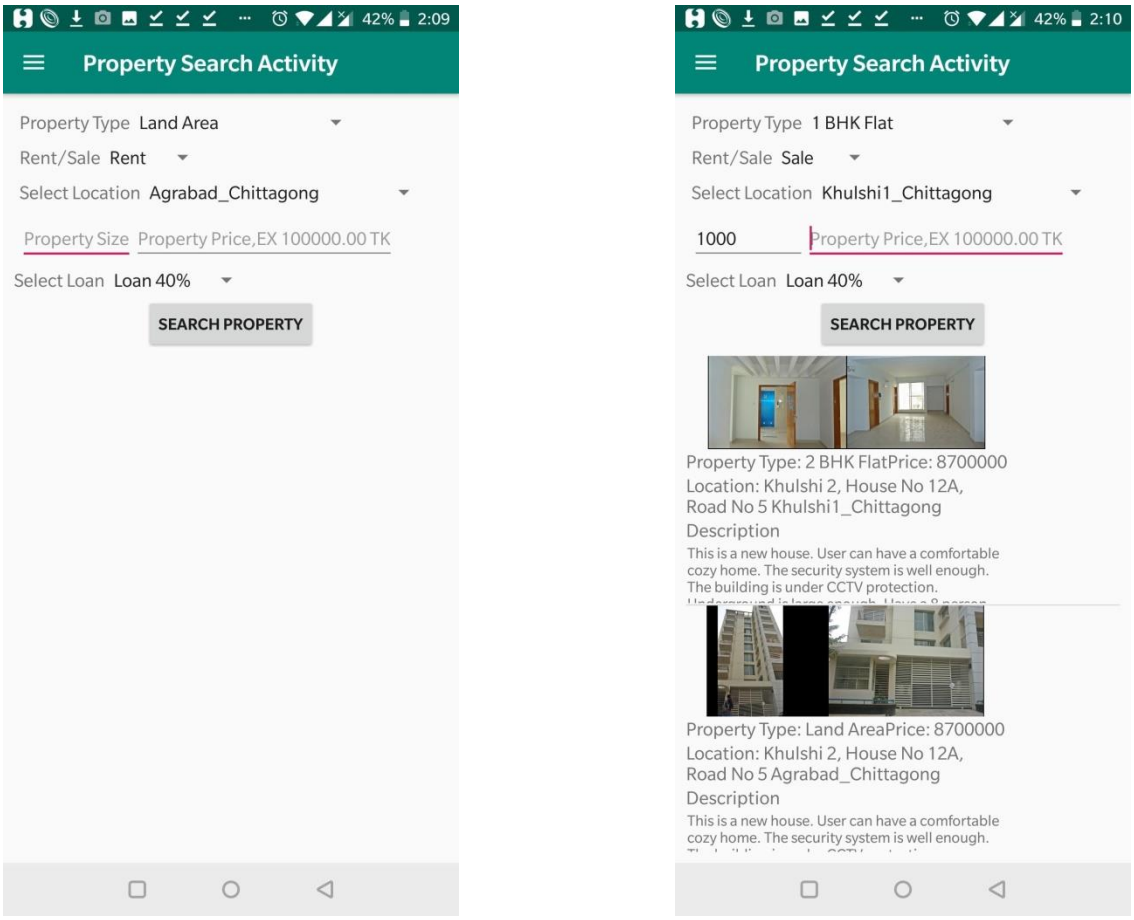


Fig. 7.6 Admin Log IN and Menu

7.8 SEARCH PROPERTY

This app has a very important feature which is searching. User can easily find a property by property type, property size, property price, address, loan percentage. After selecting the options by tapping on the search option user will find his/her desirable property. If one of the options matches it will collect all the properties and will show as a list.



.

Fig. 7.7 Search Property and Search Result

7.9 ADDING AND MANAGING PROPERTY

This app has another very important feature which is adding and managing property. Seller and admin can add a new property. If any changes needed, property can be updated. If the property already used, it can be deleted by the seller. If admin found the property fake, admin can delete the property.

