ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. 3rd Year. We owe special debt of gratitude to **Associate Professor Dr. Rajender Kumar**, Department of Computer Science & Engineering, PIET, Samalkha for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of **Professor Dr. S C Gupta**, **Head, Department of Computer Science & Engineering**, PIET, Samalkha for his full support and assistance during the development of the project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Signature:

Name : BHARAT

Roll No.: 2820182

Date :

Signature:

Name : ANOOP TRIPATHI

Roll No.: 2820116

Date :

Signature:

Name : TUSHAR GARG

Roll No.: 2820211

Date :

ABASTRACT

The era of mobile technology opens the windows to the android app. The websites are vanishing and the mobile phones are emerging. It’s the time to change from conventional websites to apps, which has become the part of our daily routine. We are introducing ‘**CITY SENSE**’ the android application. Mobile application to provide all generic public information of selected city of India. It gives us more comfort and a better user interface The app aims to provide residents and visitors with a comprehensive platform for accessing up-to date information about the city, including points of interest, events, and public transportation schedules. Through an examination of the app's features, benefits, and potential impact, this study sheds light on the transformative potential of such technology in improving urban connectivity and fostering a sense of community. The findings suggest that the city information app holds great promise for enhancing the quality of life in cities by empowering individuals with valuable information and facilitating their engagement with the urban environment.

DECLARATION

We hereby declare that the work presented in this project report entitled "**CITY SENSE**", in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology in Computer Science & Engineering.**, submitted to **Kurukshetra University, Kurukshetra**, India is an authentic record of our own work carried out during the period from Jan, 2023 to June 2023 under the guidance of **Dr. Rajender Kumar** (**Associate Professor)** of Panipat Institute of Engineering and Technology.

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Bharat[Leader]	**Anoop Tripathi**	**Tushar Garg**
2820182	**2820116**	**2820211**

CERTIFICATE

This is to certify that the **Project-I report (PROJ–CS-302A)** entitled "**City Sense**" done by **Bharat, Enrollment No 2820182 ; Anoop Tripathi Enrollment No 2820116; and Tushar Garg Enrollment No 2820211**; is an authentic work carried out by them at PIET, Samalkha, Panipat under my guidance. The matter embodied in this project work has not been submitted earlier for the award of any degree or diploma to the best of my knowledge and belief.

Dr. Rajender Kumar	**Dr. Saurabh Goel, Mr. Amit Batra**
(Project Guide)	(Project Co-ordinator)

Prof. Dr. S C Gupta

HOD (CSE)

PIET, Samalkha

Table of Content

List of Tables

Table of Figures

Chapter 1

INTRODUCTION

1.1 Topic of the System

Title: - “City Sense Mobile Application”

1.2 Project Abstract

The era of mobile technology opens the windows to the android app. The websites are vanishing and the mobile phones are emerging. It’s the time to change from conventional websites to apps, which has become the part of our daily routine. We are introducing ‘**CITY SENSE’** the android application. Mobile application to provide all generic public information of selected city of India. It gives us more comfort and a better user interface The app aims to provide residents and visitors with a comprehensive platform for accessing up-to date information about the city, including points of interest, events, and public transportation schedules. Through an examination of the app's features, benefits, and potential impact, this study sheds light on the transformative potential of such technology in improving urban connectivity and fostering a sense of community. The findings suggest that the city information app holds great promise for enhancing the quality of life in cities by empowering individuals with valuable information and facilitating their engagement with the urban environment.

1.3 Purpose of the System

The main purpose defined behind developing a city guide app is offering comprehensive and up-to-date information to not only the tourists but even the local residents of the city. This information would include everything such as the main tourist attractions, the places of sightseeing, hotels, restaurants, etc. In addition, you also get to know about the how to reach to that particular city through all the mode of transport. You also get familiar with the major cultural events or any other celebration occurring around the city. For instance, if you are travelling to Vishakhapatnam. So, even if someone is visiting the city or place for the first time, he/she won’t face any problem in availing the important info. It also makes it easier for the tourist to plan is trip accordingly.

1.4 Target User

Whenever any product is planning to be made there is always a background study on the target users.

- **The Local and Foreign Tourists**

  The app is readily helpful for both the local as well as the foreign nationals who have visited the place for holidaying.

- **The Businessmen and Corporate Houses**

  Many entrepreneurs visit new places to attend corporate meetings or any other business event. The City Sense app will give them the exact information of the venue and easiest possible way to reach there.

- **Tour Operators and Travel Agencies**

  Even the tour operators and travel agencies are not lacking behind in reaping the benefits of the city guide app. They can check the in hotels, restaurants etc.

For using this system one just need to have a smart phone with internet access and some basic knowledge of using it.

1.5 Topic Background

A city sense app is a mobile application that provides users with information about a particular city. The app typically includes features such as maps, transportation information, restaurant etc.

Travel apps can be useful for both tourists and locals. Tourists can use the app to navigate an unfamiliar city and find popular tourist attractions, while locals can use the app to discover new places to visit and things to do in their own city.

Many Travel apps also include social features that allow users to connect with other travellers or locals and share recommendations or experiences. Some apps also offer personalized recommendations based on the user's interests or previous activity within the app.

1.6 Problem Context

The problem context of a city guide app involves addressing various challenges and pain points that users may encounter when exploring a new city or seeking information about their own city. Here are some common problem areas that city guide apps aim to solve:

- Information overload: Cities are filled with a multitude of attractions, restaurants, events, and points of interest. It can be overwhelming for users to filter through this vast amount of information and make informed decisions about where to go and what to do.

- Lack of local knowledge: Travellers, especially those visiting a new city, often lack knowledge about the best places to visit, dine, or stay. They may miss out on hidden gems or unique experiences simply because they are unfamiliar with the local scene.
- Inefficient navigation: Finding one's way around an unfamiliar city can be challenging. Users may struggle with public transportation options, directions, and the most efficient routes to their desired destinations.
- Outdated or inaccurate information: City guidebooks or online resources may provide outdated information about attractions, restaurants, or events. Users need access to reliable and up-to-date information to ensure their plans align with the current offerings.
- Lack of personalization: Every individual has different preferences, interests, and requirements. A one-size-fits-all approach may not cater to the specific needs of users. Personalization is essential for delivering relevant recommendations and suggestions tailored to each user's preferences.
- Limited social interaction: Traveling can be a social experience, and users may seek opportunities to connect with like-minded individuals or receive recommendations and insights from locals or fellow travellers.
- Language and cultural barriers: Visitors to a new city may face challenges due to language barriers or cultural differences. They may require assistance in finding language support or understanding local customs and norms.

City guide apps strive to address these problem areas by providing comprehensive, user-friendly, and personalized solutions. They aim to enhance the overall experience of users by offering accurate information, efficient navigation, etc.

1.7 Rationale behind the System: Why do we need this System?

The problems that are traced needs to be considered and the researcher has decided to construct such an application that would focus on providing solutions for the above-mentioned problems.

1.7.1 Benefits of the System: What are the potential benefits derived?

Upon implemented, the system could bring about significant tangible and intangible benefits:

- ➯ **Tangible Benefits**

S. No.	Tangible Benefits	Description
1.	**No investment**	Users doesn't have to pay anything for using this system on android mobile Everything in this system is free of cost.
2.	**Accessibility of system**	This system can be accessible anywhere if the user has got the mobile.
3.	**Convenient access to information**	This convenience saves time and effort compared to searching for information through multiple sources or guidebooks.
4.	**Accurate and up-to-date information**	City Sense apps strive to provide accurate and up-to-date information about various aspects of a city.
5.	**Efficient navigation and directions**	City guide apps often include maps and navigation features, helping users navigate through a city efficiently.

Table 1: Tangible Benefits

➪ **Intangible Benefits**

S. No.	Intangible Benefits	Description
1.	**Educational and informative value**	City sense apps can provide valuable educational information about the history of city.
2.	**Time Saving**	This system ensures that user gets the value of their time and don't have to rely upon anybody else for placing their order.
3.	**Friendly User Interface**	The application that is developed would facilitate its end users with interactive graphical interfaces that would be easily adaptable.

4.	**Customer Satisfaction**	The user interface will be user-friendly and easy to use which will result in improved satisfaction of the customers.
5.	**Time-saving and stress reduction**	By providing comprehensive information and efficient navigation features, city sense apps help users save time and reduce stress.
6.	**Discovery of hidden gems**	Users can uncover hidden gems that may not be prominently featured in mainstream travel resources.

Table 2: Intangible Benefits

1.8 Objectives of the System

1.8.1 Project Objectives

The project objective of the City Sense app is to create a comprehensive and user-friendly mobile application that serves as a reliable and intuitive city guide for residents and visitors. The app aims to enhance the experience of exploring a city by providing accurate information, personalized recommendations, efficient navigation, and social interaction features. The specific objectives of the project may include:

- Information: Develop a robust database of attractions, restaurants, transportation options, and other points of interest within the city. Provide users with up-to-date and accurate information about each location, including details like operating hours, descriptions, pricing, and user reviews. Implement recommendation algorithms to offer personalized suggestions based on user preferences, interests, and past activity within the app.
- Navigation and Directions: Incorporate maps and navigation features to help users easily navigate the city. Enable users to access directions to their desired destinations, find the most efficient routes, and receive real-time updates on traffic or public transportation options.
- User-Friendly Interface: Design an intuitive and user-friendly interface that enables users to easily navigate through the app and access the desired information. Ensure a seamless and visually appealing user experience across different devices and screen sizes.

- Data Accuracy and Maintenance: Establish processes to ensure the accuracy and timeliness of information within the app. Regularly update the database, verify user-generated content, and incorporate user feedback to maintain high data quality.

The project objective of the City Sense app is to create a valuable tool that empowers users to make informed decisions, efficiently explore the city, connect with others, and enhance their overall experience of discovering and navigating a new city.

1.8.2 Learning Objectives

The proposed system will allow the developer to attain relevant knowledge and concepts in specializing android based software development and authoring tool. The objectives are disreputably emphasized on its concepts and ideas rather than trained expertise. The developer would have to undergo various sections of software development, project management, human computer interaction, usability factors along with the mechanisms of development principles so as to produce highly reusable quality software on time. But the prime aim to undergo mobile computing concepts is to increase preferred output with user satisfaction within less effort and time to encounter an easy solution for the city sense and reduce all the efforts that are required for planning for a trip.

The following objectives listed below are of prime importance to the developer: -

- Gaining knowledge of android mobile based development, android technology and XML.
- Learning the concepts of Services such as Location Based Services, weather API implementation.
- Learning how to deal with mobile based software and development along with its database management.
- Able to perceive real time application with their integration on mobile devices.
- Implementation and integration of database services with mobile applications.
- Understanding of data gathering methods and its actual implementation.
- Learning queries for updating data from the mobile to server.
- Learning and Practices of Jason.

1.9 Scope of the System

The proposed system will be beneficial in a number of ways. The scope of “**City sense mobile application**” refers to the boundaries and extent of its functionalities, features, and components. The scope outlines what the system will include and what it will not include.

- User Registration and Profiles
- City Information and Database
- Search and Filtering
- Maps and Navigation
- User-Friendly Interface
- Data Accuracy and Maintenance
- Additional Features :: weather search , currency converter

It's important to note that the specific scope of the City Sense app may be adjusted based on project requirements, resource constraints, and development timelines.

1.10 Limitation of the System

Limitations are always a part of every project. The project scope is limited to a confined boundary as listed below:

- The system is being developed for only android based phone.
- The customer will need the internet connection.
- The application will be run after installation in the smart phone.
- To run app in android smart phone require at least android 7 to install app.

1.11 Assumptions Made

This system is although easy to use in terms of its functionalities but even then, there are some assumptions that need to be made before the development of the system:

- Users need to have basic knowledge of using android phones
- Good knowledge of English and basics of Internet is required, i.e. user needs to be a computer literate.

1.12 Success Criteria

Success Factors depends upon the depth of understanding and experience gained from the applications viewed for the review and how efficiently the developer cracks the problems faced in the application that lead to development of such an application.

- Application is properly working and compatible with mentioned android OS. Application will be useless if it is unable to run properly on the physical device. There should not be any anomalous or false functioning during its working on physical device.
- System meets all requirements of the proposed solution. If the application provides desired operations and produces expected output i.e. if it is meeting user requirements,

passing of above test cases. Application must fulfil the core as well as enhanced system requirements mentioned above.

- Project will be treated as constructive if it is proper evaluation of Usability & HCI. The developer will be able to provide successful implementation of mobile application in terms of interactivity, effectiveness and efficiency.

1.13 Project Scheduling and distribution of Work among team member

TOTAL DURATION	12 weeks	START DATE	14th FEB, 2023	END DATE	8th MAY,2023

PHASE AND DURATION	**TASKS**
PROJECT DEFINITION *Duration: 1 weeks*	↳ Idea Generation ↳ Project Title Selection ↳ Abstract Draft Project Proposal ↳ Project Proposal Form
PROJECT PLANNING *Duration: 1 weeks*	↳ Work Breakdown Structure ↳ Schedule and Time Estimation
REQUIREMENT ANALYSIS *Duration: 3 weeks*	↳ **Define and Finalize Requirement Specification** » Project Background » Problem Context » User Requirements » Set Objectives » Identify Scope of Project » Features and Functionalities ↳ **Organizing Project Specification Form** ↳ **Research and Analysis** » **Research** ✓ Academic Research ✓ Secondary Research ✓ Human Computer Interaction » **Analysis** ✓ Domain Analysis

	✓ Existing System Analysis ✓ User Requirements ✓ User Profiling and Modelling ✓ Risk Analysis
SYSTEM DESIGN *Duration: 2 Weeks*	↳ **Navigational Design** » Storyboarding ↳ **Abstract Interface Design** » Mobile Application Architecture » Functionality Design » Interactivity Design ↳ **Design for Test Plan** » Acceptance Test » System Test » Interface Test
PROTOTYPING *Duration: 1 weeks*	↳ Creating Prototypes ↳ Evaluate Prototype feedbacks
PRODUCTION AND IMPLEMENTATION *Duration: 2 weeks*	↳ Program Code Generation ↳ Integrating Backend and Front end ↳ Module creation and Integration ↳ Implementation of working system
TESTING & EVALUATION *Duration: 1 weeks*	↳ Prototype Evaluation ↳ Test Plans ↳ Unit Testing ↳ Integration Testing ↳ System Testing ↳ Critical Evaluation
PROJECT ENDING *Duration: 1 weeks*	↳ Submission of the finished product

Table 3:: Development Plan

CHAPTER 2

PROBLEM DESCRIPTION

2.1 Description of the current issue

City Sense is a mobile application where people can access all information about the city, providing details of hotels, restaurants, tourist destinations, bus stops, train stations, airports and other features such as Geolocation. Navigation, CurrencyExchange and Directions.

2.1.1 Problem Area Description

The City Information Application aims to solve the problem of providing residents and tourists with important and up-to-date information about the city.

In general, the problem area addressed by using city information is to provide basic, easy and accessible information about the city and its services to residents and visitors.

2.2 Importance and cause of the problem

Today, people's eating habits are constantly evolving. There have been many people who went abroad to have fun and enjoy themselves. Tourism is the strongest and largest industry in the world economy, accounting for around 11% of global gross domestic product (GDP), employing 200 million people and serving 700 million tourists worldwide - this figure is expe cted to increase by 2020. Couple.

