PROJECT PROPOSAL

PROJECT NAME: Real Estates App

DECLARATION

App Name

Real Estate App

Platform: Android
Target Audience: Real estate professionals, home buyers, sellers, and renters

Purpose

The Real Estate Android app is designed to provide users with a comprehensive platform for browsing, listing, and managing real estate properties. It offers features such as property search, detailed listings, user reviews, and communication tools to facilitate real estate transactions.

Scope

This documentation covers the following aspects of the Real Estate app:

- **User Interface (UI)**: Descriptions of the app's screens and user interactions.
- **Features and Functionality**: Detailed information about the app's capabilities and how they work.
- **Technical Specifications**: Technical details including system requirements, APIs, and data handling.
- **Installation and Setup**: Instructions for installing and setting up the app.

ACKNOWLEDGEMENT

We would like to extend our sincere gratitude to the following individuals and for their invaluable contributions to the development and success of our Real Estate Android app:

EX:

We wish to take this opportunity to express out sincere thanks to <Name> for interest technical support and suggestions during the seminar leading to our success.

TABLE OF CONTENTS

Contents

LIST OF FIGURES

ABSTRACT

The Real Estate app is a comprehensive, user-centric platform designed to streamline the process of selling, and renting properties. This app will provide facility to the user to search Residential and Commercial property and view property. This system will provide facility to view the property by admin and user. Admin will able to upload the property information to the site and able to manage it. Leveraging cutting-edge technology and intuitive design, the app offers a robust suite of features, including advanced property search filters, interactive maps, virtual tours. Users can seamlessly browse listings, view detailed property information, and connect directly with real estate professionals. Enhanced with market insights, the app empowers users with the tools and information needed to make informed decisions and achieve their real estate goals efficiently. Whether for finding a dream home, selling a property, or discovering rental opportunities, the Real

Estate app simplifies and enriches the real estate experience for users at every stage of their journey.

CHAPTER: 1
Introduction

This chapter covers the major insights of the project. It has the introduction, then the background of the study, problem statement, purpose of the study, objectives of the study, research questions, significance of the study, scope of the study, operational definition of terms and concepts.

1.1 Introduction

This application is for gathering all crucial information about existing land for sale and houses within the country. The reason it is important is because, land is a very important factor of life and people take it seriously. Due to this reason, there are endless wrangles caused by unfaithful people who sale other people's property and usually these are middlemen!

Therefore, this system takes over that role and whoever has land to sell or a house registers with this system, forwards in all the necessary information and credentials. And now all people interested in these two in any part of the country. All the details are there. Thus creates bridging the gap between the seller and the buyer.

1.2 Motivation

When we are going to a new place, we have to stay for a period of time may be weeks or months and as we all want to stay in good room with most of facilities provided by owner. Now hunt begins for the good kind of house we spent most of time review go there and know all details or facilities of the house. And with our hard luck we had to review few houses compare and decide on one house to stay. Such a long process right. To simplify this story of go to a place to see, review and compare will make all these into one place. From one of experience like above had an idea to have all in one place to reduce the process of searching and comparing and efforts.

1.3 Objective

The main goal of our project is making an app which through the users can search for houses, review and compare. To develop a reliable system, we have some specific goals such as-

- Develop a system such that users can see and review the details of house, compare with other houses offerings and can navigate easily to the desired house through navigation.
- Develop a database so that all houses information and data can be saved properly for listing.
- User can login and search for house, go though the details of house and can watch the location of house through maps.
- Admin has the authorization to add house, update house details, remove house if it is filled.
- User will be prior as we are providing much info regarding the house.
- User friendly and attractive UI
- Handle different set of data through out application with SQLite
- Explore Gmaps API for location access and view on Map.
- To test the functionality of the developed application

1.4 SCOPE

The system covered the admin and users. Admin can add house, update house details and can remove house from list. Where user can see the list of houses and he can see detailed view of a house and can utilize the Maps to see location of house.

CHAPTER 2
Background

2.1 INTRODUCTION

Real Estate App is a project that will help the users who are looking for house for rental or to buy the house. To make there way easy this app will provide enriched UI with list of houses to go through details and navigate to location for the same. An admin has the privilege to add house, update house details and remove house and all house details are stored in database.

All these modules will be able to help the users to view and compare house details and facilities in convenient manner.