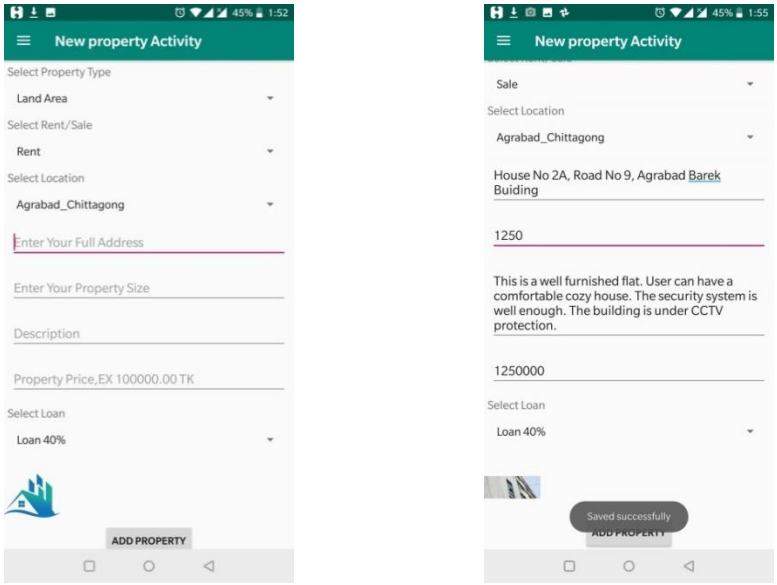


Fig. 7.8 Add new Property by Seller/ Admin

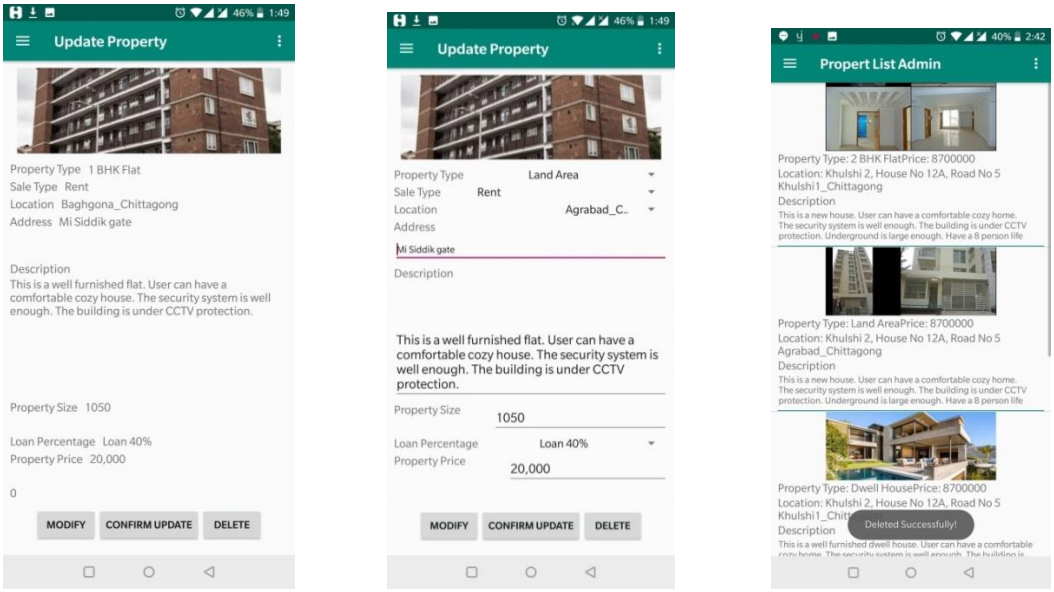


Fig. 7.9 Manage Property

7.10 MESSAGING BETWEEN BUYER AND SELLER

This app has another very important feature is messaging. Buyer and seller can communicate with each other depending on the property.