First, the travel information available to travelers on the internet is very good. However, another problem arose, namely that tourists cannot easily access their travel information while traveling. Therefore, we aim to explore how to create an understanding of the city base on the Android platform to solve this problem. Mobile apps play an important role in many areas to fulfill their carriers. A mobile application or mobile application is a computer or software program designed to run on a mobile device such as a phone/tablet or monitor. The proposed system is designed to be more efficient than the current system in terms of memory, performance and the ability to achieve better results than existing systems.In our state, many people who want to get information about the city go to the website. It helps users to view their city information online in a mobile application. With our City Sense app, users can view information anytime, anywhere. The system will have the location analysis of the city, analysis based on the difference between people, will have the dataset, and in the dataset, each person will respond to the costs in different areas, so collecting the values of some datasets at the same time of the year can help people evaluate themselves using the dataset. The information will be easy for people to understand, a simple representation of the information used, so looking at the output will help people understand the city.

From the results, ordinary people can learn about things that need improvement, which helps people understand the city. Our project is important for emerging businesses as well as for ordinary people.They can be compared with each other using data from anywhere. The project was designed and developed to create travel guides for cities in India for the convenience of domestic and foreign tourists. Tourists face many problems due to lack of proper guidance. Traditionally, tourists should hire professionals when they visit our country. This guide describes the city. We have to pay a lot of money to get such services from professional tour operators. For most tourists, this is expensive. Most of the tourists are unprofessional as they work part-time as tour guides during the summer and work elsewhere during the winter. Therefore, sometimes due to human nature, tourist guides provide information on temperature, weather, date and date, etc., necessary to make a decision. does not provide essential information to travelers who forget facts and pictures. Visitors can take it for free whenever they want.

2.3 Applications

The application of urban data can be versatile in solving problems of the past. These features will include:

Interactive Maps and Directions: The app will provide users with maps and directions for popular locations, including walking, biking, car and public transportation.

The app can also provide real-

time traffic updates to help users navigate the city more efficiently.

Restaurants and Recommendations: This app can provide recommendations for local restaurants and tourists based on their preferences and location.

Maps and Navigation: Includes an interactive city map that allows users to easily find their way around. Enable GPS to provide specific directions to places such as restaurants and famous places such as hotels.

Local Weather: This lets users know current conditions and helps them plan their activities accordingly.

Dinner and Nightlife: contains a list of restaurants, cafes, bars and clubs, including menus, reviews and ratings. Let users book appointments or order food directly from the app.

Comments and Ratings: Allows users to provide feedback, rate their preferences, share their experiences, and contribute to the app's content. This fosters a sense of community and helps other users make informed decisions.

In general, the city's information application solution is to provide residents and tourists with a general user interface to get information about the city and its services.

The app should be interactive, personalized and engaging, and provide users with real-

time updates and recommendations.

2.4 Nature of the Challenges

The nature of the challenges when developing an urban education application is as follows:

2.4.1 Domain Challenges:

To develop this application, the researcher has to go through some important problems during development. , for example:

Discussion Points: Understand the terms of interaction between humans and mobile devices, and when creating apps, remember to follow all these principles for app success.

• Mobile Marketing: Use Mobile Marketing related techniques and strategies to create mobile applications

• User Registration Module: In registration, users will register by providing their important in formation such as email id, username, mobile phone, Attributes) to register.

2.4.2 Technical Challenges:

The challenges that researchers face while working in the field are:

• Android Platform: The first thing I want to talk about here is the platform knowledge. Since the system will be built on the android platform, of course I started to learn about this technology when I didn't know anything about it, but I still need to learn a lot of subjects on Android to complete this project.

• Hardware Fragmentation: Android platform runs on many devices with different characteristics such as memory, CPU speed and graphics resolution. Fragmentation is also present at the operating system level. A famous example is the breakdown of Android devices with different screens and resolutions. That's why manufacturers should do market research to find out which phones are most popular among their audience and develop them for users first.

• Software Fragmentation: Because there are many Android versions running on the device. Therefore, as a developer, I cannot focus on a single version or the latest version because it is not easy for users to update their operating system, so many have never had it. That's why it's important for developers to understand which operating systems are the most popular and develop them keeping in mind the latest updates.

• Integrating various modules in one application: Database integration, user registration module, search module, etc. Integrating various modules such as somewhere it will be very difficult. Developers need to learn many APIs to use the system.

2.5 Feasibility Study

Feasibility Study All studies are carried out professionally, efficiently and economically with the aim of determining whether the project is viable in all aspects, yes or no. If the project is not viable, we try to make improvements or consider changes that will benefit, to make the project viable and useful for all users.This study basically determines the functioning, functioning and testing of a project or
software.

2.5.1 Technical Feasibility

A project or software is said to be viable if:-

✓ The equipment is easy to use.

✓ Technology can be easily adapted to new versions.

✓ Tools used in the project so that the project can do what it wants.

✓ This product can answer questions instantly.

✓ Technology provides security in every way.

"City Sense" is created using such technology, which can be simple and familiar with its security implementation, while also providing immediate results and appropriate actions for the user to take.

2.5.2 Economic Feasibility

A project is said to be profitable if:-

✓ The project does not require additional expenditure.

✓ This can be considered as a good investment for the user or the organization.

"City Sense" not only saves time and money, but also reduces paper usage. Once the system is registered, it can be used for a long time and becomes useful.

2.5.3 Scheduling Feasibility

A project or software is considered viable if:-

✓ The project builds in real time with sufficient resources allocated to each phase of the development process.

✓ Clear Requirements

✓ A dedicated team with the right expertise and skills.

✓ There are valuable resources.

"City Sense" is too big. You can get information about all cities at the same time, whether online or offline.

Project meets everything because it meets technology, time and economy.

2.6 Conclusion

A city information app can be a useful resource for residents and tourists, providing a central, comprehensive and useful resource for planning access to information about the city and its services. Building a successful urban data application must overcome many challenge s, including data collection and integration, user collaboration, limitations, privacy and security concerns, and financial and security concerns.

Realtime, adequate resource allocation, clear plans and policies must be established to ensure the success of urban information use. Applications should be designed to be intuitive, easy to use, and provide value to users. It should be compatible with different operating systems and devices and include features such as interactive graphics and instructions, event calendars, home suggestions, food and preferences, feedback and interaction platform, and personal.

CHAPTER 3

LITERATURE REVIEW

3.1 Introduction to Literature Review

The era of mobile technology opens the windows to the android app. The websites are vanishing and the mobile phones are emerging. It’s the time to change from conventional websites to apps, which has become the part of our daily routine. We are introducing ‘CITY SENSE’ the android application. Mobile application to provide all generic public information of selected city of India. It gives us more comfort and a better user interface The app aims to provide residents and visitors with a comprehensive platform for accessing up-to-date information about the city, including points of interest, events, and public transportation schedules. Through an examination of the app's features, benefits, and potential impact, this study sheds light on the transformative potential of such technology in improving urban connectivity and fostering a sense of community. The findings suggest that the city information app holds great promise for enhancing the quality of life in cities by empowering individuals with valuable information and facilitating their engagement with the urban environment.

3.2 Advanced Preliminary Research

- **The Scope and Objective:** With the help of this app you can easily get the specific information about the various cities like restaurants and hotels, famous places and transportation.
- Organize the information: Create a structure or framework for organizing the information you have gathered. Categorize the information based on different aspects,

such as attractions, dining, transportation, events, and accommodations. Develop a system to tag or label the information for easy retrieval.

- Conduct a comprehensive search: Use search engines, databases, and other resources to gather information about your city. Search for official websites, local directories, travel blogs, and other platforms that contain relevant information about the city's attractions, services, and events.
- Interactive maps and directions: The app could provide users with maps and directions to popular destinations, including walking, biking, driving, and public transportation options. The app could also provide real-time traffic updates to help users navigate the city more efficiently.
- Restaurant and attraction recommendations: The app could provide users with recommendations for local restaurants and tourist attractions based on their preferences and location.
- Maps and Navigation: Include Interactive maps of the city, allowing users to find their ways easily. Enable GPS functionality to provide direction to specific location, such as restaurants and hotels famous places etc.
- Local Weather: This keeps Users informed about currents events and help them to plan their activities accordingly.
- Dining and Nightlife: Include a directory of restaurants, cafes, bars, and clubs, complete with menus, reviews, and ratings. Enable users to make reservations or order food directly through the app.

- User Feedback and Ratings: Allow users to provide feedback, rate attractions, share their experiences, and contribute to the app's content. This fosters a sense of community and helps other users make informed decisions.

Overall, the proposed solution for a city information app is to provide a comprehensive and user-friendly platform for residents and visitors to access information about the city and its services. The app should be interactive, personalized, and engaging, and provide users with real-time updates and recommendations.

3.3 Domain Research

3.3.1 Android

Android is a mobile operating system developed by Google, based on a modified version of the Linux & kernel and other open source software and designed primarily for touchscreen mobile devices such as smartphones and tablets. In addition, Google has further developed Android TV for televisions, Android Auto for cars, and Wear OS for wrist watches, each with a specialized user interface. Variants of Android are also used on game consoles, digital cameras, PCs and other electronics.

Initially developed by Android Inc., which Google bought in 2005, Android was unveiled in 2007, with the first commercial Android device launched in September 2008. The operating system has since gone through multiple major releases, with the current version being 9 "Pie", released in August 2018. The core Android source code is known as Android Open Source Project (AOSP), and is primarily licensed under the Apache License.

Android is also associated with a suite of proprietary software developed by Google, called Google Mobile Services (GMS) that very frequently comes pre-installed in devices, which usually includes the Google Chrome web browser and Google Search and always includes core apps for services such as Gmail, as well as the application store and digital distribution platform Google Play, and associated development platform. These apps are licensed by manufacturers of Android devices certified under standards imposed by Google, but AOSP has been used as the basis of competing Android ecosystems, such as Amazon.com's Fire OS, which use their own equivalents to GMS.

Android has been the best-selling OS worldwide on smartphones since 2011 and on tablets since 2013. As of May 2017, it has over two billion monthly active users, the largest installed base of any operating system, and as of June 2018, the Google Play store features over 3.3 million apps.

Fig. 1:: Android Logo

3.3.2 JAVA

Java is a programming language and computing platform first released by Sun Microsystems in 1995. 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 is a general-purpose computer-programming language that is concurrent, classbased, object-oriented, and specifically designed to have as few implementation dependencies as

possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2016, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James
Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntaxfrom C and C++, but it has fewer low-level facilities than either of them.

The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (byte code compiler), GNU Classpath (standard libraries), and Iced Tea-Web (browser plugin for applets).

The latest version is Java 11, released on September 25, 2018, which follows Java 10 after only six months in line with the new release schedule. Java 8 is still supported but there will be no more security updates for Java 9.Versions earlier than Java 8 are supported by companies on a commercial basis; e.g. by Oracle back to Java 6 as of October 2017 (while they still "highly recommend that you uninstall" pre-Java 8 from at least Windows computers).

Fig. 2:: JAVA Logo

3.3.3 Firebase

Firebase evolved from Envolve, a prior startup founded by James Tamplin and Andrew Lee in 2011. Envolve provided developers an API that enables the integration of online chat functionality into their websites. After releasing the chat service, Tamplin and Lee found that it was being used to pass application data that weren't chat messages. Developers were using Envolve to sync application data such as game state in real time across their users. Tamplin and Lee decided to separate the chat system and the real-time architecture that powered it. They founded Firebase as a separate company in April 2012.

Firebase Inc. raised seed funding in May 2012. The company further raised Series A funding in June 2013.In October 2014, Firebase was acquired by Google. In October 2015, Google acquired Divshot to merge it with the Firebase team. Since the acquisition, Firebase has grown inside Google and expanded their services to become a unified platform for mobile developers. Firebase now integrates with various other Google services to offer broader products and scale for developers. In January 2017, Google acquired Fabric and Crashlytics from Twitter to join those services to the Firebase team. Firebase launched Cloud Firestore, a Document Database, in October 2017.

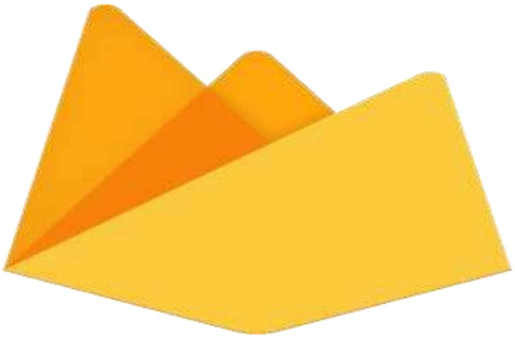


Fig. 3:: Firebase Logo

3.4 Android Structure

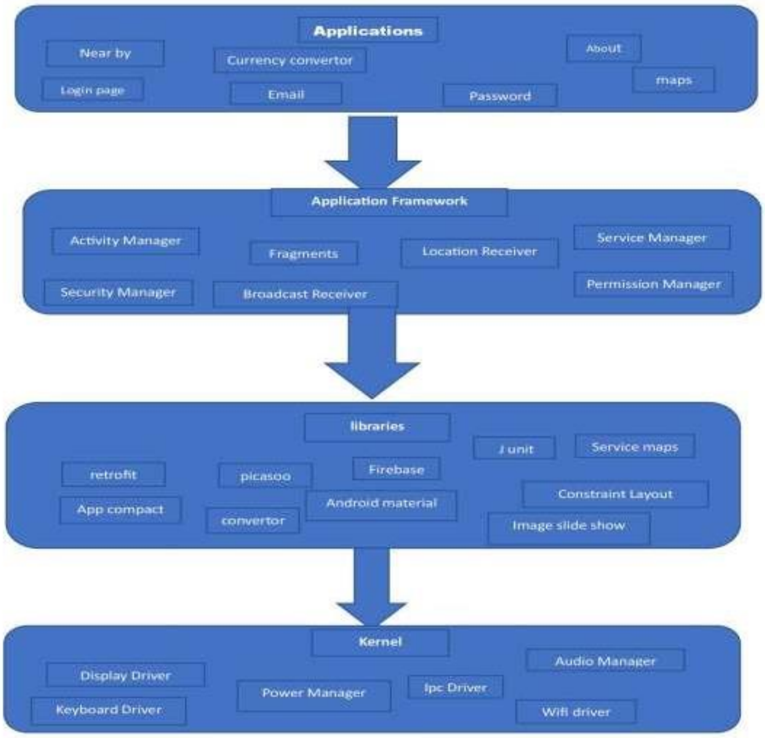


Fig. 4 :: Android Structure

3.5 Security and Permission

Identify Necessary Permissions: Determine the specific permissions required by your app based on its features and functionality. Common permissions for a city information app may include:

Location: If your app provides location-based services, such as finding nearby attractions or providing directions, you'll need to request the ACCESS_FINE_LOCATION or ACCESS_COARSE_LOCATION permission.

Internet: To retrieve data from remote servers or APIs, you'll need the INTERNET permission.

Storage: If your app needs to save data locally or access external storage, you'll require the READ_EXTERNAL_STORAGE and WRITE_EXTERNAL_STORAGE permissions.

Camera: If your app allows users to take photos or upload images, you'll need the CAMERA permission.

Network State: If your app needs to check the network connectivity status, you'll need the ACCESS_NETWORK_STATE permission.

Other permissions: Depending on additional features or services your app offers, you may need other permissions such as ACCESS_WIFI_STATE, VIBRATE, or RECEIVE_BOOT_COMPLETED.

Request Permissions at Runtime: Starting from Android 6.0 (Marshmallow), permissions are requested at runtime instead of during the app installation process. Implement the necessary code to request permissions from the user when they are needed. Clearly explain to users why each permission is required and how it benefits their experience.

Handle Permission Denials: Account for scenarios where users may deny certain permissions. Gracefully handle these situations by disabling or adjusting app features that depend on the denied permissions. Provide clear explanations and instructions to users on how to manually grant the required permissions through the app settings.

Secure Network Communication: Implement secure communication protocols, such as HTTPS, when retrieving data from external servers or APIs. Verify the authenticity of SSL/TLS certificates and consider implementing certificate pinning to prevent man-in-themiddle attacks.