2.2 EXISTING SYSTEMS

There might be few applications or websites related to real estate but mostly they are providing paid services, and they are providing rooms of hotel booking for daily charges. But we are covering most of the houses near by added by Admin. Houses can be added, update and remove.

2.3 Background

- UI design in android platform.
- Android application development.
- Entity modelling relationship.
- Database connection.
- Integration of Gmaps in android app.

2.4 THE NEED FOR REAL ESTATE APP

They carried out a study on part of Hyderabad city area in developing Real Estate App. In order to pose questions in finding a home acceptable to all family at the right place at the right cost; a real estate app with an appropriate decision support system is necessary. Factors influencing decision-making were allocated weights and scores reflecting their importance. A range of criteria that will influence the decision must be defined. Real Estate app will help in taking decision for best house that fulfil the needs and should be cost effective.

2.5. Studies

- Desing user friendly environment
- Connecting the database with the app
- Design ER Model
- Adding pictures and vector icons to app
- Create and Manage assets provided
- Navigation in android

2.5. Challenges

- Storing image in SQLite database as image size is too large to store.
- Update house details as location object isn't initialized
- FusedLocationClient isn't get last location(current location) of device.
- Enriched UI design
- Intent putExtra and getExtra values
- Handling navigation due to different flow are on picture for same activity

2.5. TECHNOLOGIES

One of the fastest growing technology applications in the real estate industry involves improving the ability of building owners, property management. User can manage both a large number of properties and a large amount of space. The overall thrust of most of the services is to get real estate information on leased or owned property into an electronic format in order to be able to administer properties, process, and pay or collect money on house.

One of the first challenges addressed by many of the latest real estate management is the need for companies to be able to convert thousands of paper documents - from leases to contracts - into a secure digital form. Once in electronic or digital form, Brown says, the software program needs to be able to integrate with the new houses and show required info about the house to the user. In addition, the information needs to be linked to the company's other applications for fixed assets, maintenance.

Technologies Used: Android studio, Gmaps, SQLite, XML, JAVA

2.8 CONCLUSION

As a conclusion, this chapter had pointed out the strength, weakness for existing systems that have been reviewed. Next, the strength of the proposed solution will be combining the strength of reviewed existing system. Proposed solution is provided to solve the weaknesses of the existing system, thus it can be apply in small-medium real estate’s enterprise and easy for user to use.

CHAPTER 3

System Design

3.0 INTRODUCTION

This chapter contains requirements gathering techniques, system design, user interface design, and system implementation.

3.1 SYSTEM STUDY

This involves different ways through which the researcher can get the information regarding the system to be designed and the system that is currently being used. Data collection methods used include; observation, documentation review and these are explained as show below.

3.2 SYSTEM DESIGN

System design is the third basic stage in systems development life cycle following planning and analysis. It involves development methodology, user interface design and database design, the data flow diagram, system flowchart and Entity relationship diagram shall be used in system design.

Entity Relationship Diagram (ERD): This clearly communicates the key entities in a certain database and their relationship with each other. These will help the researcher to show the relationship of the entities used in the database.

System flowchart: This was used to help analyze a problem in a more effective way during system design.

Data Flow Diagram (DFD): This is a diagrammatic representation of the information flows within a system, it is used to show how information enters and leaves the system, what changes the information and where information is stored.