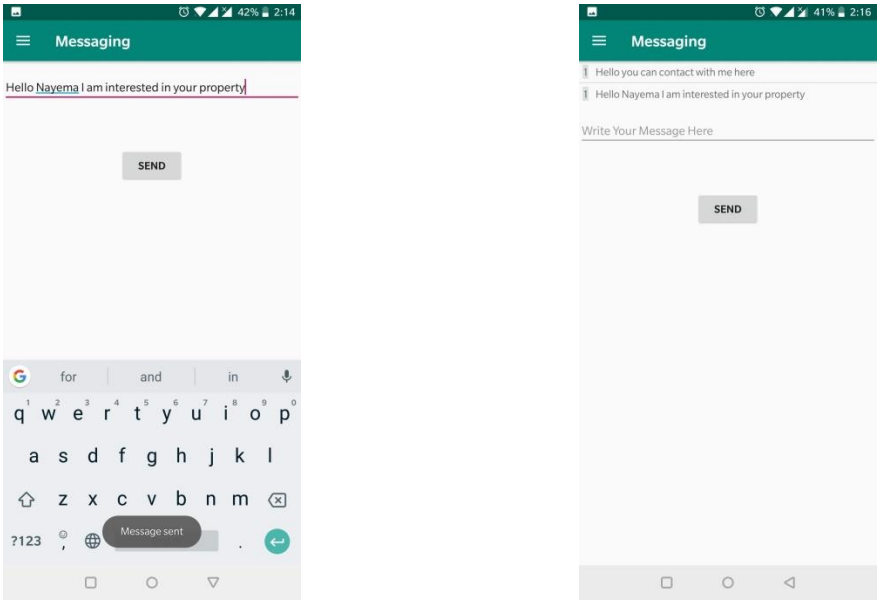


Fig. 7.10 Buyer Can Send Message to Seller and Getting Reply from Seller

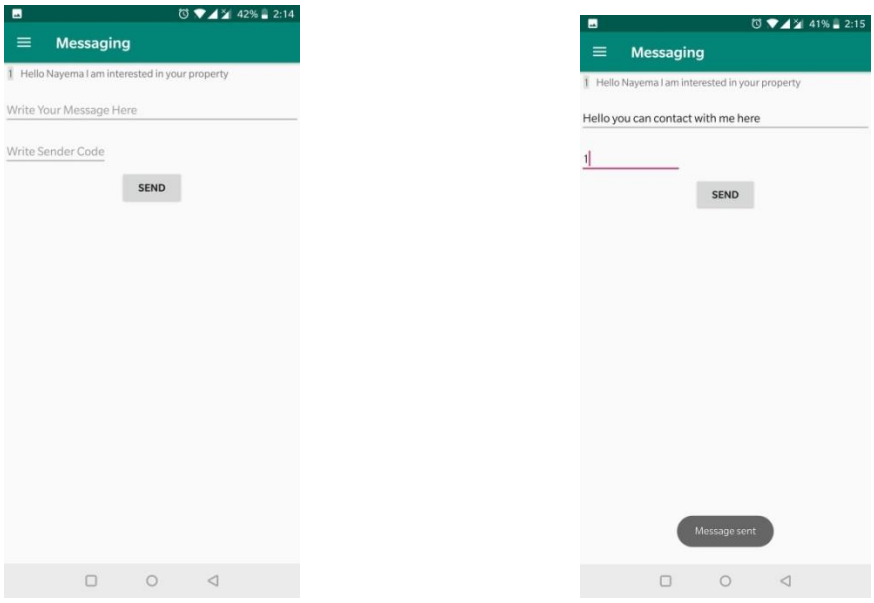


Fig. 7.11 Getting Message from Buyer and Replying to Buyer

7.11 MANAGING USER

Admin can see all the users whom are registered in the app. Admin can terminate any user if they found corrupted or change user profile if it's needed