User Authentication and Authorization: If your app includes user accounts or personalized features, implement secure authentication mechanisms. Utilize secure password storage practices, such as hashing and salting, and consider options like OAuth or OpenID Connect for secure sign-in with trusted providers. Implement appropriate authorization checks to control access to sensitive data or actions.

Secure Data Storage: Protect sensitive user data stored on the device by utilizing secure storage options provided by Android. Use encryption to safeguard sensitive information, including user preferences or cached data.

Input Validation and Sanitization: Implement proper input validation and sanitization techniques to prevent common security vulnerabilities such as SQL injection or cross-site scripting (XSS) attacks. Ensure that user inputs are validated on the client-side and properly sanitized before interacting with databases or remote servers.

Code Security and Testing: Follow secure coding practices to avoid common security pitfalls. Regularly perform security testing, including code reviews, vulnerability assessments, and penetration testing, to identify and address potential vulnerabilities in your app.

Privacy Policy and User Consent: Create and display a privacy policy within your app that clearly explains the data you collect, how you use it, and the measures you take to protect user privacy. Obtain appropriate user consent for data collection and sharing, especially if you integrate third-party services or collect personally identifiable information.

Stay Updated: Stay informed about the latest security threats, best practices, and updates related to Android security. Keep your app and its dependencies up to date with the latest patches and security fixes.

By implementing these security and permission management measures, you can enhance the security of your city information app, protect user data, and build trust with your users.

3.6 Software and Hardware Requirements

Software Requirements	
Operating System	Windows/Linux
Frontend	JAVA & ANDRIOD
Backend	Firebase
Development Environment	Andriod Studio

Table 4 :: Software requirements

Hardware Requirements	
Processor	Pentium
RAM	2GB (minimum)
Hard disk	20GB (minimum)
Architecture	64-bit

Table 5: Hardware requirements

3.7 Review of Literature

My City Information Mobile Application using Android Application	[1] K. Leela Rani, T. Anne Esther Jashmine, S. Benita Jeba Mahar, “My City Information Mobile Application using Android Application,” International Journal of Engineering Science and Computing, 9(30, pp.20288-20290, 2019. Jian Meng,Neng Xu ,“A Mobile Tourist Guide System Based on Mashup Technology “ ISBN978142447613/10©2010IEEE [2] S. Karanasios, S. Burgess, C. Sellitto, A Classification of Mobile Tourism Applications, and Chapter in book: Global Hospitality and Tourism Management Technologies, USA: IGI Global, 2012.

Vehicle Tracking and Locking System Based on GSM and GPS	[1] R. Ramani, S. Valarmathy, Dr. N. SuthanthiraVanitha, S. Selvaraju, M. Thiruppathi, R. Thangam, " Vehicle Tracking and Locking System Based on GSM and GPS", MECS I.J. Intelligent Systems and Applications, 2013, 09.

Design and Development of GPSGSM Based Tracking System with Google Map Based Monitoring	[1] Pankaj Verma, J.S Bhatia, "Design and Development of GPSGSM Based Tracking System with Google Map Based Monitoring", International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.3, No.3, June 2013

Table 6 ::Review of Literature

CHAPTER-4

Research Method

To conduct research for a city sense app, you can employ a combination of qualitative and quantitative research methods

4.1 Technical Research::

This is related to the technical part of the research means research related to selection of platform, programming language, database, technology and methodology. Conducting technical research for the proposed project is a critical factor that can affect the implementation of the project and thus is conducted before the system design and implementation phase begins. A wide-spread research is conducted that compares various platforms, programming languages, databases, technologies, methodologies and evaluates them to come to conclusion which would be chosen for the proposed project.

Justification for Choosing Android as a Platform

The decision of choosing one platform over other has always been a difficult task for the developer. Each platform has its own pros and cons so it is inadequate to say that one platform is better than other ones. Let’s see why **Android** has been chosen over other OS platforms.

What is Android?

Android is a platform that provides tools and technologies which can be used to develop and build mobile applications. The android platform is an open source that includes an operating system, middleware services and also key applications for use in mobile devices. Android environment uses the Linux operating system at its core has a kernel based on Linux kernel. Its middleware, libraries and APIs are written in core C while the applications are developed for android using java with android class extension.

Why Android?

The researcher opted to work on Android technology for the development of application because of the following reasons:

- **Huge Market:** According to **IDC Research, Inc. (2015), Android** has dominated the smartphone market with a share of 82.8%.Thus, being so popular in the market, the researcher believes that the success rate would be high, if the development is based on this technology.

Period	Android	iOS	Windows Phone	BlackBerry OS	Others
2016	90%	7%	2%	0.6%	0.4
2015	82.8%	13.9%	2.6%	0.3%	0.4%
2014	84.8%	11.6%	2.5%	0.5%	0.7%
2013	79.8%	12.9%	3.4%	2.8%	1.2%
2012	69.3%	16.6%	3.1%	4.9%	6.1%

Table 7:: OS Smartphone Share

- **Android Support with Google-Map:** Android and Google Map both are initiatives of Google itself and for ease of use they have provided the packaged libraries bundled with Android platform so use of Google-Map along with GPS technology in android application is without compatibility issues, and this compatibility can be used in the proposed system to get the current location.

- **Code Reusability Using Android Reuse Models:** Even complex applications can be built easily and rapidly using Android because Android platform allows reusability of code by providing **different Android Reuse Models**. This helps to improve the overall quality of the app and saves lot of developer's time too. The following are the reuse models provided by android:
  a) The APK (Android Package Kit)
  b) The JAR (Java Archive)
  c) The Android Library Project
- **Rich Development Environment: Android** has rich application development environment that enables effortless convenience to develop most attractive applications easily and in lesser time. And since the application needs to be made in the given time constraint it's a great choice.
- **Open Source Architecture:** Since android is an **open source platform**, less investment will be required in the development portion. And any software or program

that is produced needs to be tested which can be done without much investment. Only required investment would be a physical device to test the system.

- **Java in its Application Development Environment:** Since, android apps are developed using Java therefore all the features (Simplicity, Robustness, Portability, Multithreading, and Security etc.) that are associated with Java gets embed with the android development and gives android development an edge. Moreover, since the developer is well versed in Java, android becomes a great venture.

4,2 Comparison of Android with other Platforms

The following tables would more justify why android is chosen over other platforms:

Features	Android	IOS	Windows Phone
Company	Google	Apple Inc.	Microsoft
Programmed in	C, C++, JAVA	C, C++, Objective-C	Visual C#
Dependent on a PC or MAC	No	No	No
Source Model	Open Source	Closed, with open source components	Closed
Customizability	A lot can change almost anything.	Limited unless jailbroken	Difficult
Working state	Current	Current	Current

Table 8 ::Comparison between OS based upon features

Criteria	Android	IOS	Windows Phone	Blackberry
Level of Security	×	✓	✓	×
Support for Media Formats	✓	✓	✓	✓
Open Source	✓	×	×	×
Multitasking	✓	×	×	×
Market Demand	✓	✓	✓	✓
Support for Third Party Apps	✓	×	×	×

Adobe Flash Support	✓	×	✓	×
IDE and Tools Available	✓	×	×	×
Support for Languages	✓	✓	×	×

Table 9 ::Comparison between OS based upon criteria

Justification for programming language selection (Java)

For the development of the proposed system, the developer has decided to use **Java** as a programming language. It’s an obvious thing to select so because the system is to be developed in Android platform which uses Java programming language in its development environment.

What is Firebase?

Firebase is a Backend-as-a-Service (BaaS) platform developed by Google. It provides developers with a set of tools and services to build and manage web and mobile applications more easily. Firebase offers various features that help developers with tasks such as data storage, authentication, real-time database synchronization, cloud messaging, hosting, and more.

Justification for Firebase

Firebase offers a robust and scalable platform for application development, providing developers with ready-to-use features, real-time updates, cross-platform support, authentication, security, analytics, and serverless architecture. These justifications make Firebase a popular choice among developers looking for a reliable backend infrastructure to accelerate the development process and deliver high-quality applications.

CHAPTER-5

Analysis

5.1 Analysis of Questionnaire

5.1.1 Overall Conclusion of Interview

1. **Develop a comprehensive city database:** Create and maintain a robust database of attractions, restaurants, events, and transportation options within the city. Ensure accurate and up-to-date information with user reviews and ratings.

2. **Implement real-time updates and synchronization:** Incorporate real-time data updates and synchronization capabilities to provide instant updates to users across devices.

3. **Prioritize cross-platform compatibility:** Build the app for web, Android, IOS, and Unity platforms, ensuring a consistent user experience and seamless synchronization across devices.

4. **Focus on authentication and security:** Implement a secure authentication system and data encryption to protect user information and ensure data privacy.

5. **Optimize performance and scalability:** Design the app to handle high user traffic and large datasets, optimizing performance and scalability through efficient coding practices and infrastructure management.

6. **Emphasize usability and user experience:** Prioritize intuitive navigation, a visually appealing design, and a seamless user experience to enhance user engagement and satisfaction.

By implementing these recommendations, the city guide app can effectively meet user needs, offer a superior user experience, and provide a competitive advantage in the market. Regular user feedback and continuous improvement through iterative development will further enhance the app's success.

CHAPTER 6

PROPOSED SYSTEM

6.1 Traditional State

As traditional practice when a tourist visits to our country they have to engage professional tourist guide. The guide provides the information about the city. We have to spend handsome amount of money to get such service of a professional guide. It is expensive for most of the tourist. Mostly guides are not professional because they working part time in summer season as guide and in winter they used to work in other fields. So sometimes the guide also could not give proper information to the travellers because of human nature they can't remember fact and figures which are requires for decision making like temperature, weather conditions, dates and historical etc. Due to insufficiency of proper tourist guide tourist face many problems.

6.2 Present State

When we are planning to go anywhere, you can search on internet which place is best .Suppose you decide to go Dehradun so you will need an hotel, restaurants and famous places before the developing of this app people have to face many problem you search on internet again and again but you cannot get a good result. Also you will not able to know the weather condition of the places and you can go anytime which makes your trip spoil.

6.3 Proposed System

The application aims to develop detailed texts, pictures, and other guidance information are provided, and so people can better understand the tourist attractions and make decision objectively. A problem is shown that visitors are not able to get travel information timely when they are on the move. Therefore, we purpose to explore how to build a mobile tourist guide system based android application to solve this problem.

Scope

- It support almost 90% smart phones.
- It can be use without sign-up process.
- Provide module to module interface.
- Easy to use and to find out locations.

b. Limitations

- It required a device (Smart Phone) Which should have installed android operating system
- Sometimes it requires internet connection while connecting to internet services.
- It cannot run on computer system without installing any virtual device.

Find destination location.

State wise service.

Main information about smart cities. ✓ Contact number and address for service

User Registration and Authentication:

- Allow users to create accounts or sign up using their email addresses or social media accounts.
- Implement secure authentication mechanisms to protect user accounts and personal information.

Home Screen:

- Provide a user-friendly and visually appealing interface with easy navigation.
- Display essential information about the city, such as weather updates, current events, or popular attractions.

City Guide:

- Include a comprehensive guide of attractions, landmarks, restaurants, and other points of interest in the city.

- Provide detailed information for each location, including descriptions, photos, contact details, opening hours, and user reviews/ratings.
- Implement search and filtering options to allow users to find specific places or categories.

Notifications and Alerts:

- Send push notifications to users about important updates, upcoming events, or personalized recommendations.
- Provide users with timely alerts for changes in event schedules, closures of attractions, or other relevant information.

✓ Accessibility:

- Ensure that the app is accessible to users with disabilities by following accessibility guidelines and implementing features like screen reader compatibility, font resizing, and color contrast options.

✓ Data Management:

- Establish a reliable and secure system to manage and update the app's content, ensuring that the information remains accurate, up to date, and reliable.

As interesting directions of future work we identify the following two lines. First covering access range can be increased, rating system can also be embedded according to the user satisfaction. Apart from android it can also be made for Windows and IOS users. Navigation system can also be integrate for a particular place

To provide a user friendly interface this will link the user with their nearby places. To have information about all general places like Hospitals, Main attraction of city, Police Station,