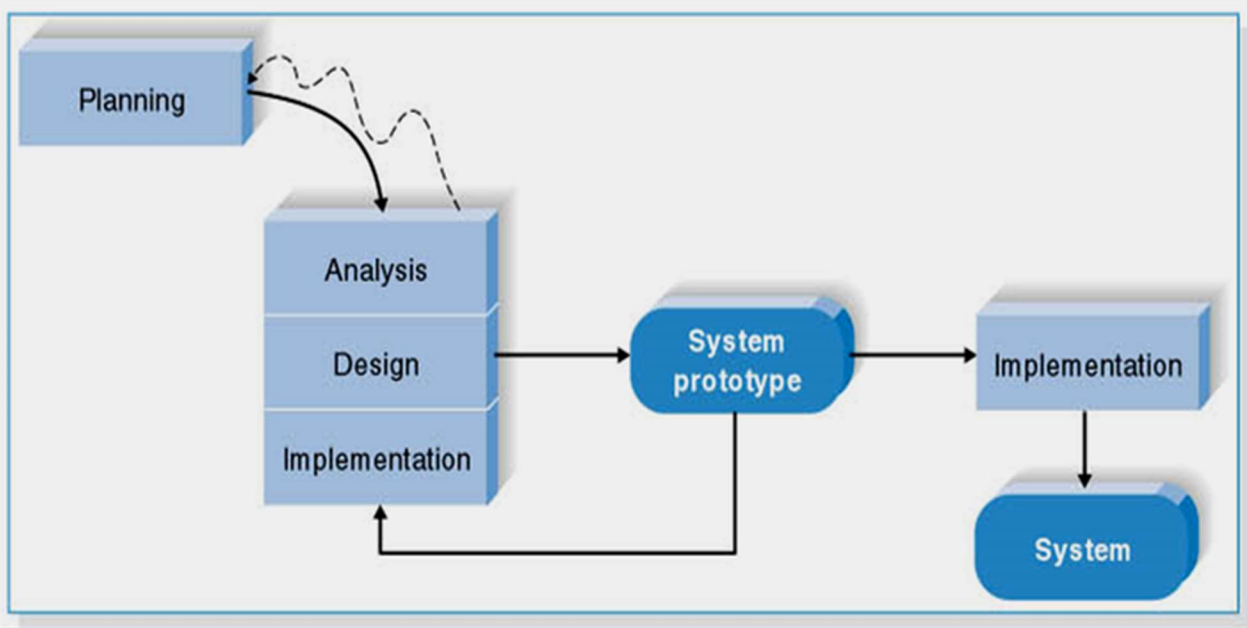
3.2.1 RAPID APPLICATION DEVELOPMENT

Rapid application development (R.A.D) is a software development methodology that uses minimal planning in favor of rapid prototyping. The planning of management system developed using RAD is interleaved with writing the system itself. The lack of extensive pre-planning generally allows systems to be written much faster, and makes it easier to change requirements.

Rapid application development is a system development methodology that involves methods like iterative development and system software prototyping. According to Whitten (2004), it is a merger of various structured techniques, especially data driven Information Engineering, with prototyping techniques to accelerate software systems development.

In rapid application development, structured techniques and prototyping are especially used to define users' requirements and to design the final system. The development process starts with the development of preliminary data models and business process models using structured techniques. In the next stage, requirements are verified using prototyping, eventually to refine the data and process models. These stages are repeated iteratively; further development results in a combined business requirements and technical design statement to be used for constructing new systems.

The system is designed and given to the users as it continues to be advanced as user requirements change. An illustration of prototyping method is shown below