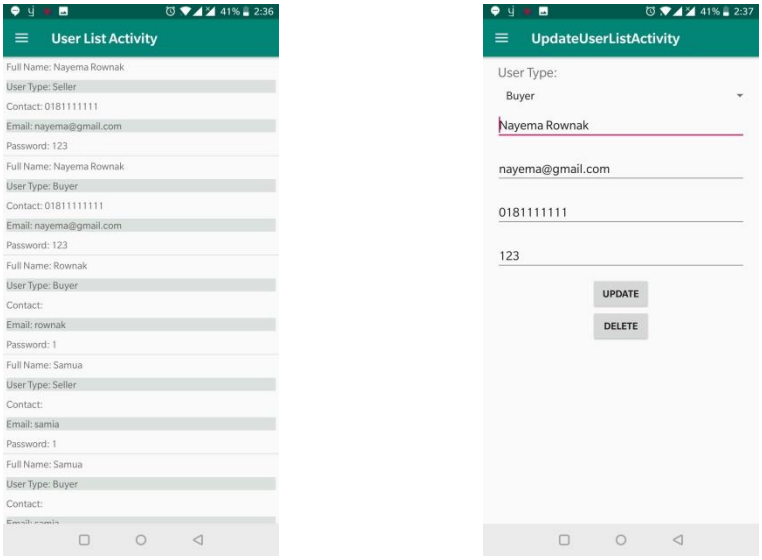


Fig. 7.12 Buyer Can Send Message to Seller and Getting Reply from Seller

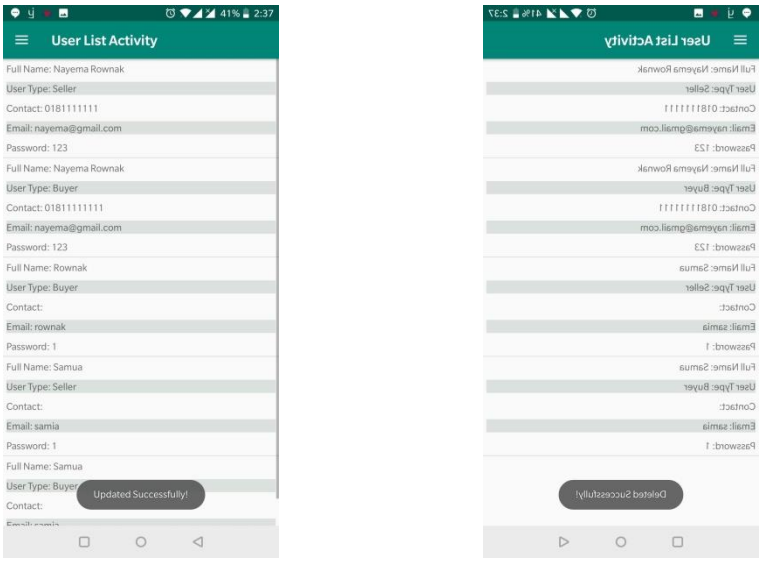


Fig. 7.13 Buyer Can Send Message to Seller and Getting Reply from Seller

CHAPTER-8
TESTING

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation.

Software testing can be stated as the Process validating and verifying that a software application:

- Meets the requirements that guided its design and development 2. Works as expected 3. Can be implemented with the same characteristics.
- Any engineering product can be tested in one of two ways:
- Knowing the specified function that a product has been designed to perform, tests can be conducted that demonstrate each function is fully operational while at the same time searching for error in each function.
- Knowing the internal working of a product, tests can be conducted to ensure that all internal operation is performed according to specification and all internal components have been adequately examined.

8.1 CLASSIFICATION OF TESTING

There are two basic type of testing. They are:

Static Testing: Static testing is related with inspecting the code. For this software, all the codes reviewed several times.

Dynamic Testing: There are two types of dynamic testing. They are:

- Black Box Testing or Functional Testing
- White Box Testing or Structural Testing

In this project, have been followed both testing

8.1.1 BLACK BOX TESTING

This is the method in which internal structure of the product is not known to the tester. The term black box is used because we do not look inside the application technical working. It tests from the end-user's perspective. Input and Output are tested and checked against the expected outputs. Programming knowledge is not required. This application was successful in black box testing because its input and corresponding output are correct.

In this testing, following terms of the app have been tested,

- In Home Screen all Properties are showing correctly
- Search function is working well
- Menu button working correctly
- Dropdown menu is showing correctly
- Adding and Managing Properties is working correctly
- Registration is working correctly
- Messaging between buyer and seller working correctly

8.1.2 WHITE BOX TESTING

This is the method in which internal structure of the product is known to developers. Developers test the application in White box testing. Internal functionality is tested here. Programming knowledge is required here.

White box testing is performed early in the stage out. White box was performed in application when project have been developing. Codes execute without error and white box testing is successful.

