

INTEGRATION OF SMART CITY BASED ON ARTIFICIAL INTELLIGENCE SYSTEM

A PROJECT REPORT

Submitted by

THRISHA.R. J (211419205172)

SUSHMITHA.A (211419205165)

SANJHANA DINESH (211419205145)

in partial fulfillment for the award of the

degree of

BACHELOR OF TECHNOLOGY

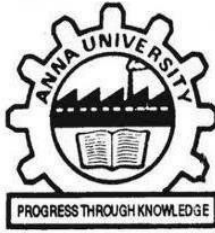
in

INFORMATION TECHNOLOGY

PANIMALAR ENGINEERING COLLEGE, POONAMALLEE

ANNA UNIVERSITY : CHENNAI 600 025

APRIL 2023



INTEGRATION OF SMART CITY BASED ON ARTIFICIAL INTELLIGENCE SYSTEM

A PROJECT REPORT

Submitted by

THRISHA.R. J (211419205172)

SUSHMITHA.A (211419205165)

SANJHANA DINESH (211419205145)

*in partial fulfillment for the award of the
degree of*

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

PANIMALAR ENGINEERING COLLEGE, POONAMALLEE

ANNA UNIVERSITY : CHENNAI 600 025

APRIL 2023

BONAFIDE CERTIFICATE

Certified that this project report “**INTEGRATION OF SMART CITY BASED ON ARTIFICIAL INTELLIGENCE SYSTEM** ” is the bonafide work of “**THRISHA.R.J(211419205172), SANJHANA DINESH(211419205145), SUSHMITHA.A(211419205165)**” who carried out the project under my supervision.

SIGNATURE

Dr. M. HELDA MERCY, M.E., (Ph.D.,)

HEAD OF THE DEPARTMENT

Department of Information Technology

Technology Panimalar Engineering College

Poonamallee, Chennai - 600 123

SIGNATURE

Mrs.S.UMA, M.Tech.,(Ph.D.,)

SUPERVISOR

Associate **Professor**

Department of Information

Panimalar Engineering College

Poonamallee, Chennai - 600 123

Submitted for the project and viva-voce examination held on _____

SIGNATURE

INTERNAL EXAMINER

SIGNATURE

EXTERNAL EXAMINER

DECLARATION

I hereby declare that the project report entitled “**INTEGRATION OF SMART CITY BASED ON ARTIFICIAL INTELLIGENCE SYSTEM**” which is being submitted in partial fulfilment of the requirement of the course leading to the award of the ‘Bachelor Of Technology in Information Technology ’ in **Panimalar Engineering College, An Autonomous institution Affiliated to Anna university- Chennai** is the result of the project carried out by me under the guidance of **Mrs.S.UMA in the Department of Information Technology**. I further declared that I or any other person has not previously submitted this project report to any other institution/university for any other degree/ diploma or any other person.

THRISHA.R.J

Date:

SANJHANA DINESH

Place: Chennai

SUSHMITHA.A

It is certified that this project has been prepared and submitted under my guidance.

Date:

(Mrs.S.UMA)

Place: Chennai

(M.Tech .,Ph.D., / IT)

ACKNOWLEDGEMENT

A project of this magnitude and nature requires kind co-operation and support from many, for successful completion . We wish to express our sincere thanks to all those who were involved in the completion of this project.

Our sincere thanks to **Our Beloved Secretary and Correspondent, Dr. P. CHINNADURAI, M.A., Ph.D.,** for his sincere endeavor in educating us in his premier institution.

We would like to express our deep gratitude to **Our Dynamic Directors, Mrs.C. VIJAYA RAJESHWARI and Dr. C. SAKTHI KUMAR,M.E.,M.B.A., Ph.D., and Dr.SARANYA SREE SAKTHIKUMAR .,B.E.,M.B.A., Ph.D.,** for providing us with the necessary facilities for completionof this project.

We also express our appreciation and gratefulness to **Our Principal Dr. K. MANI, M.E., Ph.D.,** who helped us in the completion of the project. We wish to convey our thanks and gratitude to our head of the department, **Dr. M. HELDA MERCY, M.E., Ph.D.,** Department of Information Technology, for her support and by providing us ample time to complete our project.

We express our indebtedness and gratitude to our Project co-ordinator **Mr. M. DILLI BABU, M.E.,(P.hD.,)** Associate Professor, Department of Information Technology for his guidance throughout the course of our project. We also express sincere thanks to our supervisor **MRS.S.UMA M.Tech(Ph.D.,)** for providing the support to carry out theproject successfully. Last, we thank our parents and friends for providing their extensive moral support and encouragement during the course of the project.

ABSTRACT

Now a day mobile phone is a needful part of the people's life. There is continuously rising in a number of mobile computing applications, concentrate on the people's daily life. In such applications, location dependent systems have been detected as a significant application. Such application which presents the architecture and implementation of such a location is commonly known as Smart City Guide. The main motive of the project is to explore how to realize a mobile city guide using the Android platform, including a prototype of the city guide. The project uses the research method design science. Through designing and implementing an artifact, the goal project is reached. Finally, the project is assess in four aspects including platform evaluation, general functional evaluation, scenario evaluation and non-functional evaluation. The prototype implemented include basic functionalities of city guides such as showing the map, locating points of interest (POIs). This project explores the benefits of integrating smart city technologies, including improved energy efficiency, reduced traffic congestion, and enhanced public safety. It also examines the challenges associated with integrating these technologies, including concerns related to privacy, security, and data management. Beside, the project has inspect how to combine present technologies like Google Map and the phone application into the prototype. The app comforts a new native in a city by showing information of all the nearby sites that can be used for public access. Sites include Hospital Services, Police Station, Main Attraction Of City, Famous Restaurants. As well, the project has investigated non-functional aspects including extendibility, tolerability, and usability. Overall, the project presents a comprehensive unrealized city guide on the new mobile Android platform.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF TABLES	vi
	LIST OF FIGURES	vi
	LIST OF ABBREVIATIONS	vii
1	INTRODUCTION	11
	1.1 OVERVIEW OF THE PROJECT	11
	1.2 SCOPE OF THE PROJECT	11
	1.3 OBJECTIVE OF THE PROJECT	12
	1.4 PURPOSE OF THE PROJECT	12
2	LITERATURE SURVEY	13
	2.1 SURVEY ON MOBILE USER'S DATA PRIVACY THREATS AND DEFENCE MECHANISMS	13
	2.2 THE SMART CITY INFRASTRUCTURE DEVELOPMENT & MONITORING	15
	2.3 SMART HEALTHCARE: MAKING MEDICAL CARE MORE INTELLIGENT	17
	2.4 OPERATIONAL VULNERABILITIES FOR SMART CITIES	19
	2.5 USE AND ADOPTION OF SMART SERVICES BY CITIZENS (SUCCESS OF SMART SERVICES)	21
3	SYSTEM ANALYSIS	23
	3.1 EXISTING SYSTEM	23
	3.2 PROPOSED SYSTEM	24
4	REQUIREMENT SPECIFICATION	25
	4.1 HARDWARE REQUIREMENTS	25
	4.2 SOFTWARE REQUIREMENTS	25
	4.2.1 ANDROID	25
	4.2.2 WINDOWS	26
	4.2.3 ANDROID STUDIO	27