Famous Restaurants. With their corresponding location

6.4 Data Flow Diagram (DFD)

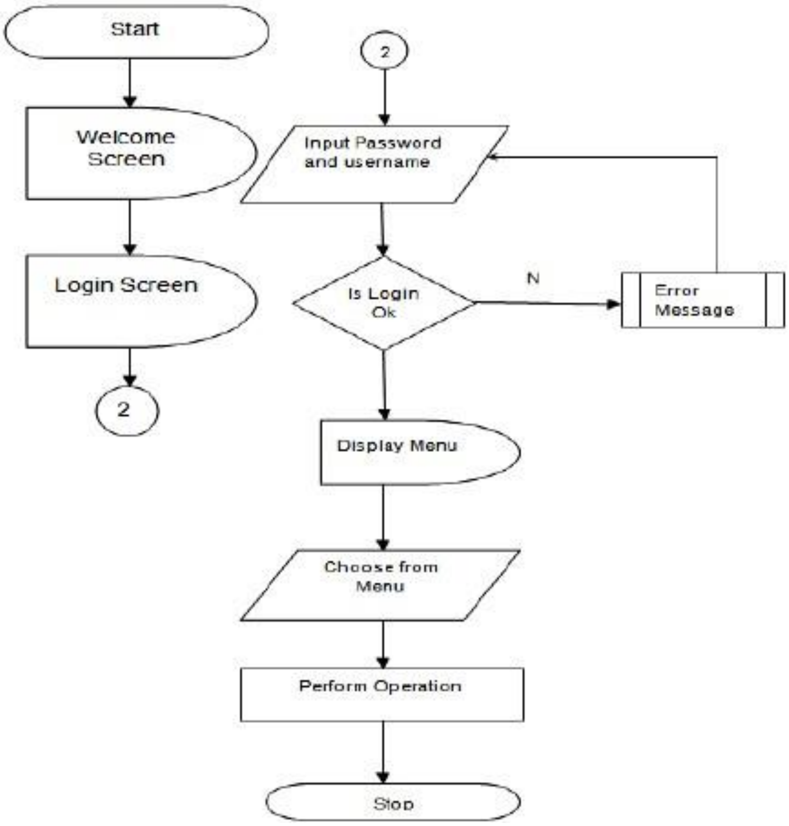


Fig. 6:: Level 1 DFD

6.5 ER Diagram

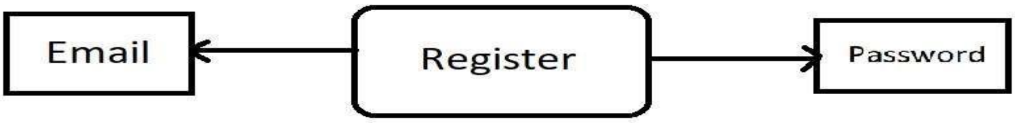


Fig. 7:: Register ER diagram

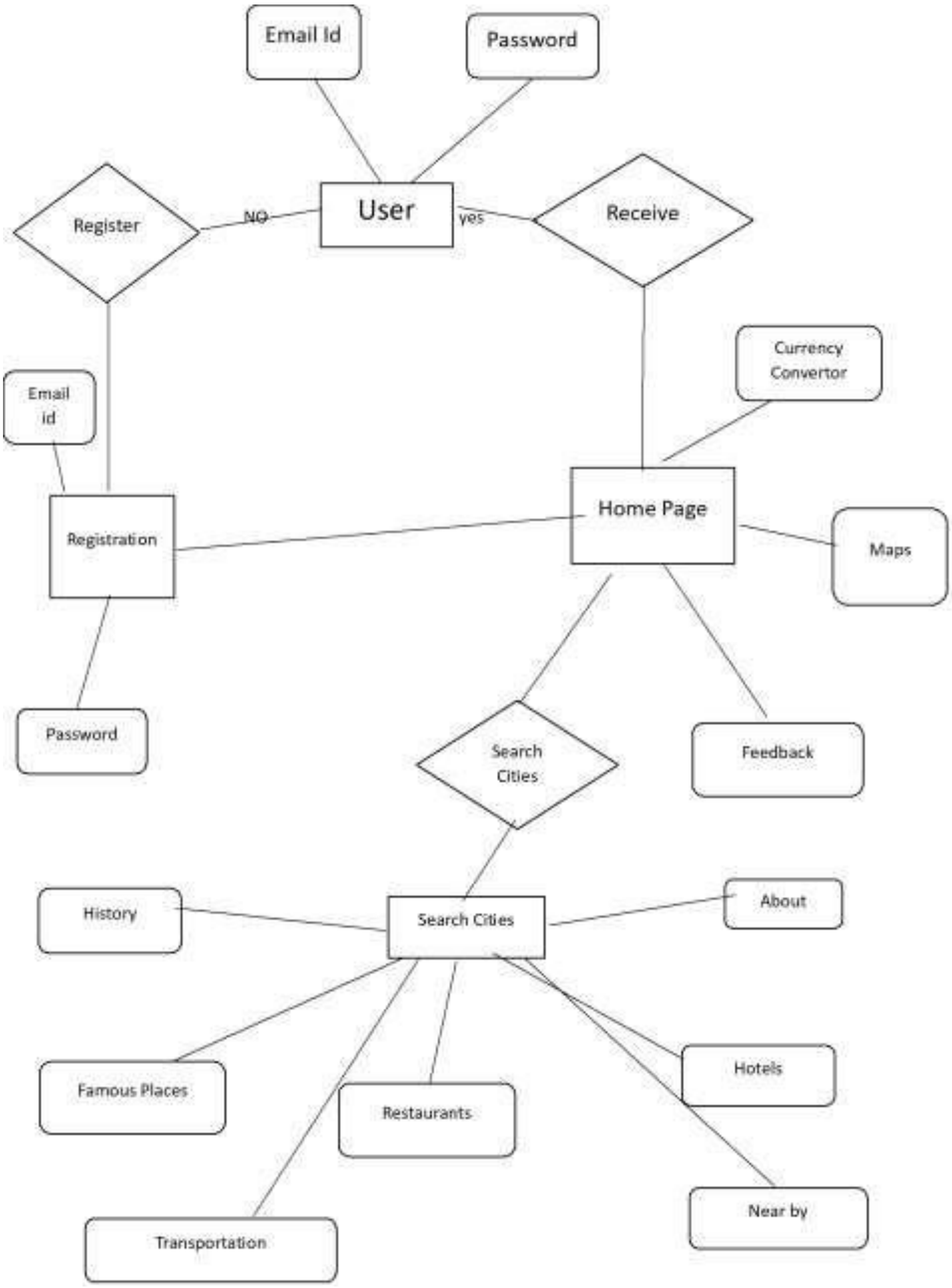


Fig. 8:: Module ER diagram

CHAPTER-7

CODING

The City Sense App comprises of following components:

- Register Component
- Login Component
- Feedback Component
- Home Page
- Navigation
- Map
- About
- Currency Exchange

7.1 Register Component:

7.1.1 Design :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
 xmlns:tools="http://schemas.android.com/tools"
 android:layout_width="match_parent"
 android:layout_height="match_parent"
 android:orientation="vertical"
 android:gravity="center"
 android:padding="15dp"
 tools:context=".Register">
 <TextView
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:textSize="40sp"
 android:textStyle="bold"
 android:gravity="center"
 android:layout_marginBottom="20dp"
 android:text="@string/register" />
 <com.google.android.material.textfield.TextInputLayout
```

```
android:layout_width="match_parent"
android:layout_height="wrap_content">
<com.google.android.material.textfield.TextInputEditText
android:id="@+id/email"
android:hint="@string/email"
android:layout_width="match_parent"
android:layout_height="wrap_content" />
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout
android:layout_width="match_parent"
android:layout_height="wrap_content">
<com.google.android.material.textfield.TextInputEditText
android:id="@+id/password"
android:hint="@string/password"
android:layout_width="match_parent"
android:layout_height="wrap_content" />
</com.google.android.material.textfield.TextInputLayout>
<ProgressBar
android:id="@+id/progressBar"
android:visibility="gone"
android:layout_width="wrap_content"
android:layout_height="wrap_content"/>
<Button
android:id="@+id/btn_register"
android:text="@string/register1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"/>
<TextView
android:id="@+id/loginNow"
android:text="@string/click_to_login"
```

```
 android:textStyle="bold"
 android:gravity="center"
 android:layout_marginTop="20dp"
 android:textSize="20sp"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"/>
</LinearLayout>
```

7.1.2 Java Code:

```
package com.project.myapplication;
import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.android.material.textfield.TextInputEditText;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
public class Register extends AppCompatActivity {
 TextInputEditText editTextEmail,editTextPassword;
 Button buttonReg;
 ProgressBar progressBar;
 FirebaseAuth mAuth;
 TextView textView;
 public void onStart() {
 super.onStart();
 // Check if user is signed in (non-null) and update UI accordingly.
 FirebaseUser currentUser = mAuth.getCurrentUser();
 if(currentUser != null){
 Intent intent = new
Intent(getApplicationContext(),HomePage.class);
 startActivity(intent);
 finish();
 }
 }
 @Override
 protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_register);
 mAuth = FirebaseAuth.getInstance();
 editTextEmail = findViewById(R.id.email);
 editTextPassword = findViewById(R.id.password);
 buttonReg = findViewById(R.id.btn_register);
 progressBar = findViewById(R.id.progressBar);
 textView = findViewById(R.id.loginNow);
 textView.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 Intent intent = new
Intent(getApplicationContext(),Login.class);
 startActivity(intent);
 finish();
 }
 });
 buttonReg.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 progressBar.setVisibility(view.VISIBLE);
 String email,password;
 email = String.valueOf(editTextEmail.getText()) ;
 password = String.valueOf(editTextPassword.getText());
 if(TextUtils.isEmpty(email))
 {
 Toast.makeText(Register.this, "Enter email",
Toast.LENGTH_SHORT).show();
 return;
 }
 if(TextUtils.isEmpty(password))
 {
 Toast.makeText(Register.this, "Enter password",
Toast.LENGTH_SHORT).show();
 return;
 }
 mAuth.createUserWithEmailAndPassword(email, password)
 .addOnCompleteListener( new
OnCompleteListener<AuthResult>() {
 @Override
public void onComplete(@NonNull
Task<AuthResult> task) {
 progressBar.setVisibility(View.GONE);
 if (task.isSuccessful()) {
 Toast.makeText(Register.this, "Account
created", Toast.LENGTH_SHORT).show();
 Intent intent = new
Intent(getApplicationContext(),Login.class);
 startActivity(intent);
finish();
 } else {
 // If sign in fails, display a message
to the user.
Toast.makeText(Register.this,
"Authentication failed.",
 Toast.LENGTH_SHORT).show();
 }
 }
 });
 }
 });
 }
}
```

7.2 Login Component:

7.2.1 Design :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
 android:orientation="vertical"
 android:gravity="center"
 android:padding="15dp"
 tools:context=".Login">
 <ImageView
 android:layout_width="100dp"
```

```
android:layout_height="100dp"
android:id="@+id/ImgView"
android:elevation="20dp"
android:layout_gravity="center"
/>
<TextView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:textSize="30sp"
android:textStyle="bold"
android:gravity="center"
android:layout_marginTop="10dp"
android:layout_marginBottom="20dp"
android:text="@string/Hi_Please_Login" />
<com.google.android.material.textfield.TextInputLayout
android:layout_width="match_parent"
android:layout_height="wrap_content">
<com.google.android.material.textfield.TextInputEditText
android:id="@+id/email"
android:hint="@string/email"
android:layout_width="match_parent"
android:layout_height="wrap_content" />
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout
android:layout_width="match_parent"
android:layout_height="wrap_content">
<com.google.android.material.textfield.TextInputEditText
android:id="@+id/password"
android:hint="@string/password"
android:inputType="textPassword"
```

```
android:layout_width="match_parent"
android:layout_height="wrap_content"
/>
</com.google.android.material.textfield.TextInputLayout>
<CheckBox
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="@string/show_password"
android:id="@+id/showpassword"/>
<ProgressBar
android:id="@+id/progressBar"
android:visibility="gone"
android:layout_width="wrap_content"
android:layout_height="wrap_content"/>
<Button
android:id="@+id/btn_login"
android:text="@string/login"
android:layout_width="wrap_content"
android:layout_height="wrap_content"/>
<TextView
android:id="@+id/registerNow"
android:text="@string/click_to_register"
android:textStyle="bold"
android:gravity="center"
android:layout_marginTop="20dp"
android:textSize="20sp"
android:layout_width="match_parent"
android:layout_height="wrap_content"/>
<TextView
android:id="@+id/forgotPass"
```

```
 android:text="@string/forget_password"
 android:textStyle="bold"
 android:gravity="center"
 android:layout_marginTop="10dp"
 android:textSize="20sp"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"/>
</LinearLayout>
```

7.2.2 Java code

```
package com.project.myapplication;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.text.method.HideReturnsTransformationMethod;
import android.text.method.PasswordTransformationMethod;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.ImageView;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.android.material.textfield.TextInputEditText;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
public class Login extends AppCompatActivity {
 TextInputEditText editTextEmail,editTextPassword;
 Button buttonLogin;
 ProgressBar progressBar;
 FirebaseAuth mAuth;
 TextView textView , textView1;
 CheckBox checkBox;
 ImageView imageView;
 @Override
 public void onStart() {
 super.onStart();
 // Check if user is signed in (non-null) and update UI accordingly.
 FirebaseUser currentUser = mAuth.getCurrentUser();
 if(currentUser != null){
 Intent intent = new
Intent(getApplicationContext(),HomePage.class);
 startActivity(intent);
 finish();
 }
 }
 @Override
 protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_login);
 mAuth = FirebaseAuth.getInstance();
 editTextEmail = findViewById(R.id.email);
 editTextPassword = findViewById(R.id.password);
 buttonLogin = findViewById(R.id.btn_login);
 progressBar = findViewById(R.id.progressBar);
 textView = findViewById(R.id.registerNow);
 textView1 = findViewById(R.id.forgotPass);
 checkBox = findViewById(R.id.showpassword);
 imageView = findViewById(R.id.ImgView);
 imageView.setImageResource(R.drawable.base);
 textView.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 Intent intent = new
Intent(getApplicationContext(),Register.class);
 startActivity(intent);
 finish();
 }
 });
 buttonLogin.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 imageView.setImageResource(R.drawable.email);
 progressBar.setVisibility(view.VISIBLE);
 String email,password;
 email = String.valueOf(editTextEmail.getText()) ;
 password = String.valueOf(editTextPassword.getText());
 if(TextUtils.isEmpty(email))
 {
 Toast.makeText(Login.this, "Enter email",
Toast.LENGTH_SHORT).show();
 return;
 }
 if(TextUtils.isEmpty(password))
 {
 Toast.makeText(Login.this, "Enter password",
Toast.LENGTH_SHORT).show();
 return;
 }
 mAuth.signInWithEmailAndPassword(email, password)
 .addOnCompleteListener( new
OnCompleteListener<AuthResult>() {
 @Override
public void onComplete(@NonNull
Task<AuthResult> task) {
 progressBar.setVisibility(View.GONE);
 if (task.isSuccessful()) {
 Toast.makeText(Login.this, "Login
Successful", Toast.LENGTH_SHORT).show();
 Intent intent = new
Intent(getApplicationContext(),HomePage.class);
 startActivity(intent);
 finish();
 } else {
 Toast.makeText(Login.this,
"Authentication failed.",
 Toast.LENGTH_SHORT).show();
 }
 }
 });
 }
 });
 textView1.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 String email = editTextEmail.getText().toString();
 if(email.equals(""))
 {
 Toast.makeText(Login.this, " Please Enter Email",
Toast.LENGTH_SHORT).show();
 }
 else {
 mAuth.sendPasswordResetEmail(email)
 .addOnSuccessListener(new
OnSuccessListener<Void>() {
 @Override
public void onSuccess(Void unused) {
 Toast.makeText(Login.this, "Email
Sent", Toast.LENGTH_SHORT).show();
 }
 })
.addOnFailureListener(new OnFailureListener() {
 @Override
 public void onFailure(@NonNull Exception e)
{
 Toast.makeText(Login.this,
e.getMessage(), Toast.LENGTH_SHORT).show();
 }
 });
 }
 }
 });
 checkBox.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
 @Override
 public void onCheckedChanged(CompoundButton compoundButton,
boolean b) {
 if(b){

editTextPassword.setTransformationMethod(HideReturnsTransformationMethod.ge
tInstance());
 imageView.setImageResource(R.drawable.see);
 }
 else{

editTextPassword.setTransformationMethod(PasswordTransformationMethod.getIn
stance());
 imageView.setImageResource(R.drawable.hide);
 }
 }
 });
 }
 }
```

7.3 Feedback Component

7.3.1 Design :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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_marginTop="120dp"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:orientation="vertical"
 android:background="@drawable/feedback"
 android:elevation="20dp"
 tools:context=".feedback">
 <EditText
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:layout_marginTop="20dp"
 android:hint="@string/name"
 android:textSize="20dp"
 android:textColor="@color/black"
 android:id="@+id/edit1"/>
 <EditText
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:layout_marginTop="30dp"
 android:hint="Feedback"
 android:textSize="20dp"
 android:textColor="@android:color/black"
 android:id="@+id/edit2"
 />
 <Button
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:text="@string/send"
 android:layout_marginTop="120dp"
 android:id="@+id/button"
 android:textSize="20dp"
 />
</LinearLayout>
```

7.3.2 Java Code

```
 package com.project.myapplication;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
public class feedback extends AppCompatActivity {
 @Override
 protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_feedback);
 EditText edit1=(EditText) findViewById(R.id.edit1);
 EditText edit2=(EditText) findViewById(R.id.edit2);
 Button btn=(Button) findViewById(R.id.button);
 btn.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View v) {
 Intent i=new Intent(Intent.ACTION_SEND);
 i.setType("message/html");
 i.putExtra(Intent.EXTRA_EMAIL,new
String[]{"bharat.dhiman112@gmail.com"});
 i.putExtra(Intent.EXTRA_SUBJECT,"Feedback From App");
 i.putExtra(Intent.EXTRA_TEXT, "Name: "+edit1.getText()+"\n
Message: "+ edit2.getText());
 try {
 startActivity(Intent.createChooser(i, "Please select
Email"));
 }
 catch(android.content.ActivityNotFoundException ex)
 {
```