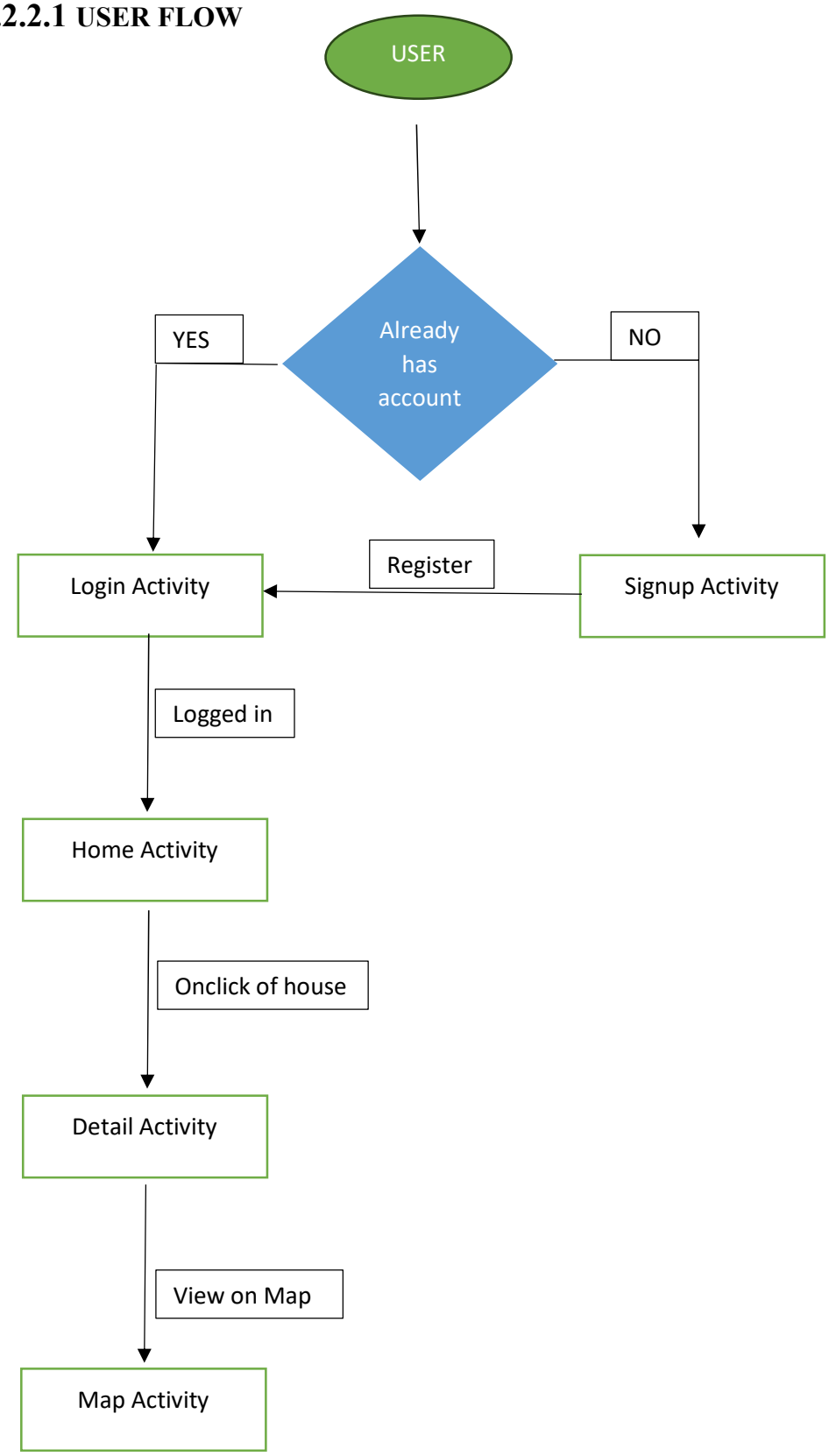


3.2.2 DATABASE DESIGN

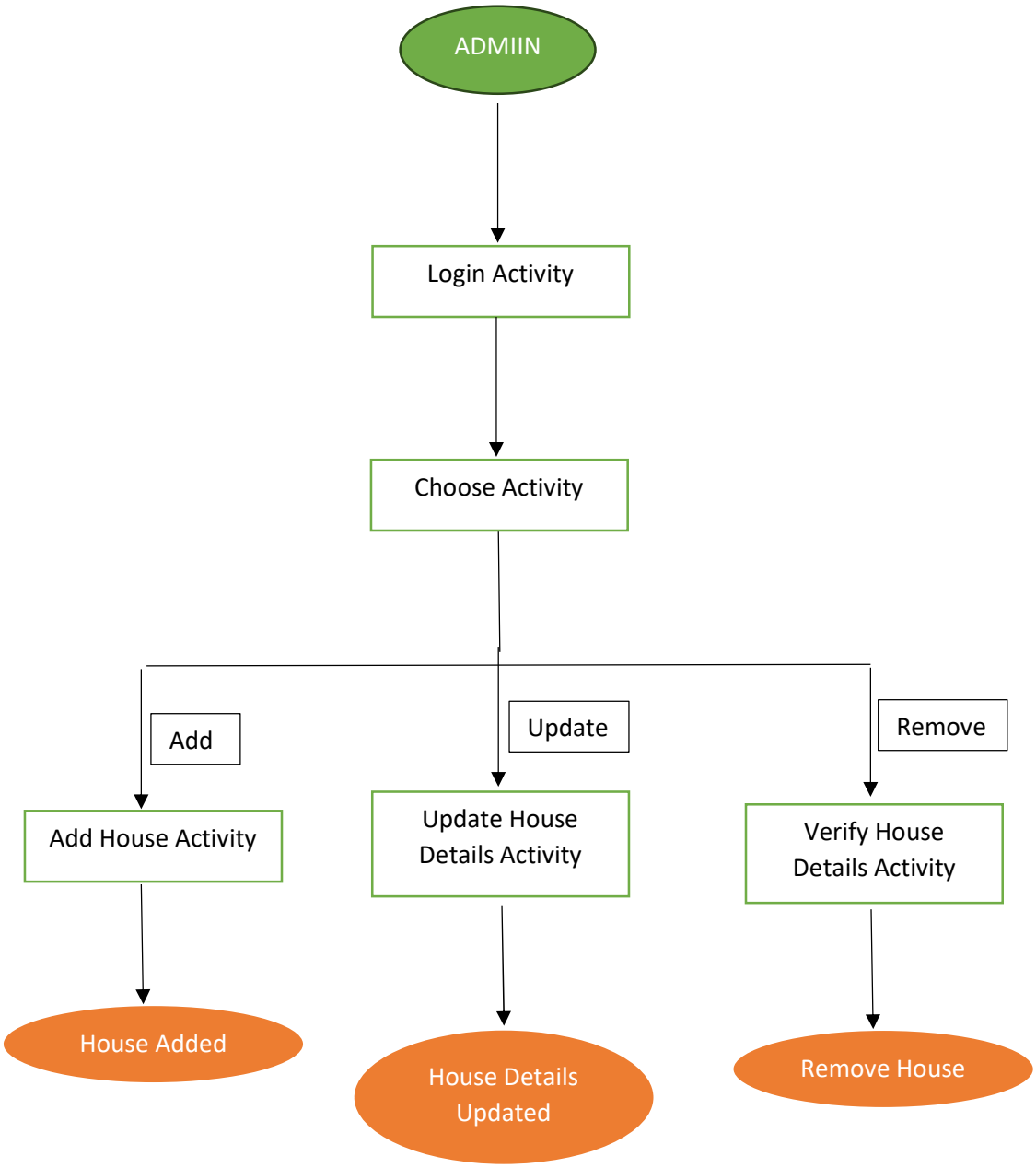
The database was designed using Structured Query Language (SQL), it was run in android studio. SQL is recommended during database design since it supports relational database systems whose advantages include the ability to provide faster access to data than flat files, and random access to data and has a built-in privilege system and most web based applications use SQL databases.