	4.2.4 OVERVIEW OF ECLIPSE	29
	4.2.5 JAVA	32
	4.3 REQUIREMENT ANALYSIS	37
	4.3.1 FUNCTIONAL REQUIREMENTS	37
	4.3.2 NON FUNCTIONAL REQUIREMENTS	38
5	FEASIBILITY STUDY	40
	5.1 ECONOMICAL FEASIBILITY	40
	5.2 TECHNICAL FEASIBILITY	41
	5.3 SOCIAL FEASIBILITY	41
6	MODULE DESCRIPTION	42
	6.1 SMART CITY GUIDE	42
	6.2 FIND DESTINATION LOCATION	43
	6.3 TOURISM AND CITY GUIDE	45
	6.4 MAIN INFORMATION ABOUT SMART CITIES	46
7	SYSTEM DESIGN AND TESTING PLAN	47
	7.1 INPUT DESIGN	47
	7.2 OUTPUT DESIGN	47
	7.3 SYSTEM TESTING	48
	7.3.1 TEST PLAN	48
	7.3.2 VERIFICATION	48
	7.3.3 VALIDATION	48
	7.3.4 BASICS OF SOFTWARE TESTING	49
	7.3.5 TYPES OF TESTING	49
8	SYSTEM DEVELOPMENT	52
	8.1 SYSTEM ARCHITECTURE	52
	8.2 DATA FLOW DIAGRAM	52
	8.3 ER DIAGRAM	54
	8.4 UML DIAGRAMS	54
	8.4.1 USE CASE DIAGRAM	55
	8.4.2 CLASS DIAGRAM	56
	8.4.3 SEQUENCE DIAGRAM	56

	8.4.4 ACTIVITY DIAGRAM	58
	8.4.5 COLLABORATION DIAGRAM	59
10	IMPLEMENTATION	60
	10.1 SAMPLE CODE	60
	10.2 OUTPUT SCREENSHOTS	92
11	CONCLUSION AND FUTURE ENHANCEMENT	96
	11.1 CONCLUSION	96
	11.2 FUTURE SCOPE	96
	REFERENCES	97

LIST OF FIGURES

S.NO	NAME OF FIGURES
1	Collection framework
2	System architecture
3	Data flow diagram
4	ER diagram
5	Usecase diagram
6	Class diagram
7	Sequence diagram
8	Activity diagram
9	Colloboration diagram

LIST OF ABBREVIATIONS

POI	Points of Interest
GIS	Geographic Information System
IDE	Integrated development environment
ADT	Android Development Tools
MTBF	Mean Time Between Failures
EPL	Eclipse Public License
UML	Unified Modeling Language
IoT	Internet of things
JCP	Java Community Process
JSR	Java Specification Requests

1.INTRODUCTION

1.1 OVERVIEW

City Guide is an essential whenever we are visiting a particular city. It gives us the valuable information about the city and saves the time. The project title is 'A COMPLETE CITY GUIDE USING DATABASE', a web-based platform used to store the details of particular city and helps all the users who have just visits our website and registered in the site, they can search for a prominent places in the city without taking any help from personal guide. The website contains the complete information about particular city like places to be visited, site maps route maps, Business environment, Job portal, information about organization that provide transport, Hospitality and total history of the city. This website can be used by any person who is having general knowledge about internet. All the users will be first considered as anonymous user later if he needs any service then he will be treated as registered user.

1.2 SCOPE OF THE PROJECT

The scope for integrating smart city technologies is broad and includes a range of sectors, including transportation, energy, healthcare, public safety, and education. Smart city technologies can be applied to various urban infrastructure systems, including buildings, transportation networks, and utilities, to improve their efficiency, reliability, and sustainability. It can be accessed by unlimited number of users. Each user will be assigned a different set of permissions for each module of the system. The user can have access to all the information in the site with limited services and provide extra services to registered users. Track all the transaction details of the customer. Confirmation of end user identity and will verify which users are authorized to receive Support.

1.3 OBJECTIVE

The prototype implemented include basic functionalities of city guides such as showing the map, locating points of interest (POIs). Beside, the project has inspect how to combine present technologies like Google Map and the phone application into the prototype. The app comforts a new native in a city by showing information of all the nearby sites that can be used for public access.

1.4 PURPOSE

The purpose of integrating smart city projects is to use technology and data-driven solutions to improve the quality of life for residents, enhance sustainability, and increase efficiency of city services. Smart cities aim to make urban areas more livable, workable, and sustainable by leveraging advances in technology and innovation.

Some of the goals of smart city projects include:

- Enhancing the quality of life for residents by providing smart infrastructure, such as public transportation systems, energy-efficient buildings, and smart waste management.
- Improving public safety by using data analytics to monitor and respond to crime and emergencies in real-time.
- Enhancing sustainability by reducing energy consumption, carbon emissions, and waste generation.
- Improving the efficiency and effectiveness of public services by using data analytics to optimize service delivery.
- Creating a more connected and inclusive city by providing digital services and access to information for all residents.

2. LITERATURE SURVEY

2.1.SURVEY ON MOBILE USER'S DATA PRIVACY THREATS AND DEFENCE MECHANISMS

Author: Jalaluddin Khana , Haider Abbasa,b*, Jalal Al-Muhtadi

Year: 2020 Publication : International conference on system modeling & advance in research

CONCEPT

Nowadays, mobile devices have become an integral part of our daily life. These have proven to be an advantageous scientific invention that fills personal and business needs in a very efficient manner. In this era, the availability of mobile services has significantly increased because of the rich variety of mobile devices and essential applications provided by mobile device manufacturers. At the same time, numerous mobile security issues and data privacy threats are challenging both manufacturers and users.

PROBLEM IDENTIFICATION

Therefore, mobile devices are an ideal target for various security issues and data privacy threats in a mobile ecosystem. In this paper, we provide a brief survey of the security challenges, threats, and vulnerabilities of a mobile ecosystem. Furthermore, we discussed some key points required to ensure mobile security and defend against data privacy threats.