```
 Toast.makeText(feedback.this, "There are no Email
Clients", Toast.LENGTH_SHORT).show();
 }
 }
 });
 }
}
```

7.4 Home Page

7.4.1 Design :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
 android:orientation="vertical"
 tools:context=".HomePage">
 <TextView
 android:id="@+id/user_details"
 android:layout_width="match_parent"
 android:layout_height="20dp"
 android:layout_marginTop="10dp"
 android:gravity="start"
 android:text="Hello World!"
 android:textAlignment="center"
 android:textSize="14sp"
 android:textStyle="bold" />
 <androidx.cardview.widget.CardView
 android:layout_width="match_parent"
 android:layout_height="200dp"
```

```
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:layout_marginEnd="20dp"
app:cardCornerRadius="20dp"
app:cardElevation="20dp">
<com.denzcoskun.imageslider.ImageSlider
android:id="@+id/imageslider"
android:layout_width="match_parent"
android:layout_height="200dp"
app:iss_auto_cycle="true"
app:iss_corner_radius="20"
app:iss_delay="1000"
app:iss_period="1000"
app:iss_title_background="@drawable/gradient" />
</androidx.cardview.widget.CardView>
<androidx.cardview.widget.CardView
android:id="@+id/card"
android:layout_width="match_parent"
android:layout_height="50dp"
android:layout_marginStart="20dp"
android:layout_marginTop="20dp"
android:layout_marginEnd="20dp"
app:cardBackgroundColor="#A368E6"
app:cardCornerRadius="100dp"
app:cardElevation="20dp">
<TextView
android:id="@+id/user_city"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="15dp"
```

```
android:gravity="center"
android:text="@string/click_to_search_city"
android:textColor="@color/white"
android:textStyle="bold" />
</androidx.cardview.widget.CardView>
<androidx.cardview.widget.CardView
android:layout_width="match_parent"
android:layout_height="240dp"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:layout_marginEnd="20dp"
app:cardCornerRadius="20dp"
app:cardElevation="20dp">
<LinearLayout
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/weatherback"
android:orientation="vertical">
<EditText
android:id="@+id/et"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="13dp"
android:hint="@string/enter_your_city_name" />
<Button
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="center_horizontal"
android:layout_marginTop="20dp"
android:onClick="getweather"
```

```
android:text="@string/get_weather"
tools:ignore="OnClick" />
<TextView
android:id="@+id/tv"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="20dp"
android:gravity="center_horizontal"
android:textColor="@color/black"
android:textSize="20sp"
android:textStyle="bold" />
</LinearLayout>
</androidx.cardview.widget.CardView>
<LinearLayout
android:layout_width="match_parent"
android:layout_height="60dp"
android:layout_gravity="center_horizontal"
android:layout_marginTop="40dp"
android:orientation="horizontal">
<ImageButton
android:id="@+id/btn_navigation"
android:layout_width="40dp"
android:layout_height="40dp"
android:layout_gravity="center"
android:layout_marginLeft="10dp"
android:background="@drawable/baseline_navigation_24"
android:text="@string/navigation" />
<ImageButton
android:id="@+id/btn_map"
android:layout_width="40dp"
```

```
android:layout_height="40dp"
android:layout_gravity="center"
android:layout_marginLeft="20dp"
android:background="@drawable/baseline_map_24"
android:text="@string/logout"
android:weightSum="100" />
<ImageButton
android:id="@+id/btn_exchange"
android:layout_width="40dp"
android:layout_height="40dp"
android:layout_gravity="center"
android:layout_marginLeft="20dp"
android:background="@drawable/baseline_currency_exchange_24"
android:text="@string/logout"
android:weightSum="100" />
<ImageButton
android:id="@+id/btn_about"
android:layout_width="40dp"
android:layout_height="40dp"
android:layout_gravity="center"
android:layout_marginStart="20dp"
android:background="@drawable/baseline_info_24"
android:text="@string/logout"
android:weightSum="100" />
<ImageButton
android:id="@+id/btn_feedback"
android:layout_width="40dp"
android:layout_height="40dp"
android:layout_gravity="center"
android:layout_marginLeft="20dp"
```

```
 android:background="@drawable/baseline_feedback_24"
 android:text="@string/logout"
 android:weightSum="100" />
 <ImageButton
 android:id="@+id/btn_logout"
 android:layout_width="40dp"
 android:layout_height="40dp"
 android:layout_gravity="center"
 android:layout_marginLeft="20dp"
 android:background="@drawable/baseline_logout_24"
 android:hint="@string/logout"
 android:weightSum="100" />
 </LinearLayout>
</LinearLayout>
```

7.4.2 Java Code :

```
package com.project.myapplication;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import androidx.cardview.widget.CardView;
import com.denzcoskun.imageslider.ImageSlider;
import com.denzcoskun.imageslider.constants.ScaleTypes;
import com.denzcoskun.imageslider.models.SlideModel;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import java.util.ArrayList;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;
import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;
public class HomePage extends AppCompatActivity {
 FirebaseAuth auth;
 ImageButton button, button1,button2,button3,button4,button5;
 TextView textView, textView1 ,textView3;
 EditText editText;
 ImageSlider imageSlider;
 CardView cardView;
 FirebaseUser user;
 String url = "https://api.openweathermap.org/data/2.5/weather?q={city
name}&appid={API key}";
 String apikey = "93609261145cb55f8e16c1fb75a76a56";
 @Override
 protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_home_page);
 auth = FirebaseAuth.getInstance();
 user = auth.getCurrentUser();
 textView = findViewById(R.id.user_details);
 button = findViewById(R.id.btn_logout);
 textView1 = findViewById(R.id.user_city);
 textView3 = findViewById(R.id.tv);
 editText = findViewById(R.id.et);
 cardView = findViewById(R.id.card);
 imageSlider = findViewById(R.id.imageslider);
 button1 = findViewById(R.id.btn_navigation);
 button2 = findViewById(R.id.btn_about);
 button3 = findViewById(R.id.btn_feedback);
 button4 = findViewById(R.id.btn_map);
 button5 = findViewById(R.id.btn_exchange);
 button1.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 Toast.makeText(HomePage.this, "Navigation selected",
Toast.LENGTH_SHORT).show();
 Intent intent = new Intent(getApplicationContext(),
Map.class);
 startActivity(intent);
 }
 });
 button2.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 Toast.makeText(HomePage.this, "About selected",
Toast.LENGTH_SHORT).show();
 Intent intent = new Intent(getApplicationContext(),
About.class);
 startActivity(intent);
 }
 });
 button3.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 Toast.makeText(HomePage.this, "Feedback selected",
Toast.LENGTH_SHORT).show();
 Intent intent = new Intent(getApplicationContext(),
feedback.class);
 startActivity(intent);
 }
 });
 button7.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 Toast.makeText(HomePage.this, "Map selected",
Toast.LENGTH_SHORT).show();
 Intent intent = new
Intent(getApplicationContext(),Gmap.class);
 startActivity(intent);
 }
 });
 button5.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 Toast.makeText(HomePage.this, "Currency Exchange selected",
Toast.LENGTH_SHORT).show();
 Intent intent = new
Intent(getApplicationContext(),exchange.class);
 startActivity(intent);
 }
 });
 ArrayList<SlideModel> slideModels = new ArrayList<>();
 slideModels.add(new SlideModel(R.drawable.r1, ScaleTypes.FIT));
 slideModels.add(new SlideModel(R.drawable.r3, ScaleTypes.FIT));
 slideModels.add(new SlideModel(R.drawable.r4, ScaleTypes.FIT));
 slideModels.add(new SlideModel(R.drawable.arr4, ScaleTypes.FIT));
 slideModels.add(new SlideModel(R.drawable.aar5, ScaleTypes.FIT));
 imageSlider.setImageList(slideModels, ScaleTypes.FIT);
 if (user == null) {
 Intent intent = new Intent(getApplicationContext(),
Login.class);
 startActivity(intent);
 finish();
 } else {
 textView.setText("Welcome! " + user.getEmail());
 button.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 Toast.makeText(HomePage.this, "Logout Selected",
Toast.LENGTH_SHORT).show();
 FirebaseAuth.getInstance().signOut();
 Intent intent = new Intent(getApplicationContext(),
Login.class);
 startActivity(intent);
finish();
 }
 });
 }
 textView1.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 Intent intent = new Intent(getApplicationContext(),
MainActivity.class);
 startActivity(intent);
 }
 });
 cardView.setOnClickListener(new View.OnClickListener() {
```

```
 @Override
 public void onClick(View view) {
 Intent intent = new Intent(getApplicationContext(),
MainActivity.class);
 startActivity(intent);
 }
 });
 }
 public void getweather(View v)
 {
 Retrofit retrofit=new Retrofit.Builder()
 .baseUrl("https://api.openweathermap.org/data/2.5/")
 .addConverterFactory(GsonConverterFactory.create())
 .build();
 weatherapi myapi = retrofit.create(weatherapi.class);
 Call<Example> exampleCall =
myapi.getweather(editText.getText().toString().trim(),apikey);
 exampleCall.enqueue(new Callback<Example>() {
 @Override
 public void onResponse(Call<Example> call, Response<Example>
response) {
 if(response.code()==404){
 Toast.makeText(HomePage.this,"Please Enter a valid
City",Toast.LENGTH_LONG).show();
 }
 else if(!(response.isSuccessful())){
 Toast.makeText(HomePage.this,response.code()+"
",Toast.LENGTH_LONG).show();
 return;
 }
 Example mydata=response.body();
 Main main=mydata.getMain();
 Double temp=main.getTemp();
 Integer humidity = main.getHumidity();
 Integer pressure = main.getPressure();
 Integer temperature=(int)(temp-273.15);
 textView3.setText("Temperature is:: " +
String.valueOf(temperature)+" °C\n" + "Humidity is:: " + humidity +"%"+ "\n
Pressure is:: " +pressure + " hPa\n");
 }
 @Override
 public void onFailure(Call<Example> call, Throwable t) {
 Toast.makeText(HomePage.this, t.getMessage(),
Toast.LENGTH_SHORT).show();
 }
 });
 }
}
```

7.5 Navigation

7.5.1 Design :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
 android:layout_width="match_parent"
 android:layout_height="match_parent"
 xmlns:tools="http://schemas.android.com/tools"
 xmlns:app="http://schemas.android.com/apk/res-auto"
 android:orientation="vertical"
 android:gravity="center"
 android:padding="20dp"
 android:background="@drawable/navback"
```

```
tools:context=".Map">
<androidx.cardview.widget.CardView
android:layout_width="match_parent"
android:layout_height="200dp"
app:cardCornerRadius="20dp"
app:cardElevation="20dp">
<LinearLayout
android:layout_width="match_parent"
android:layout_height="200dp"
android:orientation="vertical"
android:background="@drawable/iphoto">
<EditText
android:id="@+id/source"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="10dp"
android:layout_marginStart="15dp"
android:layout_marginEnd="15dp"
android:hint="@string/enter_source_location"/>
<EditText
android:id="@+id/destination"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="10dp"
android:layout_marginStart="15dp"
android:layout_marginEnd="15dp"
android:hint="@string/enter_destination_location"/>
<Button
android:layout_width="wrap_content"
android:layout_height="wrap_content"
```

```
 android:layout_marginTop="10dp"
 android:layout_gravity="center"
 android:text="@string/open_navigation"
 android:id="@+id/btnsubmit"/>
 </LinearLayout>
 </androidx.cardview.widget.CardView>
</LinearLayout>
```

7.5.2 Java Code :

```
package com.project.myapplication;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
public class Map extends AppCompatActivity {
 @Override
 protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.map_navigation);
 EditText editTextSource = findViewById(R.id.source);
 EditText editTextDestination = findViewById(R.id.destination);
 Button button = findViewById(R.id.btnsubmit);
 button.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 String source = editTextSource.getText().toString();
 String destination =
editTextDestination.getText().toString();
```

```
 if (source.equals("") && destination.equals("")){
 Toast.makeText(Map.this, "Enter both source and
destination", Toast.LENGTH_SHORT).show();
 }
 else{
 Uri uri = Uri.parse("https://www.google.com/maps/dir/"
+ source + "/" + destination);
 Intent intent = new Intent(Intent.ACTION_VIEW,uri);
 intent.setPackage("com.google.android.apps.maps");
 intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
 startActivity(intent);
 }
 }
 });
 }
}
```

7.6 Map

7.6.1 Design :

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
 xmlns:android="http://schemas.android.com/apk/res/android"
 xmlns:app="http://schemas.android.com/apk/res-auto"
 android:layout_width="match_parent"
 android:layout_height="match_parent">
 <!--fragment to display our maps-->
 <fragment xmlns:tools="http://schemas.android.com/tools"
 android:id="@+id/map"
 android:name="com.google.android.gms.maps.SupportMapFragment"
 android:layout_width="match_parent"
 android:layout_height="match_parent"
 tools:context=".Gmap" />
 <!--search view to search our location-->
 <androidx.appcompat.widget.SearchView
 android:id="@+id/idSearchView"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:layout_margin="10dp"
 android:background="@drawable/bg_round"
 android:elevation="5dp"
 app:iconifiedByDefault="false"
 app:queryHint="Search Here" />
</RelativeLayout>
```

7.6.2 Java Code :

```
package com.project.myapplication;
import android.location.Address;
import android.location.Geocoder;
import android.os.Bundle;
import androidx.appcompat.widget.SearchView;
import androidx.fragment.app.FragmentActivity;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import java.io.IOException;
import java.util.List;
public class Gmap extends FragmentActivity implements OnMapReadyCallback {
 private GoogleMap mMap;
 // creating a variable
 // for search view.
 SearchView searchView;
 @Override
 protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_gmap);
 // initializing our search view.
 searchView = findViewById(R.id.idSearchView);
 // Obtain the SupportMapFragment and get notified
 // when the map is ready to be used.
 SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.map);
 Bundle extras = getIntent().getExtras();
 //Accessing the values form intent using key
 if(extras != null){
 String cityname = extras.getString("cityname");
 searchView.setQuery(cityname, true);
 }
 // adding on query listener for our search view.
 searchView.setOnQueryTextListener(new
SearchView.OnQueryTextListener() {
 @Override
 public boolean onQueryTextSubmit(String query) {
 // on below line we are getting the
// location name from search view.
String location = searchView.getQuery().toString();
 // below line is to create a list of address
// where we will store the list of all address.
List<Address> addressList = null;
 // checking if the entered location is null or not.
 if (location != null || location.equals("")) {
```

```
 // on below line we are creating and initializing a
geo coder.
 Geocoder geocoder = new Geocoder(Gmap.this);
 try {
 // on below line we are getting location from
the
 // location name and adding that location to
address list.
 addressList =
geocoder.getFromLocationName(location, 1);
 } catch (IOException e) {
 e.printStackTrace();
 }
 // on below line we are getting the location
// from our list a first position.
Address address = addressList.get(0);
 // on below line we are creating a variable for our
location
 // where we will add our locations latitude and
longitude.
 LatLng latLng = new LatLng(address.getLatitude(),
address.getLongitude());
 // on below line we are adding marker to that
position.
 mMap.addMarker(new
MarkerOptions().position(latLng).title(location));
 // below line is to animate camera to that
position.

mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng, 20));
```

```
 }
return false;
 }
 @Override
 public boolean onQueryTextChange(String newText) {
 return false;
 }
 });
 // at last we calling our map fragment to update.
 mapFragment.getMapAsync(this);
 }
 @Override
 public void onMapReady(GoogleMap googleMap) {
 mMap = googleMap;
 }
}
```

7.7 About

7.7.1 Design :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
 android:orientation="vertical"
 android:background="@drawable/pexels"
 tools:context=".About">
 <androidx.cardview.widget.CardView
 android:layout_width="match_parent"
 android:layout_height="50dp"
```

```
android:layout_marginTop="20dp"
app:cardCornerRadius="20dp">
<ImageView
android:layout_width="50dp"
android:layout_height="50dp"
android:id="@+id/about1"
android:layout_marginStart="20sp"
android:background="@drawable/nav_about"/>
<TextView
android:layout_width="wrap_content"
android:layout_height="30dp"
android:text="Application Name"
android:id="@+id/about2"
android:layout_marginStart="100sp"
android:textSize="20sp"
android:textColor="@color/black"
/>
<TextView
android:id="@+id/about3"
android:layout_width="wrap_content"
android:layout_height="20dp"
android:text="@string/app_name"
android:layout_marginStart="100sp"
android:layout_marginTop="30dp"
android:textColor="@color/lightblack"
/>
</androidx.cardview.widget.CardView>
<androidx.cardview.widget.CardView
android:layout_width="match_parent"
android:layout_height="50dp"
```