3.2.2 Diagrams

3.2.2.1 USER FLOW

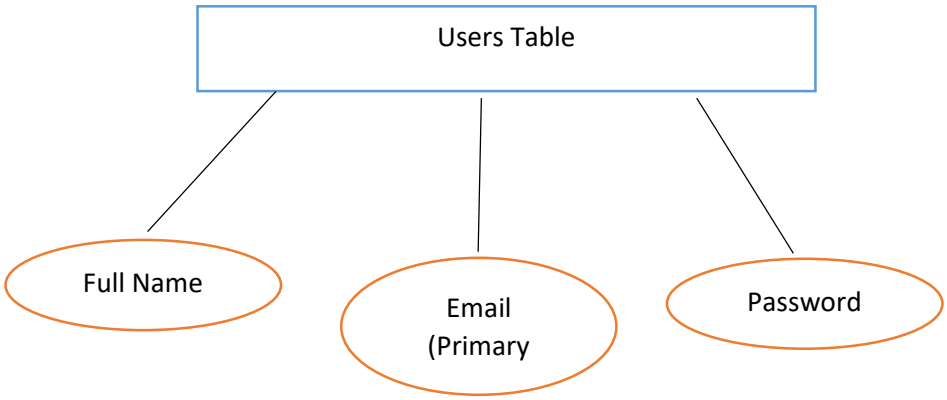


3.2.2.2 ADMIN FLOW

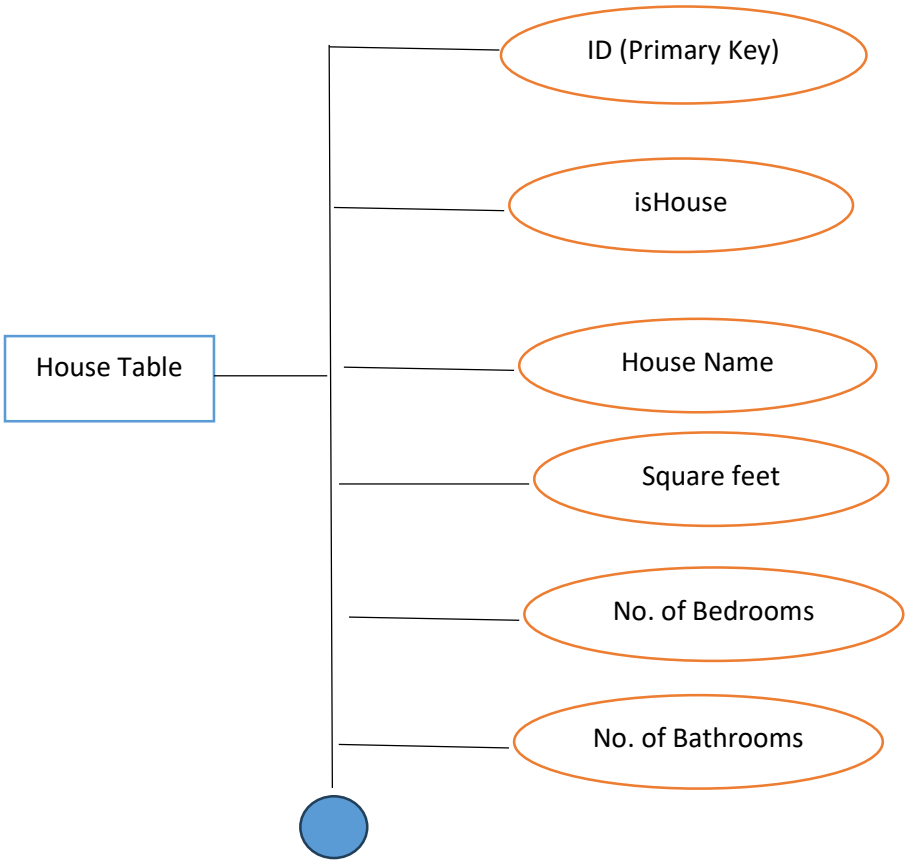


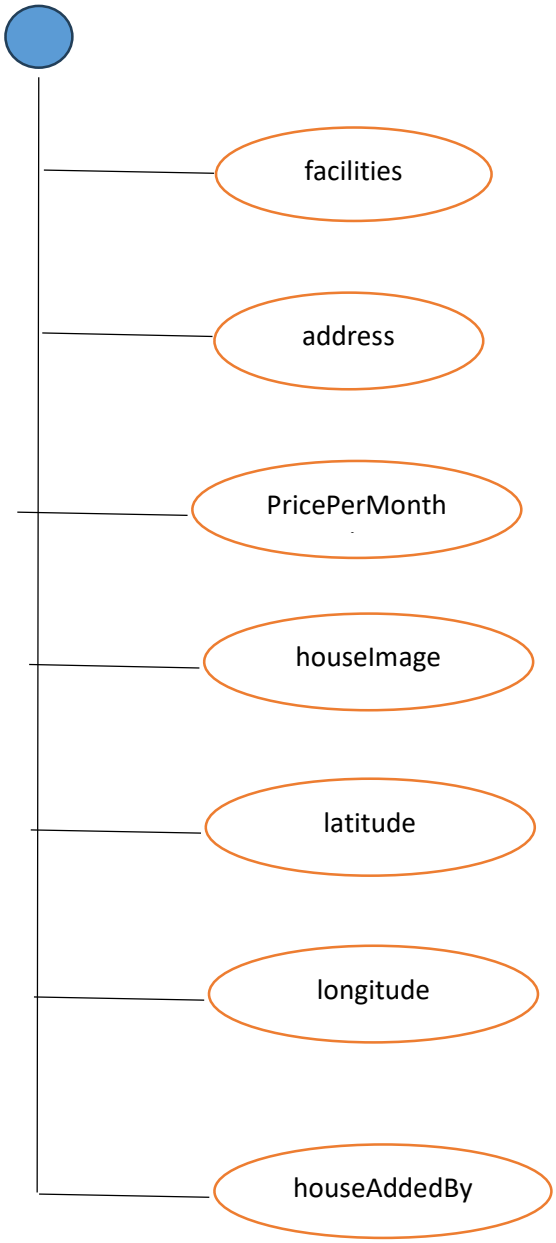
3.2.2.3 ER Diagrams

3.2.2.3.1 Users Table



3.2.2.3.2 House Table




facilities
address
PricePerMonth
houseImage
latitude
longitude
houseAddedBy

3.3 BUILDING A MOBILE ANDROID APPLICATION FOR REAL ESTATES APP

After doing extensive research on Real Estates App and available mobile applications for it, it was determined that applications for Real Estates App are very little in amount, and most of them (on mobile platforms) are not actual tools for online searching for estates but rather just things to help Online searching do better and mostly to explain to people what Real Estates are. A simple and easy to use idea for an application to help Real Estates App has come forth from the research which is one tackles the online search and buying of plots and estates. The application will target this aspect due to various reasons. The main reason would be because most applications available on mobile focus on search elements from the user, such as searching and locating plots, estates land pieces of land.