KNOWLEDGE GAINED

Mobile Security; Malware; Data Privacy; Threats and Vulnerabilities

WORK DONE

The emphasis of the discussion is, strong protection and the restriction of malicious activity at the application developer end, application stores end, and operating system and mobile device manufactures end by preventing the user from using non-recommended applications (which may be malicious) and considering biometric features for the authentication of real users in the mobile devices. Also briefly discussing the defence mechanisms that are considered to be a relatively better approach for securing personal and business related data or information in the mobile devices.

GAP

Mobile device applications offer a level of convenience that the world never before considered. At any location (home, office, hotel, playground, road, parking, museum, travelling in different countries, or anyplace in the world), any mobile user can use applications to fulfil their daily needs, including communicating, buying, searching, making payments, selling, entertainment, and finding general information. This extreme level of comfort has brought with it an extreme number of security risks.

2.2 THE SMART CITY INFRASTRUCTURE DEVELOPMENT & MONITORING

Author: Mahmoud AL-HADER, Ahmad RODZI .

Year: 2014 Publication : International conference on system modelling & advance in research

CONCEPT

The smart city infrastructure is the introductory step for establishing the overall smart city framework and architecture. Very few smart cities are recently established across the world. Some examples are: Dubai, Malta, Kochi (India), Singapore. The scope of these cities is mainly limited to construct a technology park converting the industrial real estate to state of the art information technology using the evolution in the telecom and IP networks including insignificant asset management automation system.

PROBLEM IDENTIFICATION

The development background is to create an operational platform that would manage the power consumption and operational resources in order to reduce the overall running operational cost.

KNOWLEDGE GAINED

Smart Infrastructure, GIS, Smart City, Geopsatial application, Infrastructure Development, Infrastructure Monitoring.

WORKDONE

This paper will debate the smart infrastructure development framework and the surveying positional accuracy of locating the assets as a base of the smart city development architecture integrated with all the facilities and systems related to the smart city framework. The paper will discuss also the main advantages of the proposed architecture including the quantifiable and non-quantifiable benefits.

GAP

In order to provide a high level advisory services on emerging the smart state issues and trends. The potential in the master real estate developers are the scale of the projects and the huge man power that they are utilizing in their real estate development projects. Accordingly they are building several cities across the world. Due to these wide range real estate development activities, a detailed study of the existing systems with all associated database engines and business platforms are need to be tackled

2.3.SMART HEALTHCARE: MAKING MEDICAL CARE MORE INTELLIGENT

Author:ShuoTian¹WenboYang¹JehaneMichael
LeGrangePengWangWeiHuangZheweiYe.

Year: 2019 Publication : International conference on system modelling & advance in research

CONCEPT

With the development of information technology, the concept of smart healthcare has gradually come to the fore. Smart healthcare uses a new generation of information technologies, such as the internet of things (IoT), big data, cloud computing, and artificial intelligence, to transform the traditional medical system in an all-round way, making healthcare more efficient, more convenient, and more personalized.

PROBLEM IDENTIFICATION

Smart healthcare uses a new generation of information technologies, such as the internet of things (IoT), big data, cloud computing, and artificial intelligence, to transform the traditional medical system in an all-round way, making healthcare more efficient, more convenient, and more personalized.

KNOWLEDGE GAINED

Smart healthcare, Informatization, Health management, Surgery, Clinical decision, Smart equipment, Personalization, Hospital management, Internet of things

WORK DONE

We first list the key technologies that support smart healthcare and introduce the current status of smart healthcare in several important fields. Then we expound the existing problems with smart healthcare and try to propose solutions to them. Finally, we look ahead and evaluate the future prospects of smart healthcare.

GAP

Smart healthcare integration refers to the use of technology and data to improve the quality of healthcare services and patient outcomes. The integration of smart healthcare technologies can help healthcare providers to deliver more efficient, effective, and patient-centered care. Smart healthcare incorporating a new generation of information technology has emerged. Smart healthcare is not just a simple technological advancement, but also an all-round, multi-level change.

2.4 OPERATIONAL VULNERABILITIES FOR SMART

CITIES Author: Rob Kitchin¹ and Martin Dodge²

Year: 2017 Publication : International conference on system modelling & advance in research

CONCEPT

In this paper we examine the current state of play with regards to the security of smart city initiatives. Smart city technologies are promoted as an effective way to counter and manage uncertainty and urban risks through the effective and efficient delivery of services, yet paradoxically they create new vulnerabilities and threats, including making city infrastructure and services insecure, brittle, and open to extended forms of criminal activity.

PROBLEM IDENTIFICATION

Smart city technologies are no different being afflicted with a range of security vulnerabilities and risks and an on-going struggle is now evident between the cybersecurity industry and criminals and variously-motivated hackers. However, while the base motivations to break into these systems might remain timeless (e.g., theft, impersonation, vandalism, malicious attack; see Schneier, 2003), the nature of their performance is different.

KNOWLEDGE GAINED

Crime, cyber attacks, mitigation, risk, security, smart cities, urban resilience

WORKDONE

We identify five forms of vulnerabilities with respect to smart city technologies, detail the present extent of cyberattacks on networked infrastructure and services, and present a number of illustrative examples. We then adopt a normative approach to explore existing mitigation strategies, suggesting a wider set of systemic interventions (including security-by-design, remedial security patching and replacement, formation of core security and computer emergency response teams, a change in procurement procedures, and continuing professional development). We discuss how this approach might be enacted and enforced through market-led and regulation/management measures, and examine a more radical preventative approach to security.

GAP

There are two key security risks with respect to the emergence of smart cities. The first is the security of newly installed ‘intelligent’ technologies and ‘smart’ upgrades to existing infrastructures

2.5 USE AND ADOPTION OF SMART SERVICES BY CITIZENS (SUCCESS OF SMART SERVICES)

Author: Belanche-Gracia et al. [2015](#); Chatterjee et al. [2018](#); Cilliers and Flowerday [2014](#); Cilliers and Flowerday [2015](#); Van Heek et al. [2016](#); van Zoonen [2016](#) Year: 2017 Publication : International conference on system modelling & advance in research

Year: 2018

CONCEPT

Privacy is a core issue within smart cities and one that can be directly linked to the minimal understanding of privacy from local government and business in the way they collect and process personal data. Often they do not provide the community with the opportunity and mechanism for consent

PROBLEM IDENTIFICATION

It was found that perceived security and privacy significantly affect the use and adoption of smart services by citizens.

KNOWLEDGE GAINED

A number of studies highlighted the importance of perceived security and privacy in smart cities services by citizens

WORKDONE

The study explored how specific technologies (smart bin, smart parking), and data usage (predictive policing, social media monitoring) may produce various privacy concerns.

GAP

Studies differ on the extent of privacy concerns depending on the type of technologies, data usage and location. According to van Zoonen ([2016](#)) there are four areas of concern amongst people in smart cities that range from low levels (impersonal data, service purpose), to extremely high (personal data, surveillance purpose).