```
android:layout_marginTop="20dp"
app:cardCornerRadius="20dp">
<ImageView
android:layout_width="50dp"
android:layout_height="50dp"
android:layout_marginStart="20sp"
android:background="@drawable/baseline_email_24"/>
<TextView
android:layout_width="wrap_content"
android:layout_height="30dp"
android:text="@string/email"
android:layout_marginStart="100sp"
android:textSize="20sp"
android:textColor="@color/black"
/>
<TextView
android:layout_width="wrap_content"
android:layout_height="20dp"
android:text="bharat.dhiman11@gmail.com"
android:layout_marginStart="100sp"
android:layout_marginTop="30dp"
android:textColor="@color/lightblack"
/>
</androidx.cardview.widget.CardView>
<androidx.cardview.widget.CardView
android:layout_width="match_parent"
android:layout_height="50dp"
android:layout_marginTop="20dp"
app:cardCornerRadius="20dp">
<ImageView
```

```
android:id="@+id/rate1"
android:layout_width="50dp"
android:layout_height="50dp"
android:layout_marginStart="20sp"
android:background="@drawable/baseline_star_rate_24"/>
<TextView
android:id="@+id/rate2"
android:layout_width="wrap_content"
android:layout_height="30dp"
android:text="Rate us"
android:layout_marginStart="100sp"
android:textSize="20sp"
android:textColor="@color/black"
/>
<TextView
android:id="@+id/rate3"
android:layout_width="wrap_content"
android:layout_height="20dp"
android:text="Give your Feedback"
android:layout_marginStart="100sp"
android:layout_marginTop="30dp"
android:textColor="@color/lightblack"
/>
</androidx.cardview.widget.CardView>
<androidx.cardview.widget.CardView
android:layout_width="match_parent"
android:layout_height="50dp"
android:layout_marginTop="20dp"
app:cardCornerRadius="20dp">
<ImageView
```

```
android:id="@+id/share1"
android:layout_width="50dp"
android:layout_height="50dp"
android:layout_marginStart="20sp"
android:background="@drawable/baseline_share_24"/>
<TextView
android:id="@+id/share2"
android:layout_width="wrap_content"
android:layout_height="30dp"
android:text="@string/share"
android:layout_marginStart="100sp"
android:textSize="20sp"
android:textColor="@color/black"
/>
<TextView
android:id="@+id/share3"
android:layout_width="wrap_content"
android:layout_height="20dp"
android:text="@string/share_app_with_others"
android:layout_marginStart="100sp"
android:layout_marginTop="30dp"
android:textColor="@color/lightblack"
/>
</androidx.cardview.widget.CardView>
<TextView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="@string/powered_by_n_bat_software_developer"
android:textColor="@color/white"
android:layout_marginTop="250dp"
```

```
 android:textAlignment="center"
 android:textStyle="bold"
 android:textSize="20sp"/>
</LinearLayout>
```

7.7.2 Java Code :

```
package com.project.myapplication;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
public class About extends AppCompatActivity {
ImageView imageView,imageView1,imageView2;
TextView textView,textView1 , textView3,textView4,textView5,textView6;
 @Override
 protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_about);
 imageView = findViewById(R.id.about1);
 imageView1 = findViewById(R.id.share1);
 imageView2 = findViewById(R.id.rate1);
 textView = findViewById(R.id.about2);
 textView1 = findViewById(R.id.about3);
 textView3 = findViewById(R.id.share3);
 textView4 = findViewById(R.id.share2);
 textView5 = findViewById(R.id.rate2);
 textView6 = findViewById(R.id.rate3);
 imageView.setOnClickListener(new View.OnClickListener() {
```

```
 @Override
 public void onClick(View view) {
 Intent intent = new
Intent(getApplicationContext(),Info.class);
 startActivity(intent);
 }
 });
 textView1.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 Intent intent = new
Intent(getApplicationContext(),Info.class);
 startActivity(intent);
 }
 });
 textView.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 Intent intent = new
Intent(getApplicationContext(),Info.class);
 startActivity(intent);
 }
 });
 textView3.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 try {
 Intent intent = new Intent(Intent.ACTION_SEND);
 intent.setType("text/plain");
 intent.putExtra(Intent.EXTRA_SUBJECT,"My CITY
INFORMATION");

intent.putExtra(Intent.EXTRA_TEXT,"https://play.google.com/store/apps/detai
ls?id="+getApplicationContext().getPackageName());
 startActivity(Intent.createChooser(intent,"Share
With"));
 }catch (Exception e){
 Toast.makeText(About.this, "Unable to share this app.",
Toast.LENGTH_SHORT).show();
 }
 }
 });
 textView7.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 try {
 Intent intent = new Intent(Intent.ACTION_SEND);
 intent.setType("text/plain");
 intent.putExtra(Intent.EXTRA_SUBJECT,"My CITY
INFORMATION");

intent.putExtra(Intent.EXTRA_TEXT,"https://play.google.com/store/apps/detai
ls?id="+getApplicationContext().getPackageName());
 startActivity(Intent.createChooser(intent,"Share
With"));
 }catch (Exception e){
 Toast.makeText(About.this, "Unable to share this app.",
Toast.LENGTH_SHORT).show();
 }
 }
 });
 imageView1.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 try {
 Intent intent = new Intent(Intent.ACTION_SEND);
 intent.setType("text/plain");
 intent.putExtra(Intent.EXTRA_SUBJECT,"My CITY
INFORMATION");

intent.putExtra(Intent.EXTRA_TEXT,"https://play.google.com/store/apps/detai
ls?id="+getApplicationContext().getPackageName());
 startActivity(Intent.createChooser(intent,"Share
With"));
 }catch (Exception e){
 Toast.makeText(About.this, "Unable to share this app.",
Toast.LENGTH_SHORT).show();
 }
 }
 });
 textView5.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 Uri uri =
Uri.parse("https://play.google.com/store/apps/details?id="+getApplicationCo
ntext().getPackageName());
 Intent i = new Intent(Intent.ACTION_VIEW,uri);
 try {
 startActivity(i);
 }catch (Exception e)
 {
 Toast.makeText(About.this, e.getMessage(),
Toast.LENGTH_SHORT).show();
 }
 }
 });
 textView6.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 Uri uri =
Uri.parse("https://play.google.com/store/apps/details?id="+getApplicationCo
ntext().getPackageName());
 Intent i = new Intent(Intent.ACTION_VIEW,uri);
 try {
 startActivity(i);
 }catch (Exception e)
 {
 Toast.makeText(About.this, e.getMessage(),
Toast.LENGTH_SHORT).show();
 }
 }
 });
 imageView2.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 Uri uri =
Uri.parse("https://play.google.com/store/apps/details?id="+getApplicationCo
ntext().getPackageName());
 Intent i = new Intent(Intent.ACTION_VIEW,uri);
 try {
```

```
 startActivity(i);
 }catch (Exception e)
 {
 Toast.makeText(About.this, e.getMessage(),
Toast.LENGTH_SHORT).show();
 }
 }
 });
 }
}
```

7.8 Currency Exchange

7.8.1 Design :

```
<?xml version="1.0" encoding="UTF-8"?>
 <RelativeLayout
 xmlns:tools="http://schemas.android.com/tools"
 xmlns:app="http://schemas.android.com/apk/res-auto"
 xmlns:android="http://schemas.android.com/apk/res/android"
 tools:context=".exchange"
 android:orientation="vertical"
 android:focusableInTouchMode="true"
 android:focusable="true"
 android:background="@drawable/currencyback"
 android:layout_height="match_parent"
 android:layout_width="match_parent">
 <androidx.cardview.widget.CardView
 android:focusable="true"
 android:layout_height="300dp"
 android:layout_width="match_parent"
 app:cardCornerRadius="25dp"
 android:clickable="true"
```

```
android:layout_marginEnd="10dp"
android:layout_marginTop="20dp"
android:layout_marginStart="10dp"
android:id="@+id/card_view">
-<LinearLayout
android:orientation="vertical"
android:layout_height="wrap_content"
android:layout_width="match_parent"
android:layout_marginTop="8dp"
android:id="@+id/masterLinearLayout"
android:paddingBottom="10dp"
android:paddingTop="10dp">
<LinearLayout
android:orientation="vertical"
android:layout_height="wrap_content"
android:layout_width="match_parent"
android:layout_marginTop="8dp"
android:id="@+id/dropdown_menu_layout"
android:paddingBottom="10dp"
android:paddingTop="10dp">
<LinearLayout
android:layout_height="wrap_content"
android:layout_width="match_parent"
android:id="@+id/layoutOne"
android:gravity="top|bottom|center_vertical|center|fill">
<TextView
android:layout_height="wrap_content"
android:layout_width="0dp"
android:layout_marginStart="10dp"
android:id="@+id/convertFromTitle"
```

```
 android:textColor="#000000"
 android:text="Convert From"
 android:layout_weight="1.0"/>
 <TextView
 android:layout_height="wrap_content"
 android:layout_width="0dp"
 android:id="@+id/convertToTitle"
 android:textColor="#000000"
 android:text="Convert To"
 android:layout_weight="1.0"/>
</LinearLayout>
 <LinearLayout
 android:layout_height="match_parent"
 android:layout_width="match_parent"
 android:id="@+id/layoutTwo">
 <TextView
 android:background="@android:drawable/editbox_background"
 android:layout_height="wrap_content"
 android:layout_width="0dp"
 android:layout_marginStart="10dp"
 android:id="@+id/convert_from_dropdown_menu"
 android:textColor="#000000"
 android:text=""
 android:layout_weight="1.0"
 android:drawableRight="@drawable/baseline_arrow_drop_down_24"
 android:drawableEnd="@drawable/baseline_arrow_drop_down_24"
 android:hint="Select Country"/>
 <TextView
 android:background="@android:drawable/editbox_background"
 android:layout_height="wrap_content"
 android:layout_width="0dp"
 android:id="@+id/convert_to_dropdown_menu"
 android:textColor="#000000"
 android:text=""
 android:layout_weight="1.0"
 android:drawableRight="@drawable/baseline_arrow_drop_down_24"
 android:drawableEnd="@drawable/baseline_arrow_drop_down_24"
 android:hint="Select Country"/>
</LinearLayout>
</LinearLayout>
 -<LinearLayout
 android:orientation="vertical"
 android:layout_height="wrap_content"
 android:layout_width="match_parent"
 android:id="@+id/linearLayout3"
 android:paddingBottom="10dp"
 android:paddingTop="10dp"
 android:layout_margin="8dp">
 -<LinearLayout
 android:orientation="horizontal"
 android:layout_height="wrap_content"
 android:layout_width="match_parent"
 android:id="@+id/linarLayout01"
 android:gravity="top|bottom|center_vertical|fill_vertical|center|fill">
 <TextView
 android:layout_height="wrap_content"
 android:layout_width="0dp"
 android:layout_marginStart="10dp"
 android:id="@+id/amountToConvertValueText"
 android:textColor="#000000"
 android:text="Amount To Convert"
 android:layout_weight="1.0"/>
 <TextView
 android:layout_height="wrap_content"
 android:layout_width="0dp"
 android:id="@+id/ConversionResultText"
 android:gravity="center"
 android:textColor="#000000"
 android:text="conversion Rate"
 android:layout_weight="1.0"/>
</LinearLayout>
 <LinearLayout
 android:orientation="horizontal"
 android:layout_height="wrap_content"
 android:layout_width="match_parent"
 android:id="@+id/linearLayout4">
 <EditText
 android:layout_height="wrap_content"
 android:layout_width="0dp"
 android:layout_marginStart="10dp"
 android:id="@+id/amountToConvertValueEditText"
 android:textColor="#000000"
 android:text=""
 android:layout_weight="1.0"
 android:textSize="14sp"
 android:singleLine="true"
 android:maxLines="1"
 android:inputType="number"/>
 <TextView
 android:layout_height="wrap_content"
 android:layout_width="0dp"
 android:id="@+id/conversionRateText"
 android:gravity="center"
 android:textColor="#000000"
 android:layout_weight="1.0"
 android:textSize="18sp"/>
</LinearLayout>
</LinearLayout>
 <RelativeLayout
 android:layout_height="wrap_content"
 android:layout_width="match_parent"
 android:padding="20sp"
 android:layout_marginLeft="10dp">
 <Button
 android:layout_height="wrap_content"
 android:layout_width="match_parent"
 android:id="@+id/conversionsButton"
 android:paddingBottom="7dp"
 android:paddingTop="7dp"
 android:gravity="center"
 android:textColor="#000000"
 android:text="Convert"/>
</RelativeLayout>
</LinearLayout>
</androidx.cardview.widget.CardView>
</RelativeLayout>
```

7.8.2 Java Code :

```
package com.project.myapplication;
import androidx.appcompat.app.AppCompatActivity;
import android.app.Dialog;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.TextView;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import org.json.JSONException;
import org.json.JSONObject;
import java.math.BigDecimal;
import java.math.RoundingMode;
import java.util.ArrayList;
public class exchange extends AppCompatActivity {
 //Create objects for all the widgets
 TextView convertFromDropdownTextView, convertToDropdownTextView,
conversionRateText;
 EditText amountToConvert;
 ArrayList<String> arrayList;
 Dialog fromDialog;
 Dialog toDialog;
 Button convertButton;
 String convertFromValue, convertToValue, conversionValue;
 String[] country = {"AFN","ALL","ARS","AUD",
"BHD","BYN","BRL","BND","BGN","KHR","CAD","CLP","CNY","COP","CRC","CZK",
"DK
K","DZD","EGP","EUR","HKD","HUF","ISK","INR","IDR","IRR","IQD","ILS","JPY",
"JOD","KZT","KES","KWD","MMK","LAK","LBP","MOP","MYR","MXN","MDL","MA
D","NG
N","NRP","TWD","NZD","NOK","OMR","PKR","PHP","PLN","GBP","QAR","RON","RU
B",
"SAR","SGD","ZAR","LKR","SEK","CHF","SYP","BDT","TZA","THB","TND","TRY","A
E
D","UGX","UAH","USD","UZS","VND","KRW","ZMW"};
 @Override
 protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_exchange);
 convertFromDropdownTextView =
findViewById(R.id.convert_from_dropdown_menu);
 convertToDropdownTextView =
findViewById(R.id.convert_to_dropdown_menu);
 convertButton = findViewById(R.id.conversionsButton);
 amountToConvert = findViewById(R.id.amountToConvertValueEditText);
 conversionRateText = findViewById(R.id.conversionRateText);
 arrayList = new ArrayList<>();
 for (String i : country) {
 arrayList.add(i);
 }
 convertFromDropdownTextView.setOnClickListener(new
View.OnClickListener() {
 @Override
 public void onClick(View v) {
 fromDialog = new Dialog(exchange.this);
 fromDialog.setContentView(R.layout.from_spinner);
 fromDialog.getWindow().setLayout(650, 800);
 fromDialog.show();
 EditText editText =
fromDialog.findViewById(R.id.edit_text);
 ListView listView =
fromDialog.findViewById(R.id.list_view);
 ArrayAdapter<String> adapter = new
ArrayAdapter<>(exchange.this, android.R.layout.simple_list_item_1,
arrayList);
 listView.setAdapter(adapter);
 editText.addTextChangedListener(new TextWatcher() {
 @Override
public void beforeTextChanged(CharSequence s, int
start, int before, int count) {
 }
 @Override
public void onTextChanged(CharSequence s, int start,
int before, int count) {
 adapter.getFilter().filter(s);
 }
 @Override
public void afterTextChanged(Editable editable) {
 }
 });
 listView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
 @Override
 public void onItemClick(AdapterView<?> parent, View
view, int position, long id) {
```