The Fields that's are closely handled

- ✓ Admin Login Portion
- ✓ Inserting New Property
- ✓ Register in the App as Buyer & Seller
- ✓ Inserting, Modifying, Removing Property by Admin and Seller
- ✓ Terminating User by Admin
- ✓ Searching Option
- ✓ Messaging Between Buyer & Seller

CHAPTER-9
EVALUATION

9.1 EVALUATION

In order to evaluate I have checked the usability of my system by 20 people. Here, I have presented what they responded or how they evaluated.

Testing Area	Worked	Modified Work	Not Worked
Login Panel for Admin with correct email & Password	15	5	0
Login Panel for Admin with wrong email & Password	18	2	0
Login Panel for User with correct email & Password	15	5	0
Login Panel for User with wrong email & Password	16	4	0
Inserting, Modifying & Removing Property	17	3	0
Search	14	2	4
Messaging between Buyer & Seller	14	6	0
Terminating User as a Admin	20	0	0

Responses as Percentage

Comments	Percentage
Useful	75%
Very Useful	2%
Moderate Useful	15%
Not Useful	8%

CHAPTER-10
CONCLUSION AND FUTURE WORK

10.1 CONCLUSION

Android Application on Real Estate Management System is mainly a Real Estate Management App which is designed for both Buyer and Seller. Buyer and Seller both can have a secure, admin controlled platform. Buyer can be assured about the Property is not Fake. Seller can be sure the Buyer is authorized. Admin has a closed look on the user. If Admin found something wrong about the User can terminate the User. Above All Buyer and Seller can communicate with each other through the App by Messaging. Message option are completely confidential.

10.2 FUTURE WORK

In future, there will be added more functionalities and information in Real Estate Management System. Following functionalities can be added such as:

- Better and beautiful user interface
- Search option will be more optimized
- Message option will be more user friendly
- Property Details will be more informative
- Multiple Images will be included in the App
- More Features will be included in the Property Details
- Map will be included in the App using Address
- Buyer can Text Seller using Contact Number through the App
- Buyer can Call Seller using App
- User Image will be include in the App
- Admin will send Email for the verification
- A verification Code will be add in the App

REFERENCES

- Most helpful search engine: https://www.google.com
- Android studio: https://developer.android.com
- JAVA: http://docs.oracle.com/javase
- Tutorials: https://youtube.com
- Converter: http://www.convertcsv.com
- Gradle dependencies: https://github.com/programingjd/justified
- https://stackoverflow.com
- https://www.tutorialspoint.com
- http://www.wikipedia.com

APPENDICES

IMPLEMENTATION IN XML

Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context=".MainActivity"
    tools:showIn="@layout/app_bar_main">

    <ListView
        android:id="@+id/list_property"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    </ListView>

</android.support.constraint.ConstraintLayout>
```

List_property.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:orientation="vertical"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:layout_margin="10dp">
   <ImageView
      android:id="@+id/imgIcon"
      android:layout_width="match_parent"
      android:layout_height="100dp"
      android:src="@drawable/real_state_logo"/>
   <LinearLayout
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:orientation="horizontal">
     <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/textViewPropertyType"
        android:textSize="16dp"
        android:layout_gravity="left"
        android:text="Property Type"
        />

      <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="16dp"
        android:id="@+id/textViewPrice"
        android:layout_gravity="right"
        android:text="Price"
```

```
    />


</LinearLayout>
<TextView
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:textSize="16dp"
   android:id="@+id/textViewAddress"
   android:layout_gravity="right"
   android:text="Address"
   />
<TextView
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:text="Description"
   android:textSize="16dp"
   />
<TextView
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:layout_margin="1dp"
   android:id="@+id/textViewDescription"
   android:text="Description"
   android:textSize="13dp"
   />
<View
   android:layout_width="match_parent"
   android:layout_height="1dp"
   android:layout_marginTop="5dp"
   android:background="@color/colorPrimary"/>
```

```
    </LinearLayout>

</android.support.constraint.ConstraintLayout>
```