3. SYSTEM ANALYSIS

3.1. EXISTING SYSTEM

In Existing system the person who are visiting a particular city need to gather information from the person who is staying in the city or take the help of the guide in the city. Gather of all these information you need to visit the city. This possess a lot of time and pre-planning. In order to get each piece of information we need to go for help desk.

LIMITATIONS OF EXISTING SYSTEM

- The existing system is a manual system. Here the city information needs to save his information in the form of excel sheets or Disk Drives.
- There is no sharing is possible if the data is in the form of paper or Disk drives.
- The integration of smart city technologies requires significant investments in infrastructure, including sensors, communication networks, and data centers.
- The collection and use of large amounts of data in smart city systems raise concerns about data privacy and security.
- The cost of implementing these technologies may be higher for cities in developing countries, which may have limited financial resources.
- Cities may struggle to find the necessary expertise and resources to implement these initiatives successfully.
- The manual system gives us very less security for saving data; some data may be lost due to mismanagement.
- It's a limited system and fewer users friendly.

3.2. PROPOSED SYSTEM

- The Proposed System provides an online information about the particular city going to visit. The development of this new system contains the following activities, which try to automate the entire process keeping in the view of database integration approach.
- User Friendliness is provided in the application with various controls provided by system Rich User Interface.
- The system makes the overall project management much easier and flexible.
- It can be accessed over the Intranet.
- The city information files can be stored in centralized database which can be maintained by the system.

Integrating a smart city system requires the implementation of various technologies that can efficiently and effectively improve the quality of life of citizens.

developing a proposed system for integration of smart city requires careful planning, implementation, and maintenance. By integrating the latest technologies, engaging citizens, and optimizing resource utilization, a smart city can improve the quality of life for citizens, and pave the way for a sustainable future.

4. SYTEM REQUIREMENTS

4.1. HARDWARE REQUIREMENTS

System -

Pentium-IV Speed -

2.4GHZ Hard disk -

40GB RAM -

512MB

4.2. SOFTWARE REQUIREMENTS

Operating System - Windows

XP Coding language - Java

IDE – Android Studio

4.2.1 ANDRIOD

PENTIUM DUAL

CORE

The Pentium Dual-Core brand was used for mainstream x86-architecture microprocessors from Intel from 2006 to 2009 when it was renamed to Pentium. The processors are based on either the 32-bit *Yonah* or (with quite different microarchitectures) 64-bit *Merom-2M*, *Allendale*, and *Wolfdale-3M* core, targeted at mobile or desktop computers.

4.2.2. WINDOWS

Microsoft Windows is a group of several graphical operating system families, all of which are developed, marketed, and sold by Microsoft. Each family catersto a certain sector of the computing industry.

Active Windows families include Windows NT and Windows Embedded; these may encompass subfamilies, e.g. Windows Embedded Compact (Windows CE)

or Windows Server. Defunct Windows families include Windows 9x, Windows Mobile and Windows Phone.

4.2.3. ANDROID STUDIO

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. The current stable version is 3.3, which was released in January 2019.

Android is a mobile operating system (OS) based on the Linux kernel and currently developed by Google. With a user interface based on direct manipulation, Android is designed primarily for touch screen mobile devices such as smart phones and tablet computers, with specialized user interfaces for televisions (Android TV), cars (Android Auto), and wrist watches (Android Wear).

The OS uses touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, and a virtual keyboard. Despite being primarily designed for touch screen input, it has also been used in game consoles, digital cameras, regular PCs, and other electronics.

As of 2015, Android has the largest installed base of all operating systems. As of July 2013, the Google Play store has had over one million Android applications ("apps") published, and over 50 billion applications downloaded. An April–May 2013 survey of mobile application developers found that 71% of them create applications for Android; another 2015 survey found that 40% of full-time professional developers see Android as the "priority" target platform, which is more than iOS (37%) or other platforms.

At Google I/O 2014, the company revealed that there were over one billion active monthly Android users, up from 538 million in June 2013. Android's source code is released by Google under open source licenses, although most Android devices ultimately ship with a combination of open source and proprietary software, including proprietary software developed and licensed by Google.

Initially developed by Android, Inc., which Google bought in 2005, Android was unveiled in 2007, along with the founding of the Open Handset Alliance – a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices. Android is popular with technology

companies which require a ready-made, low-cost and customizable operating system for high-tech devices.

Android's open nature has encouraged a large community of developers and enthusiasts to use the open-source code as a foundation for community-driven projects, which add new features for advanced users or bring Android to devices which were officially released running other operating systems. The operating system's success has made it a target for patent litigation as part of the so-called "smartphone wars" between technology companies.

4.2.4. OVERVIEW OF ECLIPSE:

In the context of computing, Eclipse is an integrated development environment (IDE) for developing applications using the Java programming language and other programming languages such as C/C++, Python, PERL, Ruby etc.

The Eclipse platform which provides the foundation for the Eclipse IDE is composed of plug-ins and is designed to be extensible using additional plug-ins. Developed using Java, the Eclipse platform can be used to develop rich client applications, integrated development environments and other tools. Eclipse can be used as an IDE for any programming language for which a plug-in is available.

The Java Development Tools (JDT) project provides a plug-in that allows Eclipse to be used as a Java IDE, PyDev is a plugin that allows Eclipse to be used as a Python IDE, C/C++ Development Tools (CDT) is a plug-in that allows Eclipse to be used for developing application using C/C++, the Eclipse Scala plug-in allows Eclipse to be used an IDE to develop Scala applications and PHPEclipse is a plug-in to eclipse that provides complete development tool for PHP.

Licensing

Eclipse platform and other plug-ins from the Eclipse foundation is released under the Eclipse Public License (EPL). EPL ensures that Eclipse is free to download and install. It also allows Eclipse to be modified and distributed.

Eclipse Releases

Every year, since 2006, the Eclipse foundation releases the Eclipse Platform and a number of other plug-ins in June.

Eclipse Platform Technical Overview

The Eclipse Platform (or simply "the Platform" when there is no risk of confusion) is designed and built to meet the following requirements:

- Support the construction of a variety of tools for application development.
- Support an unrestricted set of tool providers, including independent software vendors (ISVs).
- Support tools to manipulate arbitrary content types (e.g., HTML, Java, C, JSP, EJB, XML, and GIF).
- Facilitate seamless integration of tools within and across different content types and tool providers.
- Support both GUI and non-GUI-based application development environments.
- Run on a wide range of operating systems, including Windows®, Linux™, Mac OS X, Solaris AIX, and HP-UX.
- Capitalize on the popularity of the Java programming language for writing tools.