```
convertFromDropdownTextView.setText(adapter.getItem(position));
 fromDialog.dismiss();
convertFromValue = adapter.getItem(position);
 }
 });
 }
 });
 convertToDropdownTextView.setOnClickListener(new
View.OnClickListener() {
 @Override
 public void onClick(View v) {
 toDialog=new Dialog(exchange.this);
 toDialog.setContentView(R.layout.to_spinner);
 toDialog.getWindow().setLayout(650,800);
 toDialog.show();
 EditText editText=toDialog.findViewById(R.id.edit_text);
 ListView listView=toDialog.findViewById(R.id.list_view);
 ArrayAdapter<String> adapter=new
ArrayAdapter<>(exchange.this,
android.R.layout.simple_list_item_1,arrayList);
 listView.setAdapter(adapter);
 editText.addTextChangedListener(new TextWatcher() {
 @Override
public void beforeTextChanged(CharSequence s, int
start, int count, int after) {
 }
 @Override
public void onTextChanged(CharSequence s, int start,
int count, int after) {
 adapter.getFilter().filter(s);
 }
 @Override
 public void afterTextChanged(Editable s) {
 }
 });
 listView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
 @Override
public void onItemClick(AdapterView<?> parent, View
view, int position, long id) {

convertToDropdownTextView.setText(adapter.getItem(position));
 toDialog.dismiss();
convertToValue=adapter.getItem(position);
 }
 });
 }
 });
 convertButton.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View v) {
 try {
 Double
amountToConvert=Double.valueOf(exchange.this.amountToConvert.getText().toSt
ring());