Main_activity.xml:

```
package demo.realestate;

import android.content.Intent;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.view.View;
import android.support.design.widget.NavigationView;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.AdapterView;
import android.widget.ListView;
import android.widget.Toast;

import java.util.ArrayList;
import static demo.realestate.DatabaseHandler.spuserFullName;
import static demo.realestate.DatabaseHandler.spuserType;
```

```
import static demo.realestate.DatabaseHandler.spContactNum;
import static demo.realestate.DatabaseHandler.sp_email;
import static demo.realestate.DatabaseHandler.sp_password;
import static demo.realestate.DatabaseHandler.sp_user_id;

public class MainActivity extends AppCompatActivity
        implements NavigationView.OnNavigationItemSelectedListener {

    private DatabaseHandler db;
    private ListView lv;
    private DataAdapterProperty data;
    private ModelProperty dataModel;
    private ArrayList<ModelProperty> propertList;
    public static int login=0;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        if(login==0)
            visibility_logout();
        else if(login==1)
            visibility_login_admin();
        else if(login==2)
            visibility_login_seller();
        else if(login==3)
            visibility_login_buyer();
```

```
        //Instantiate database handler
        db = new DatabaseHandler(this);
        lv = (ListView) findViewById(R.id.list_property);
        PropertyList();

//        Toast.makeText(getApplicationContext(), spuserFullName,
Toast.LENGTH_LONG).show();
        DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
                this, drawer, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
        drawer.addDrawerListener(toggle);
        toggle.syncState();

        NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
        navigationView.setNavigationItemSelectedListener(this);
    }

    private void PropertyList() {
        propertList = new ArrayList<>(db.getAllProperty());
//        contactsList = db.getAllContacts();
        data = new DataAdapterProperty(this, propertList);

        lv.setAdapter(data);

        lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

                dataModel = propertList.get(position);
```

```
            Intent intent = new Intent(MainActivity.this,PropertyDetailsActivity.class);
            intent.putExtra("property",propertList.get(position));
            startActivity(intent);

        }
    });

}

@Override
public void onBackPressed() {
    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    if (drawer.isDrawerOpen(GravityCompat.START)) {
        drawer.closeDrawer(GravityCompat.START);
    } else {
        super.onBackPressed();
    }
}




@SuppressWarnings("StatementWithEmptyBody")
@Override
public boolean onNavigationItemSelected(MenuItem item) {
    // Handle navigation view item clicks here.
    int id = item.getItemId();

    if (id == R.id.nav_home) {
        startActivity(new Intent(MainActivity.this,MainActivity.class));
```

```
} else if (id == R.id.nav_search_property) {
    startActivity(new Intent(MainActivity.this,PropertySearchActivity.class));

}else if (id == R.id.nav_register) {
    startActivity(new Intent(MainActivity.this,RegisterActivity.class));

}else if (id == R.id.nav_user_list) {
    startActivity(new Intent(MainActivity.this,UserListActivity.class));

} else if (id == R.id.nav_new_property) {
    startActivity(new Intent(MainActivity.this,NewpropertyActivity.class));

}  else if (id == R.id.nav_property_list_admin) {
    startActivity(new Intent(MainActivity.this,PropertListAdminActivity.class));

}  else if (id == R.id.nav_property_list) {
    startActivity(new Intent(MainActivity.this,PropertyListUserActivity.class));

}  else if (id == R.id.nav_user_profile) {
    startActivity(new Intent(MainActivity.this,MyProfileActivity.class));

} else if (id == R.id.nav_login_admin) {
    startActivity(new Intent(MainActivity.this,LoginAdminActivity.class));

}  else if (id == R.id.nav_login) {
    startActivity(new Intent(MainActivity.this,LoginActivity.class));

} else if (id == R.id.nav_logout) {
    visibility_logout();
    login = 0;
    sp_user_id=0;
```

```
        spuserFullName="";
        spuserType="";
        spContactNum="";
        sp_email="";
        sp_password="";
        startActivity(new Intent(MainActivity.this,MainActivity.class));

    }

    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    drawer.closeDrawer(GravityCompat.START);
    return true;
}

public void login_check()
{
    if(login==0)
        visibility_logout();
    else if(login==1)
        visibility_login_admin();
    else if(login==2)
        visibility_login_seller();
    else if(login==3)
        visibility_login_buyer();
}

public void visibility_login_admin()
{

    NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
    navigationView.setNavigationItemSelectedListener(this);
```

```
    android.view.Menu menu =navigationView.getMenu();

    MenuItem target_home = menu.findItem(R.id.nav_home);
    MenuItem target_new_property = menu.findItem(R.id.nav_new_property);
    MenuItem target_search_property = menu.findItem(R.id.nav_search_property);
    MenuItem target_property_list = menu.findItem(R.id.nav_property_list_admin);