The Eclipse Platform's principal role is to provide tool providers with mechanisms to use, and rules to follow, that lead to seamlessly-integrated tools. These mechanisms are exposed via well-defined API interfaces, classes, and methods. The Platform also provides useful building blocks and frameworks that facilitate developing new tools.

4.2.5. JAVA

The Java language has undergone several changes since JDK 1.0 as well as numerous additions of classes and packages to the standard library. Since J2SE 1.4, the evolution of the Java language has been governed by the Java Community Process (JCP), which uses Java Specification Requests (JSRs) to

INTRODUCTION

Java is one of the world's most important and widely used computer languages, and it has held this distinction for many years. Unlike some other computer languages whose influence has wearied with passage of time, while Java's has grown.

APPLICATION OF JAVA

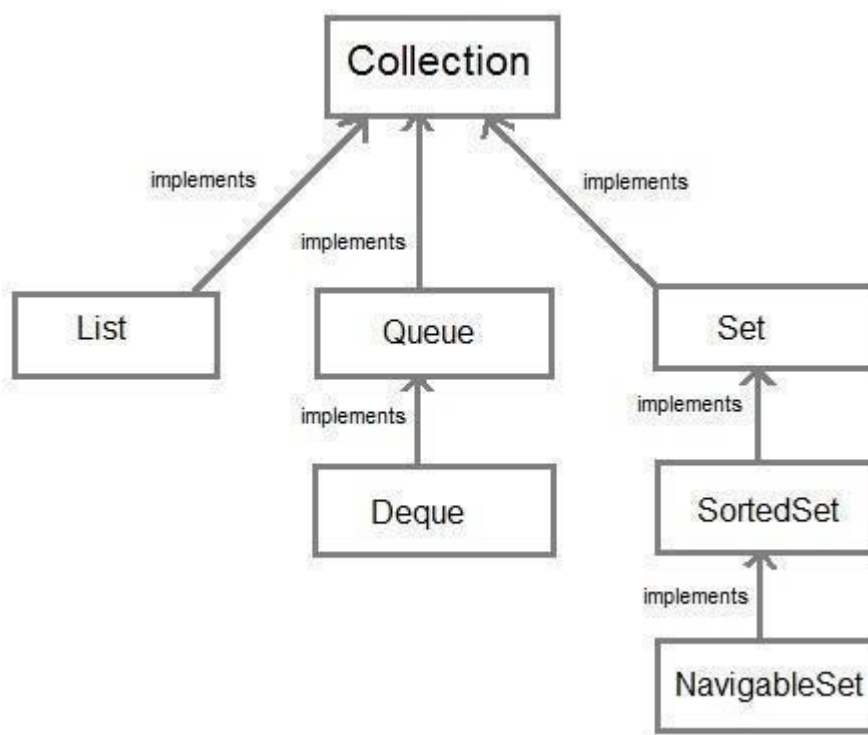
Java is widely used in every corner of world and of human life. Java is not only used in softwares but is also widely used in designing hardware controlling software components. There are more than 930 million JRE downloads each year and 3 billion mobile phones run java.

Following are some other usage of Java:

1. Developing Desktop Applications
2. Web Applications like Linkedin.com, Snapdeal.com etc
3. Mobile Operating System like Android
4. Embedded Systems
5. Robotics and games etc.

FEATURES OF JAVA

The prime reason behind creation of Java was to bring portability and security feature into a computer language. Beside these two major features, there were many other features that played an important role in moulding out the final form of this outstanding language.



COLLECTION FRAMEWORK

Collection framework was not part of original Java release. Collections was added to J2SE 1.2. Prior to Java 2, Java provided adhoc classes such as Dictionary, Vector, Stack and Properties to store and manipulate groups of objects. Collection framework provides many important classes and interfaces to collect and organize group of alike objects.

4.3. REQUIREMENT ANALYSIS

Requirement analysis, also called requirement engineering, is the process of determining user expectations for a new modified product. It encompasses the tasks that determine the need for analysing, documenting, validating and managing software or system requirements. The requirements should be documentable, actionable, measurable, testable and traceable related to identified business needs or opportunities and define to a level of detail, sufficient for system design.

4.3.1. FUNCTIONAL REQUIREMENTS

It is a technical specification requirement for the software products. It is the first step in the requirement analysis process which lists the requirements of particular software systems including functional, performance and security requirements. The function of the system depends mainly on the quality hardware used to run the software with given functionality.

Usability

It specifies how easy the system must be use. It is easy to ask queries in any format which is short or long, porter stemming algorithm stimulates the desired response for user.

Robustness

It refers to a program that performs well not only under ordinary conditions but also under unusual conditions. It is the ability of the user to cope with errors for irrelevant queries during execution.

Security

The state of providing protected access to resource is security. The system provides good security and unauthorized users cannot access the system there by providing high security.

Reliability

It is the probability of how often the software fails. The measurement is often expressed in MTBF (Mean Time Between Failures). The requirement is needed in order to ensure that the processes work correctly and completely without being aborted. It can handle any load and survive and survive and even capable of working around any failure.

Compatibility

It is supported by version above all web browsers. Using any web servers like localhost makes the system real-time experience.

Flexibility

The flexibility of the project is provided in such a way that is has the ability to run on different environments being executed by different users.

Safety

Safety is a measure taken to prevent trouble. Every query is processed in a secured manner without letting others to know one's personal information.

4.3.2. NON- FUNCTIONAL REQUIREMENTS

Portability

It is the usability of the same software in different environments. The project can be run in any operating system.

Performance

These requirements determine the resources required, time interval, throughput and everything that deals with the performance of the system.

Accuracy

The result of the requesting query is very accurate and high speed of retrieving information. The degree of security provided by the system is high and effective.

Maintainability

Project is simple as further updates can be easily done without affecting its stability. Maintainability basically defines that how easy it is to maintain the system. It means that how easy it is to maintain the system, analyse, change and test the application. Maintainability of this project is simple as further updates can be easily done without affecting its stability.

5.FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

The feasibility study investigates the problem and the information needs of the stakeholders. It seeks to determine the resources required to provide an information systems solution, the cost and benefits of such a solution, and the feasibility of such a solution.

The goal of the feasibility study is to consider alternative information systems solutions, evaluate their feasibility, and propose the alternative most suitable to the organization. The feasibility of a proposed solution is evaluated in terms of its components.

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system is well within the budget and this was achieved because most of the technologies used are freely available.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

Before integrating a smart city system, it is essential to conduct a feasibility study to evaluate its potential success and identify any potential risks. A feasibility study is an essential step in evaluating the potential success of integrating a smart city system. By considering the technical, economic, social, environmental, legal, and operational feasibility, decision-makers can identify the potential risks and benefits and make informed decisions about integrating a smart city system.