3.3.1 ANDROID APPLICATION BUILDING BLOCKS

The Application building block of Android is build out of four major components which are Activities, Services, Broadcast receiver and Content provider. Activities are basically user interfaces which allow for users' interactions. One activity usually just represents one single screen of an interface. One application usually consists of multiple activities working together. For example a calendar application might have one activity to display all written events while another activity displays the form to actually input information when adding the events. A different application can also open up an activity from another application. For example opening up email activity for sending email using the camera after taking a picture to send the picture to someone. Services are basically tasks which run in the background of the system. They handles long running applications and remote processes. They do not have a user interface and are usually invoked by an activity.

An example for the usage of a service would be to say when an activity opens up a music player and then starts playing music, but then the activity decides to open up another activity or application, the service is what keeps the music playing in the background even though a different

application has been opened. It can be asked to stop by binding an activity to it to send Intents (messages) for it to stop playing music. The Broadcast receiver responds to any changes to the system being broadcast to it with the use of Intents. It can wake up any processes it is assigned to. Basically it registers codes which would not be running until an external element awakes it. For example, a Broadcast receiver would open up a list of network connections once it senses an available network to connect to. It also does not have any UI. And finally the Content Providers allows data sharing between applications by accessing any file system or database (e.g. SQLite) on the Android system. It is also used to write data private to your application and not shared. The diagram below represents how the different component interact together. The application being built also works on the same principles.

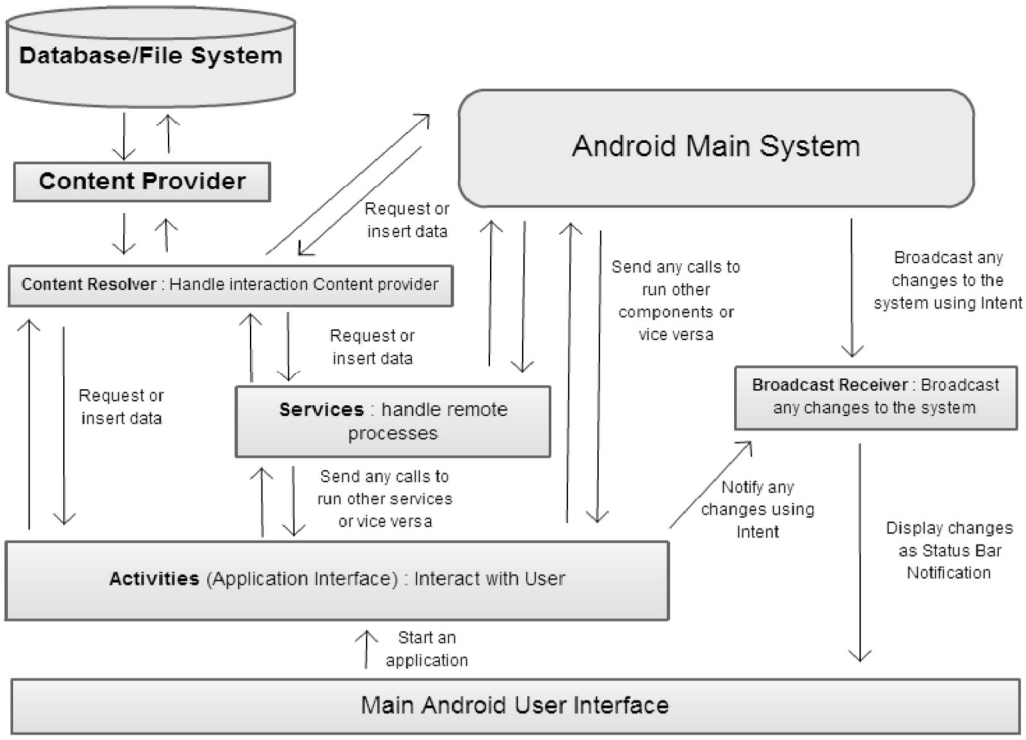


Figure 1: Interaction between main components for Android

3.3.2 APPLICATION USER INTERFACE/INPUT

According to the Android Developer website on User interface and input here are some guidelines to follow to build a good application. A good UI for an application should take into account that the application will be used in a variety of different screen sizes. So it is essential that the application is designed with this in mind. By using wrap_content and match_parent attributes for the view component of the application(usually on the xml files), the application can resize itself to either the minimum size to fit the content in that view or follow the size of the parent view which should make the view size much more flexible. This will be used to make sure that the size of the overlay being built is flexible and can be changed according to the user's needs.