    MenuItem target_my_property_list = menu.findItem(R.id.nav_property_list);
    MenuItem target_user_list = menu.findItem(R.id.nav_user_list);
    MenuItem target_user_register = menu.findItem(R.id.nav_register);
    MenuItem target_user_profile = menu.findItem(R.id.nav_user_profile);
    MenuItem target_login = menu.findItem(R.id.nav_login);
    MenuItem target_login_admin = menu.findItem(R.id.nav_login_admin);
    MenuItem target_logout = menu.findItem(R.id.nav_logout);

    target_home.setVisible(true);
    target_new_property.setVisible(true);
    target_search_property.setVisible(true);
    target_property_list.setVisible(true);
    target_my_property_list.setVisible(false);
    target_user_list.setVisible(true);
    target_user_register.setVisible(true);
    target_user_profile.setVisible(false);
    target_login.setVisible(false);
    target_login_admin.setVisible(false);
    target_logout.setVisible(true);



}
public void visibility_login_seller()
```

```
{

    NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
    navigationView.setNavigationItemSelectedListener(this);

    android.view.Menu menu =navigationView.getMenu();

    MenuItem target_home = menu.findItem(R.id.nav_home);
    MenuItem target_new_property = menu.findItem(R.id.nav_new_property);
    MenuItem target_search_property = menu.findItem(R.id.nav_search_property);
    MenuItem target_property_list = menu.findItem(R.id.nav_property_list_admin);
    MenuItem target_my_property_list = menu.findItem(R.id.nav_property_list);
    MenuItem target_user_list = menu.findItem(R.id.nav_user_list);
    MenuItem target_user_register = menu.findItem(R.id.nav_register);
    MenuItem target_user_profile = menu.findItem(R.id.nav_user_profile);
    MenuItem target_login = menu.findItem(R.id.nav_login);
    MenuItem target_login_admin = menu.findItem(R.id.nav_login_admin);
    MenuItem target_logout = menu.findItem(R.id.nav_logout);

    target_home.setVisible(true);
    target_new_property.setVisible(true);
    target_search_property.setVisible(true);
    target_property_list.setVisible(false);
    target_my_property_list.setVisible(true);
    target_user_list.setVisible(false);
    target_user_register.setVisible(false);
    target_user_profile.setVisible(false);
    target_login.setVisible(false);
    target_login_admin.setVisible(false);
    target_logout.setVisible(true);
```

```
}

public void visibility_login_buyer()
{

    NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
    navigationView.setNavigationItemSelectedListener(this);

    android.view.Menu menu =navigationView.getMenu();

    MenuItem target_home = menu.findItem(R.id.nav_home);
    MenuItem target_new_property = menu.findItem(R.id.nav_new_property);
    MenuItem target_search_property = menu.findItem(R.id.nav_search_property);
    MenuItem target_property_list = menu.findItem(R.id.nav_property_list_admin);
    MenuItem target_my_property_list = menu.findItem(R.id.nav_property_list);
    MenuItem target_user_list = menu.findItem(R.id.nav_user_list);
    MenuItem target_user_register = menu.findItem(R.id.nav_register);
    MenuItem target_user_profile = menu.findItem(R.id.nav_user_profile);
    MenuItem target_login = menu.findItem(R.id.nav_login);
    MenuItem target_login_admin = menu.findItem(R.id.nav_login_admin);
    MenuItem target_logout = menu.findItem(R.id.nav_logout);

    target_home.setVisible(true);
    target_new_property.setVisible(false);
    target_search_property.setVisible(true);
    target_property_list.setVisible(false);
    target_my_property_list.setVisible(false);
    target_user_list.setVisible(false);
    target_user_register.setVisible(false);
    target_user_profile.setVisible(false);
```

```
    target_login.setVisible(false);
    target_login_admin.setVisible(false);
    target_logout.setVisible(true);


}

public void visibility_logout()
{

    NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
    navigationView.setNavigationItemSelectedListener(this);

    android.view.Menu menu =navigationView.getMenu();

    MenuItem target_home = menu.findItem(R.id.nav_home);
    MenuItem target_new_property = menu.findItem(R.id.nav_new_property);
    MenuItem target_search_property = menu.findItem(R.id.nav_search_property);
    MenuItem target_property_list = menu.findItem(R.id.nav_property_list_admin);
    MenuItem target_my_property_list = menu.findItem(R.id.nav_property_list);
    MenuItem target_user_list = menu.findItem(R.id.nav_user_list);
    MenuItem target_user_register = menu.findItem(R.id.nav_register);
    MenuItem target_user_profile = menu.findItem(R.id.nav_user_profile);
    MenuItem target_login = menu.findItem(R.id.nav_login);
    MenuItem target_login_admin = menu.findItem(R.id.nav_login_admin);
    MenuItem target_logout = menu.findItem(R.id.nav_logout);

    target_home.setVisible(true);
    target_new_property.setVisible(false);
    target_search_property.setVisible(true);
    target_property_list.setVisible(false);
```

```
        target_my_property_list.setVisible(false);
        target_user_list.setVisible(false);
        target_user_register.setVisible(true);
        target_user_profile.setVisible(false);
        target_login.setVisible(true);
        target_login_admin.setVisible(true);
        target_logout.setVisible(false);

    }
}
```

AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="demo.realestate">

    <uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name"
            android:theme="@style/AppTheme.NoActionBar">
```

```
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity
    android:name=".LoginActivity"
    android:label="@string/title_activity_newproperty"
    android:theme="@style/AppTheme.NoActionBar" />
<activity
    android:name=".NewpropertyActivity"
    android:label="@string/title_activity_newproperty"
    android:theme="@style/AppTheme.NoActionBar" />
<activity
    android:name=".RegisterActivity"
    android:label="@string/title_activity_register"
    android:theme="@style/AppTheme.NoActionBar" />
<activity
    android:name=".PropertySearchActivity"
    android:label="@string/title_activity_property_search"
    android:theme="@style/AppTheme.NoActionBar" />
<activity
    android:name=".PropertyDetailsActivity"
    android:label="@string/title_activity_property_details"
    android:theme="@style/AppTheme.NoActionBar" />
<activity
    android:name=".UserListActivity"
    android:label="@string/title_activity_user_list"
    android:theme="@style/AppTheme.NoActionBar" />
<activity
```

```
        android:name="com.theartofdev.edmodo.cropper.CropImageActivity"
        android:theme="@style/Base.Theme.AppCompat" />
    <activity
        android:name=".UpdateUserListActivity"
        android:label="@string/title_activity_update_user_list"
        android:theme="@style/AppTheme.NoActionBar" />
    <activity
        android:name=".LoginAdminActivity"
        android:label="@string/title_activity_login_admin"
        android:theme="@style/AppTheme.NoActionBar" />
    <activity
        android:name=".MyProfileActivity"
        android:label="@string/title_activity_my_profile"
        android:theme="@style/AppTheme.NoActionBar" />
    <activity
        android:name=".PropertListAdminActivity"
        android:label="@string/title_activity_propert_list_admin"
        android:theme="@style/AppTheme.NoActionBar" />
    <activity
        android:name=".UpdatePropertyActivity"
        android:label="@string/title_activity_update_property"
        android:theme="@style/AppTheme.NoActionBar" />
    <activity
        android:name=".MsgSellerActivity"
        android:label="@string/title_activity_msg_seller"
        android:theme="@style/AppTheme.NoActionBar" />
    <activity
        android:name=".PropertyListUserActivity"
        android:label="@string/title_activity_property_list_user"
        android:theme="@style/AppTheme.NoActionBar"></activity>
  </application>

</manifest>
```