6. MODULUE DESCRIPTION

- Smart City Guide
- Find destination location
- Tourism And City Guide
- Main information about smart cities

6.1. SMART CITY GUIDE

Smart city is defined by IBM as the use of information and communication technology to sense, analyse and integrate the key information of core systems in running cities. At the same time, smart city can make the information about main attraction of the city, hospital services, emergency contact numbers, famous restaurants of that city.

Integrating a smart city project can be a complex and challenging process. Here are some general steps and considerations that can guide you through the process:

- Develop a vision and strategy: Before starting any smart city project, it is important to have a clear vision and strategy. Identify the main goals and objectives, and determine how technology can help achieve these goals.

- Engage stakeholders: Engage with stakeholders such as government agencies, community groups, and private sector organizations to build support and gather input. Stakeholders can provide valuable insights into the needs and priorities of the community.
- Conduct a needs assessment: Conduct a needs assessment to identify the current infrastructure, resources, and technology available in the city. This will help identify gaps and opportunities for improvement.
- Identify and prioritize projects: Based on the needs assessment, identify potential smart city projects and prioritize them based on their potential impact, feasibility, and cost.
- Develop a plan and budget: Develop a detailed plan and budget for each project, including timelines, resource requirements, and expected outcomes.
- Select technology vendors and partners: Identify technology vendors and partners that can provide the necessary hardware, software, and expertise to implement the projects.
- Pilot and test projects: Pilot and test the projects in a small-scale setting to identify any issues or challenges and refine the implementation plan.
- Scale up and monitor: Once the projects have been successfully piloted, scale them up to the full implementation level. Monitor the projects to ensure they are meeting their intended goals and make any necessary adjustments.
- Evaluate and improve: Evaluate the impact of the smart city projects and identify areas for improvement. Use the feedback to refine the projects and develop new ones to continue to advance the city's smart city vision.

Remember, smart city projects are not just about technology but are also about improving the quality of life for residents, enhancing sustainability, and increasing efficiency of city services. Therefore, involving the community and stakeholders, as well as ensuring that the projects align with the city's overall vision and strategy, is crucial for success.

6.2. FIND DESTINATION LOCATION

This Project in Java provides info regarding the various aspects of city such as tourism, institutes, industry, geographical maps, ATM locations, etc. The implementation of this project solves most of the problems a new visitor faces while coming to a new city.

Finding the destination location for a smart city integration project will depend on the goals and objectives of the project. Here are some general steps and considerations that can guide you in finding the destination location:

- **Identify the needs and priorities:** Identify the needs and priorities of the city, including areas that require infrastructure improvements, areas with high levels of congestion, areas with high crime rates, and areas with inadequate access to public services.
- **Conduct a data analysis:** Conduct a data analysis to identify patterns and trends in the city's data. This can include data on traffic, public transportation, crime rates, energy consumption, and water usage.
- **Engage with stakeholders:** Engage with stakeholders such as government agencies, community groups, and private sector organizations to gather input on the areas that require attention and the potential solutions.

- Identify potential projects: Based on the data analysis and stakeholder engagement, identify potential smart city projects that can address the identified needs and priorities.
- Evaluate feasibility: Evaluate the feasibility of each project, including the cost, timeline, and potential impact.
- Prioritize projects: Prioritize the projects based on their potential impact, feasibility, and alignment with the city's overall vision and strategy.
- Identify a location: Based on the prioritized projects, identify a location for the smart city integration project. This could be a specific area of the city or a group of projects that can be implemented in a specific order.
- Develop a plan and budget: Develop a detailed plan and budget for the smart city integration project, including timelines, resource requirements, and expected outcomes.
- Monitor and evaluate: Monitor the implementation of the project to ensure that it is meeting its intended goals and make any necessary adjustments. Evaluate the impact of the project and use the feedback to refine the project and develop new ones.

Overall, finding the destination location for a smart city integration project requires careful consideration of the city's needs and priorities, as well as engagement with stakeholders and evaluation of feasibility and impact.

6.3. TOURISM AND CITY GUIDE

It provides the detailed information about the area, the famous places of those area, restaurant, hotels, shopping mall etc. and all the related details of these places. This provides the user very easy way to visit any place.

Tourism and city guides are essential tools for travellers to explore and experience a city's unique culture, history, attractions, and amenities. Here are some tips on how to create effective tourism and city guides:

- **Research:** Conduct research on the city's history, culture, attractions, and events. This will help you create a comprehensive guide that reflects the city's unique identity.
- **Identify top attractions:** Identify the top attractions in the city, such as landmarks, museums, parks, and restaurants. Include detailed descriptions and photos to help visitors plan their itinerary.
- **Provide practical information:** Provide practical information such as transportation options, weather, currency, and safety tips. This will help visitors navigate the city with ease and confidence.
- **Include off-the-beaten-path destinations:** Include lesser-known destinations that offer unique experiences and insights into the city's local culture. This will appeal to visitors who want to explore beyond the typical tourist hotspots.
- **Highlight events and festivals:** Highlight events and festivals that take place throughout the year, such as cultural celebrations, music festivals,

and sports events. This will give visitors a sense of the city's vibrant atmosphere and community spirit.

- Provide maps: Provide maps of the city that show the location of top attractions, transportation routes, and other key information. This will help visitors navigate the city more easily.
- Offer insider tips: Offer insider tips on the best places to eat, shop, and experience the city like a local. This will give visitors a more authentic and memorable experience.
- Use multimedia: Use multimedia such as photos, videos, and interactive maps to make the guide more engaging and visually appealing.

Overall, an effective tourism and city guide should be comprehensive, informative, and engaging, and should provide visitors with all the information they need to make the most of their visit to the city.

6.4. MAIN INFORMATION ABOUT SMART CITIES

This will definitely help the users for the purpose of saving their valuable time which can't be got back which is also economically viable. This system provides a registration form for all who wants to get the services. This can be categorized based on the type of users. They can search for a prominent places in the city without taking any help from personal guide.

Smart cities integrate technologies such as sensors, data analytics, automation, and the Internet of Things (IoT) to collect data, analyze it, and use it to improve city services and enhance citizens' quality of life. Smart city technologies can be used to optimize transportation systems, reduce energy consumption, improve waste management, enhance public safety, and provide better healthcare services, among other things.

7. SYSTEM DESIGN AND TESTING PLAN

7.1. INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

7.2. OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

The output form of an information system should accomplish one or more of

the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

7.3. SYSTEM TESTING

7.3.1. Test plan

Software testing is the process of evaluation a software item to detect differences between given input and expected output. Also to assess the feature of a software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words software testing is a verification and validation process.

7.3.2. Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

7.3.3. Validation

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

7.3.4. Basics of software testing

There are two basics of software testing: black box testing and white box testing.

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.

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.

Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

Integration Testing

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

Functional Testing

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

System Testing

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

Stress Testing

Stress testing is the testing to evaluate how system behaves under unfavorable conditions. Testing is conducted at beyond limits of the specifications. It falls under the class of black box testing.

Performance Testing

Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements. It falls under the class of black box testing.

Usability Testing

Usability testing is performed to the perspective of the client, to evaluate how the GUI is user-friendly? How easily can the client learn? After learning how to use, how proficiently can the client perform? How pleasing is it to use its design? This falls under the class of black box testing.

Acceptance Testing

Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected. It falls under the class of black box testing.

Regression Testing

Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing.

Testing part in integration

Testing is a critical aspect of integrating a smart city project. It ensures that the system is functioning correctly, meets the required performance standards, and delivers the intended benefits.

Testing is a crucial step in integrating a smart city project. By using various testing methods, decision-makers can identify and address any issues before the system is deployed, ensure that the system meets the required performance standards, and deliver the intended benefits to citizens.

8.SYSTEM DEVELOPMENT

8.1. SYSTEM ARCHITECTURE

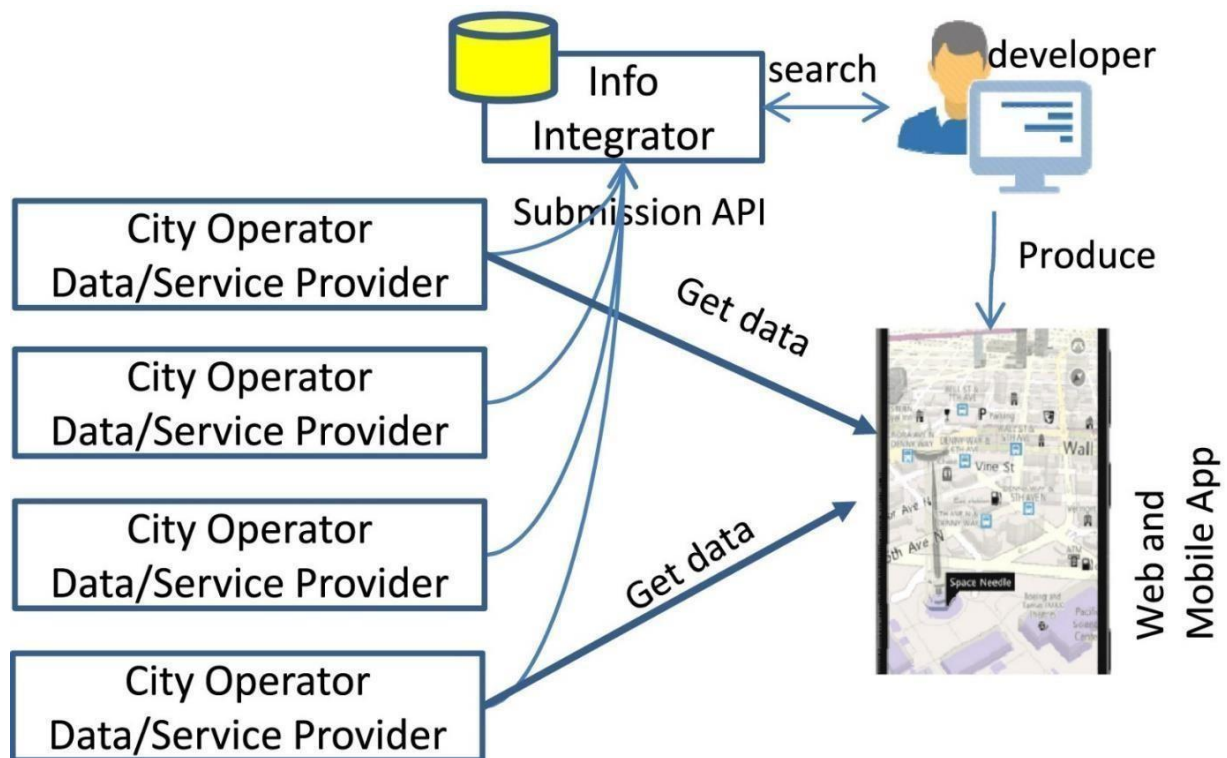


Figure 1

The city operators are combined to info integrator with API. The integrated content is get searched by developer and then produce the user interest by obtaining data from city operator.

8.2. DATA FLOW DIAGRAM

Data Flow Diagram (DFD) is a two-dimensional diagram that describes how data is processed and transmitted in a system. The graphical depiction recognizes each source of data and how it interacts with other data sources to reach a mutual output. In order to draft a data flow diagram one must

- Identify external inputs and outputs
- Determine how the inputs and outputs relate to each other