Smart use of RelativeLayout could also be used to align components apart such as side by side, instead of just LinearLayout which provides a straight line type of interface. Using configuration qualifiers would allow the runtime to automatically select the best layout design for different screen sizes. Such as using two pane pattern for large screens, so that content is displayed in two screens for bigger screen but only show one pane on smaller screens. Some User input guidelines includes using onTouchEvent() function on the view which received touch events. It is called on every gesture made on the view. MotionEvent tracks the user's finger from the moment it touches the screen until it leaves the screen, and sends the data to onTouchEvent().Velocity of input (e.g. gestures) can also be tracked using VelocityTracker. The OnDragListener could also be used to detect the dragging of an object on a view. This could prove useful while manipulating the overlay around the screen and position it correctly.

3.3.3 APPLICATION TESTING

Each main activity in the application will be tested to make sure that it works accordingly. The guidelines provided by Android Developer website states that test cases should be build based on Android's custom testing framework based on JUNIT due to the fact that most android applications internal codings are build using JAVA whereas its UI can be built using XML. UI testing can be done using Instrumentation on the UI thread. Usually Test methods are added to ensure that the

correct behavior is achieve with the UI. Unit Testing can also be done to check for any bugs in the application. It is especially useful in detecting slow to draw or stutters in the display of motion/animation. Finally, the ui automator can be used to test your UI using automatically created functional UI test cases.

CHAPTER **4**

Software Requirements

4.1 SYSTEM STUDY

- This app is developed using Java and XML for front end.
- Database used: SQLite
- Operating System: Android Operating System
- Google maps

4.2 Hardware Configuration

- Min-API: 24

App Tested on:

- Android Version: Android 13
- Ram: 6GB
- ROM: 128GB

CHAPTER **5**

Implementation Screenshots

5.1 Home Activity

5.2 User Activities

5.3 Admin Activities

CHAPTER **6**

Conclusion & Future Scope

6.1 Conclusion

In this application we'll try to solve a problem where a user can search through, review and compare houses, can see and navigate to location of house and choose suitable one with the requirements. An Admin had privilege to add house, update house details and remove house from listing. This is primary functioning of application.

6.2 Future Scope of Development

There are some features we are thinking to add up in the very next version of the primary version. Those features are –

- Filter the house list according to price range selected by user.
- Search housed nearby with access of current location and house location.
- Book an appointment to visit the house and meet the owner for booking via app.

7.0 REFERENCES

Android SDK Documentation: https://developer.android.com/develop

SQLite Documentation: https://developer.android.com/training/data-storage/sqlite

Datatypes in SQLite Documentation: https://www.sqlite.org/datatype3.html

Google Maps SDK Documentation:
https://developers.google.com/maps/documentation/android-sdk#get-started

Fused Location Provider:
https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient

BIBLIOGRAPHY

[1] Wikipedia. Android Operating System
http://en.wikipedia.org/wiki/Mobile_operating_system

[2] Android. Inside the Android Framework, 2011.
stid.googlecode.com/files/Inside_the_Android_Application_Framework.pdf

[3] Wikipedia. Andriod Operating System, 2011.
http://en.wikipedia.org/wiki/Android_%28operating_system%29

[4] Google. Google Code Homepage.
http://code.google.com

[5] Android Wiki. Diagram of the Android Architecture.
http://en.androidwiki.com/wiki/Diagram_of_the_Android_architecture

[6] Android.com. Dev Guide.
http://developer.android.com/guide/topics/fundamentals/activities.html

[7] Android.com. APO.
http://developer.android.com/reference/android/app/Activity.html

[8] Android.com. The Developer's Guide.
http://developer.android.com/guide/index.html

[9] Android.com. Intents and Intent Fillers.
http://developer.android.com/guide/topics/intents/intents-filters.html

[10] Android Wiki. Andriod SDK.
http://en.androidwiki.com/wiki/Android_SDK

[11] Android Wiki. Using the Android SDK Tools.
http://en.androidwiki.com/wiki/Using_the_Android_SDK_tools

[12] Android.com. Installing the SDK.
http://developer.android.com/sdk/installing.html