getConversionRate(convertFromValue,convertToValue,amountToConvert);
 }
 catch (Exception e) {
 }
 }
 });
 }
 public String getConversionRate(String convertFrom,String
convertTo,Double amountToConvert) {
 RequestQueue queue = Volley.newRequestQueue(this);
 String url =
"https://free.currconv.com/api/v7/convert?q="+convertFrom+"_"+convertTo+"&c
ompact=ultra&apiKey=10fef9fd6dde11508a0c";
 StringRequest stringRequest = new StringRequest(Request.Method.GET,
url, new Response.Listener<String>() {
 @Override
 public void onResponse(String response) {
 JSONObject jsonObject = null;
 try {
 jsonObject = new JSONObject(response);
Double conversionRateValue = round(((Double)
jsonObject.get(convertFrom+"_"+convertTo)), 2);
 conversionValue = "" + round((conversionRateValue *
amountToConvert), 2);
 conversionRateText.setText(conversionValue);
 } catch (JSONException e) {
 e.printStackTrace();
 }
 }
 }, new Response.ErrorListener() {
 @Override
 public void onErrorResponse(VolleyError error) {
 }
 });
 queue.add(stringRequest);
 return null;
 }
 public static double round(double value,int places)
 {
 if(places<0) throw new IllegalArgumentException();
 BigDecimal bd=BigDecimal.valueOf(value);bd=bd.setScale(places,
RoundingMode.HALF_DOWN);
 return bd.doubleValue();
 }
}
```

CHAPTER-8

TESTING

Testing is the major quality control measure employed during software development. Testing is the process of executing a program with the intent of finding an error. No piece of code is completely ready unless it has been fully tested. This stage is very important, it is verified whether the code developed meet the requirement specifications or not. Moreover all validations are also checked in the testing stage.

8.1 Need for Testing

Testing is vital to the success of the system. Testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. Inadequate testing or non-testing leads to errors that may not appear until a month later. A small system error can conceivably explode into a much larger problem effective testing early in the process translates directly into long term cost saving form a reduced number of errors. Second reason for testing is its utility as a user oriented vehicle before implementation. Finally, testing leads to software reliability. Identifying and removing faults, during testing can make the software more reliable. This increases user‘s confidence in the system.

8.2 Testing Methods

The two popular testing methods are :

- Black Box Testing
- White Box Testing

8.2.1 Black box testing

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

This testing can be done by the user who does not have the complete knowledge about the project. Here are the generic steps followed to carry out any type of Black Box Testing.

- Initially requirements and specifications of the system are examined.

- Tester chooses valid inputs and some invalid inputs to verify that the system is able to detect them.
- Tester determines expected outputs for all those inputs.
- Software tester constructs test cases with the selected inputs.
- The test cases are executed.
- Software tester compares the actual outputs with the expected outputs.
- Defects if any are fixed and re-tested.

Tests Performed	**Pass**	**Fail**
When user tried to login with wrong ID.		✓
When user tries to login without registration.		✓
When user enters wrong email ID.		✓
When user enters mobile number of less than 10 digits.		✓
When user tries to enter number as his/her name.		✓

Table 10:: Black-Box Test Cases

8.2.2 White box testing

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing.

Black box testing is often used for validation and white box testing is often used for verification. To apply this testing, tester should have complete knowledge about the coding in which project is created.

White box testing involves the testing of the software code for the following:

- Internal security holes
- Broken or poorly structured paths in the coding processes
- The flow of specific inputs through the code
- Expected output

- The functionality of conditional loops
- Testing of each statement and function on an individual basis.

Tests performed	**Pass**	**Fail**
On clicking a particular button appropriate page opens.	✓	
Validation is implemented correctly in forms.	✓	
Login is performed using correct username and password.	✓	
Redirecting to other pages automatically.	✓	

Table 11:: White box testing

CHAPTER-9

RESULTS

9.1Register Module:

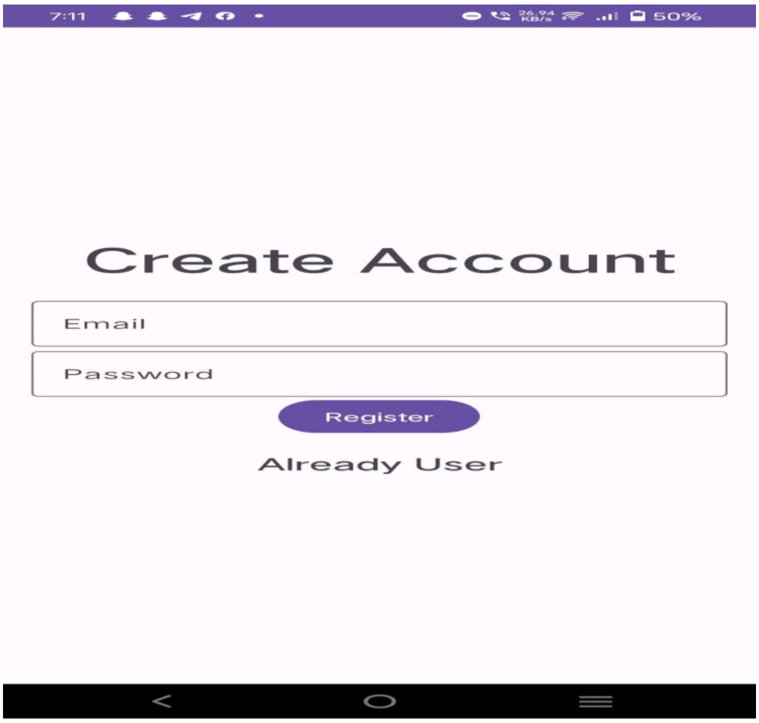


Fig. 8:: Test module

9.2Login page:

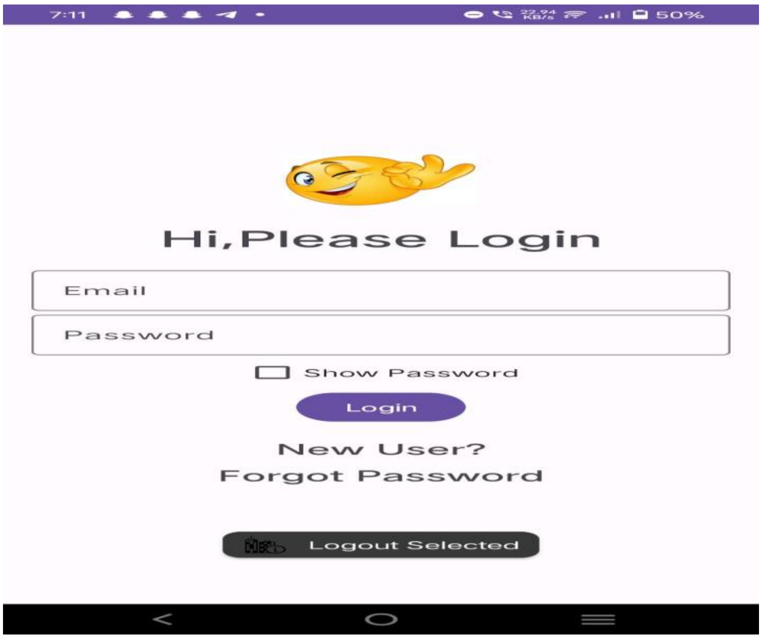


Fig. 9:: Login Page

9.3 Home Page:

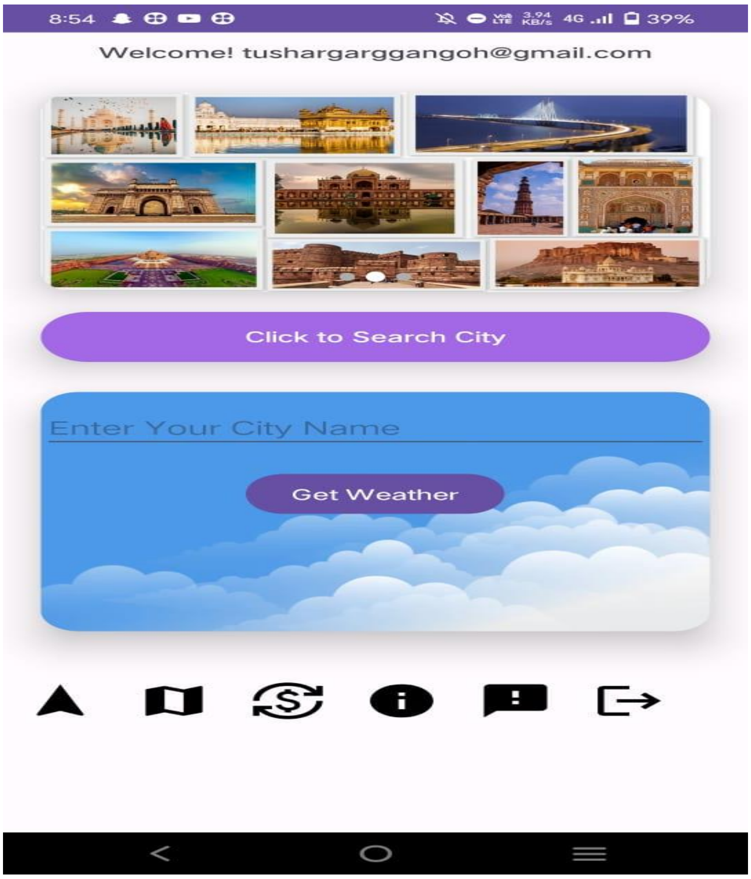
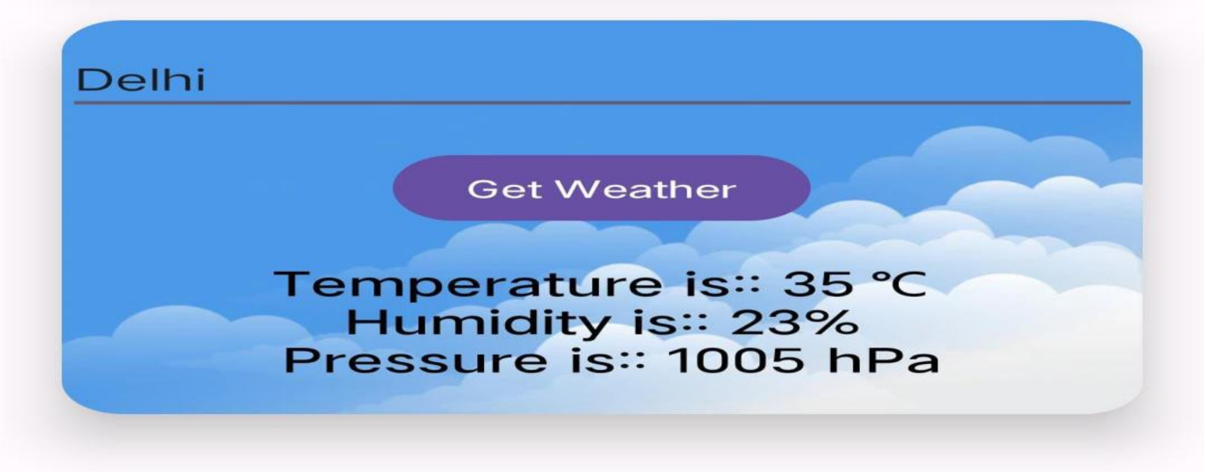


Fig. 10:: Home Page

9.4 Weather Module:



Temperature is:: 35 °C
Humidity is:: 23%
Pressure is:: 1005 hPa

Fig. 11:: Weather Module

9.5 Feedback Module:

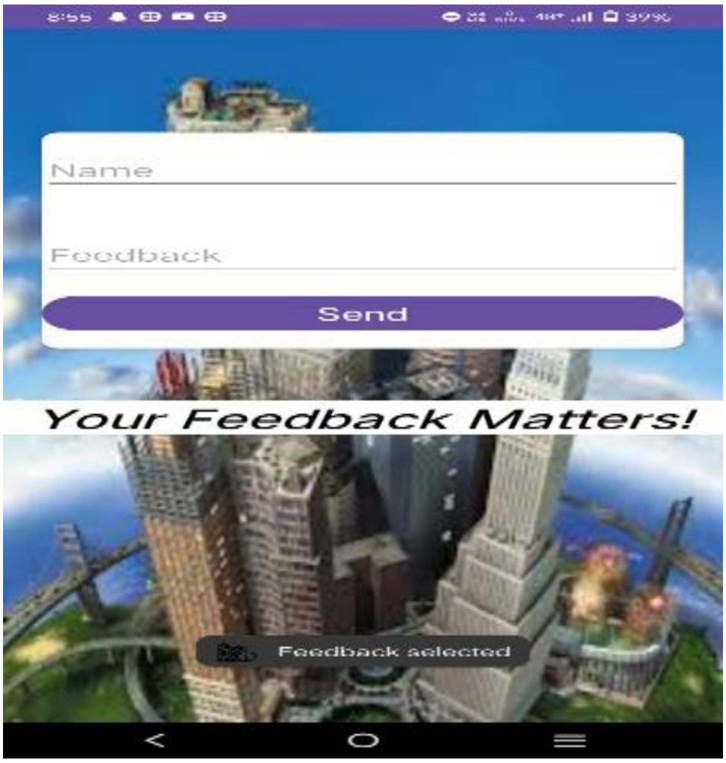


Fig.12:: Feedback Module

9.6Navigation module:

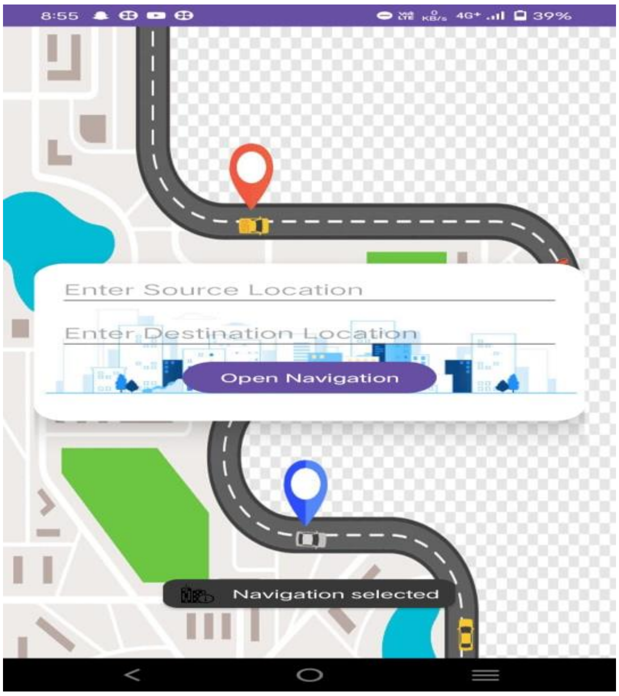


Fig. 13:: Navigation Module

9.7Map Module:

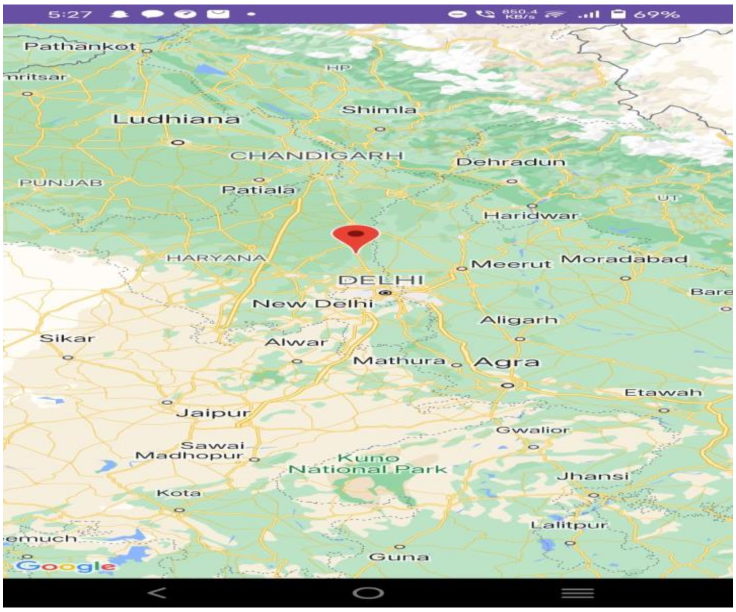


Fig. 14:: Map Module

9.8Currency Converter Module:

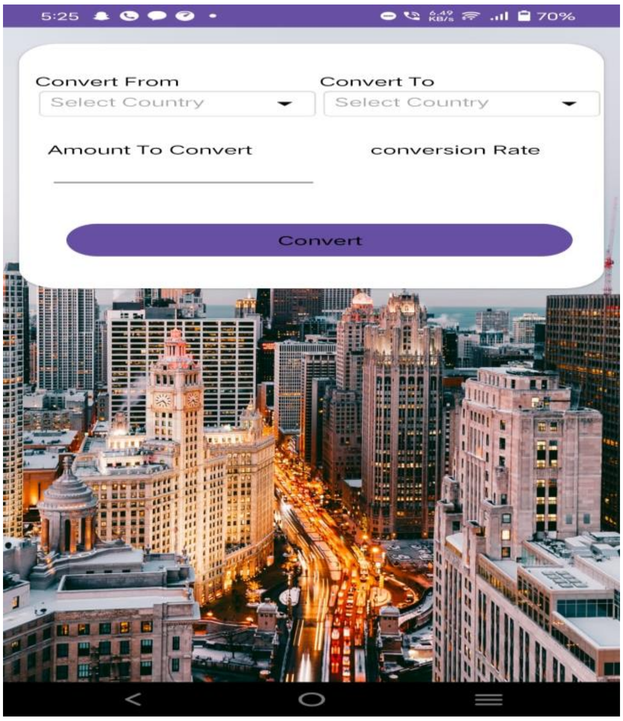
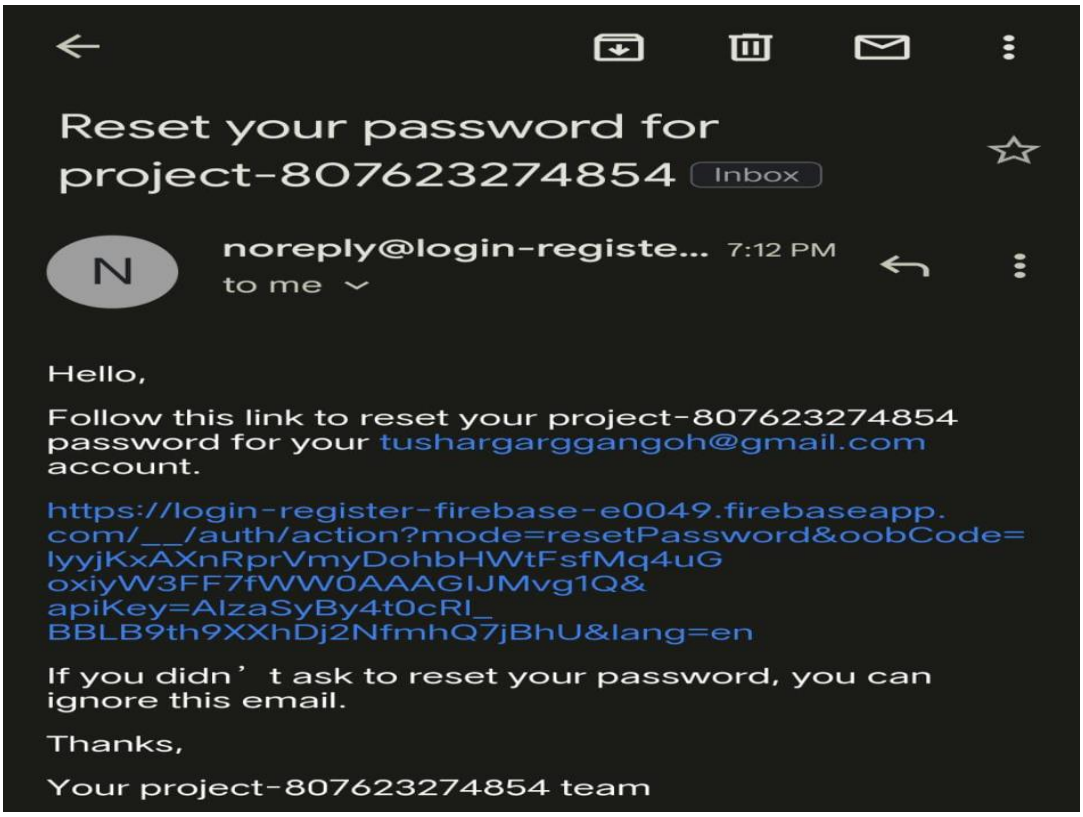


Fig. 15:: Currency Converter Module

9.9 Forgot Email:



Reset your password for project-807623274854 Inbox

noreply@login-registe... 7:12 PM
to me

Hello,

Follow this link to reset your project-807623274854 password for your tushargarggangoh@gmail.com account.

https://login-register-firebase-e0049.firebaseapp.com/__/auth/action?mode=resetPassword&oobCode=lyyjKxAXnRprVmyDohbHWtFsfMq4uGoxiyW3FF7fWW0AAAGIJMvg1Q&apiKey=AIzaSyBy4t0cRI_BBLB9th9XXhDj2NfmhQ7jBhU&lang=en

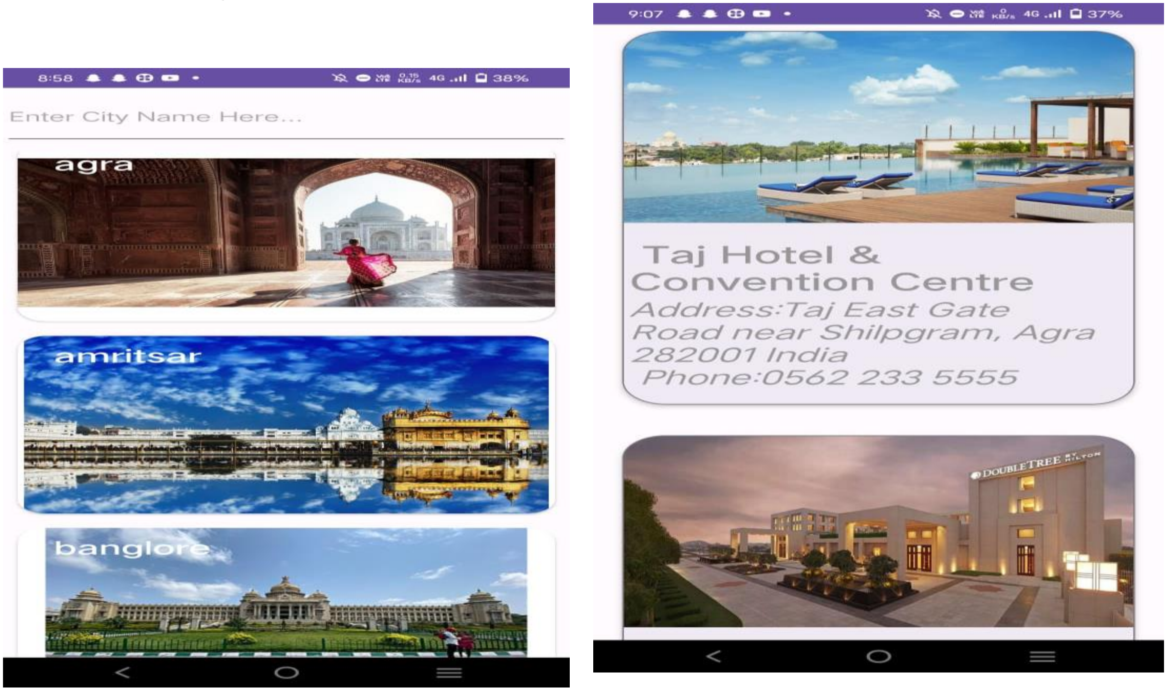
If you didn’ t ask to reset your password, you can ignore this email.

Thanks,

Your project-807623274854 team

Fig. 16:: Forgot Email

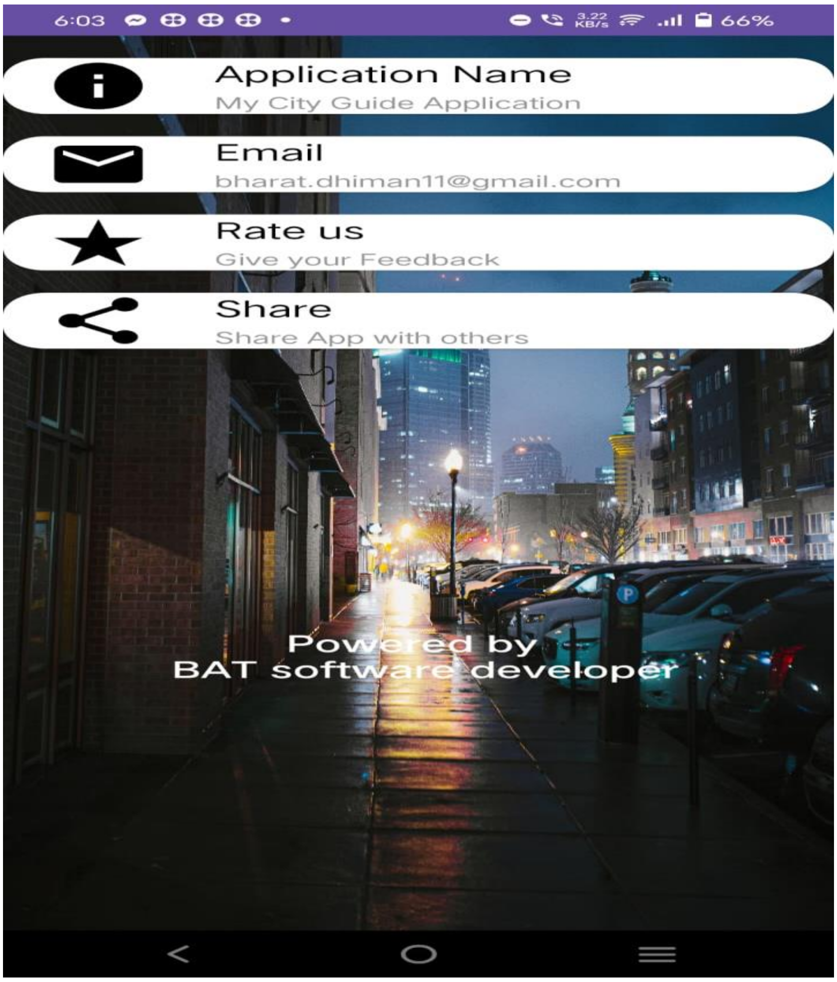
9.10 City Detail Module:



Taj Hotel & Convention Centre
Address:Taj East Gate Road near Shilpgram, Agra 282001 India
Phone:0562 233 5555

Fig. 17 ::City Detail Module

9.11About Module:



Application Name
My City Guide Application

Email
bharat.dhiman11@gmail.com

Rate us
Give your Feedback

Share
Share App with others

Powered by
BAT software developer

Fig. 18::About Module

CHAPTER 10
CONCLUSION AND FUTURE SCOPE

10.1 Conclusion

In conclusion, the city information app we have developed provides a comprehensive and user-friendly platform for individuals to access various aspects of their city. By incorporating a wide range of features and functionalities, the app aims to enhance the overall experience of residents and visitors alike.

Throughout the development process, we focused on ensuring that the app offers accurate and up-to-date information about local attractions, events, transportation options, dining establishments, and other essential services. By leveraging real-time data and integrating with relevant city databases, we aimed to provide users with the most reliable and relevant information possible.

The app's user interface is designed to be intuitive and visually appealing, enabling users to navigate through different sections seamlessly. We incorporated search functionality and filters to assist users in finding specific information quickly and efficiently. Additionally, we included interactive maps and directions to help users navigate the city and reach their desired destinations effortlessly.

One of the key features of our app is the personalized recommendations and notifications. By allowing users to set preferences and interests, the app can deliver tailored recommendations for events, places to visit, and dining options that align with their preferences. Furthermore, the app sends notifications about upcoming events, local news, and important updates, keeping users informed and engaged with their city.

The app also promotes community engagement by providing a platform for users to share their experiences, leave reviews, and connect with other users. This social aspect of the app encourages community involvement and fosters a sense of belonging among residents.

In summary, our city information app strives to be a comprehensive and user-centric platform that empowers individuals to explore and engage with their city. By providing accurate information, personalized recommendations, and fostering community interaction, we aim to enhance the overall experience and connection between users and their city.

10.2 Future Scope

City Sense will help the peoples to save their time and also save the money. We can add more interesting features like many adding many features User-generated content, such as reviews, ratings, and recommendations, can further enhance the community aspect of your Integration and restaurants, retailers, and service providers to offer exclusive deals, discounts, and promotions within the app. This not only adds value to users but also drives foot traffic and boosts the local economy.

CHAPTER 11

REFERNCES

[1] K. Leela Rani, T. Anne Esther Jashmine, S. Benita Jeba Mahar, "My City Information Mobile Application using Android Application," International Journal of Engineering Science and Computing, vol. 9, no. 30, pp. 20288-20290, 2019.

[2] J. Meng and N. Xu, "A Mobile Tourist Guide System Based on Mashup Technology," in Proc. IEEE International Conference on Computer Science and Information Technology, ISBN 978-1-4244-7618-3/10, 2010.

[3] S. Karanasios, S. Burgess, C. Sellitto, "A Classification of Mobile Tourism Applications," in Global Hospitality and Tourism Management Technologies, USA: IGI Global, 2012.

[4] R. Ramani, S. Valarmathy, Dr. N. SuthanthiraVanitha, S. Selvaraju, M. Thiruppathi, and R. Thangam, "Vehicle Tracking and Locking System Based on GSM and GPS," MECS I.J. Intelligent Systems and Applications, vol. 9, 2013.

[5] P. Verma and J.S. Bhatia, "Design and Development of GPSGSM Based Tracking System with Google Map Based Monitoring," International Journal of Computer Science, Engineering and Applications (IJCSEA), vol. 3, no. 3, June 2013.

[6] S. Sonavane, S. Agarwal, and N. Ahire, "GPS Supported Android Application for City Bus Scheduling & Tracking System," International Journal of Enhanced Research in Management & Computer Applications, vol. 3, issue 12, December 2014.