Level 0 DFD Diagram

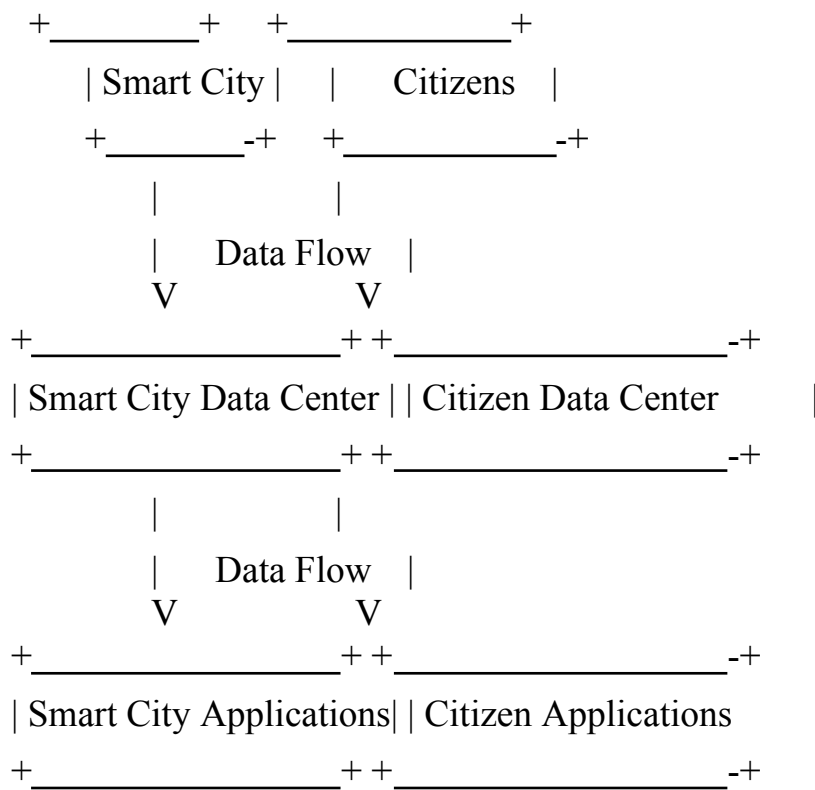
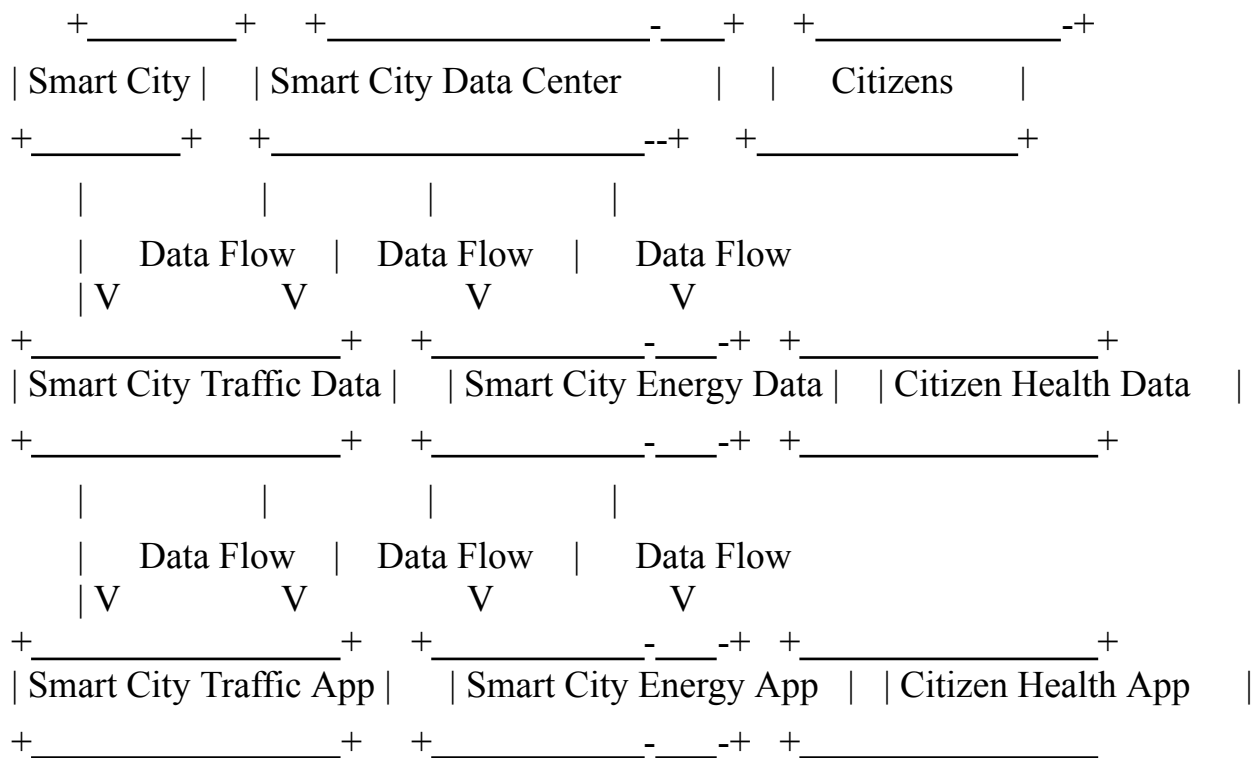


Figure 2

Level 1 DFD Diagram



+ Figure 3

Role of DFD:

It is a documentation support which is understood by both programmers and nonprogrammers. As DFD postulates only what processes are accomplished not how they are performed.

A physical DFD postulates where the data flows and who processes the data.

It permits analyst to isolate areas of interest in the organization and study them by examining the data that enter the process and viewing how they are altered when they leave.

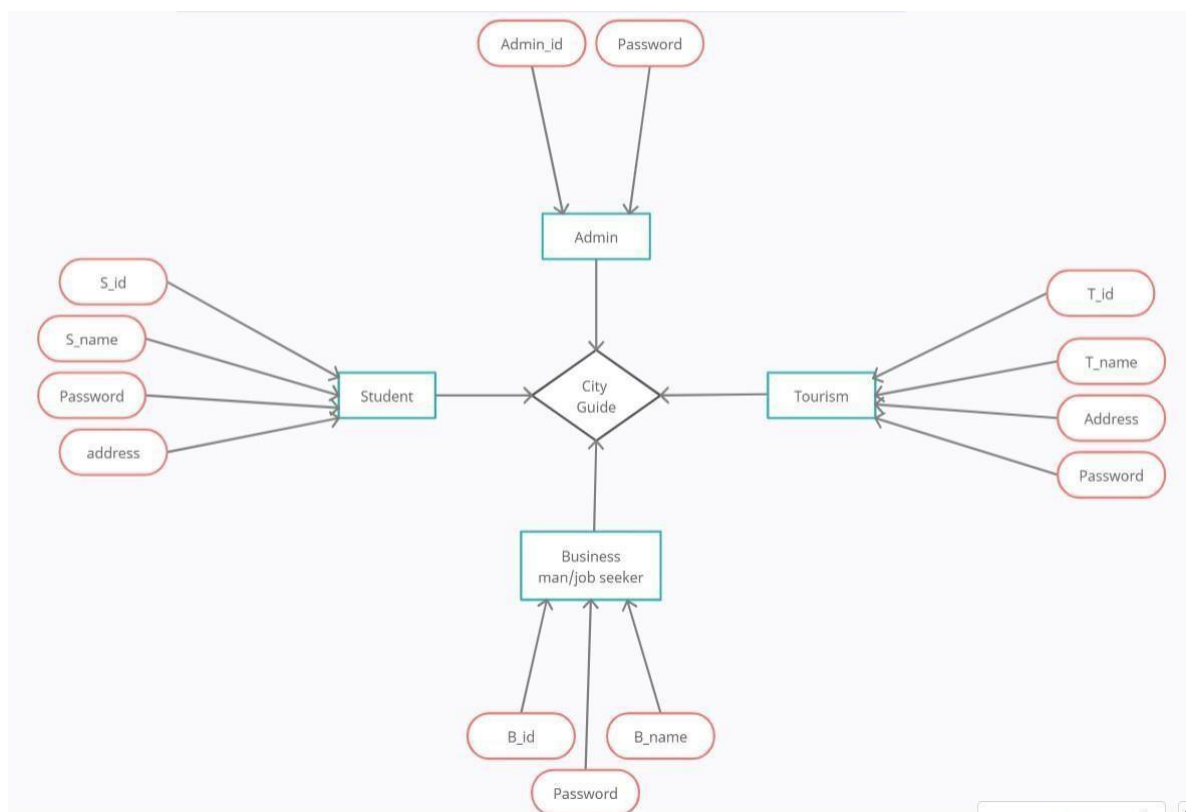


Figure 4

8.3. ER DIAGRAM

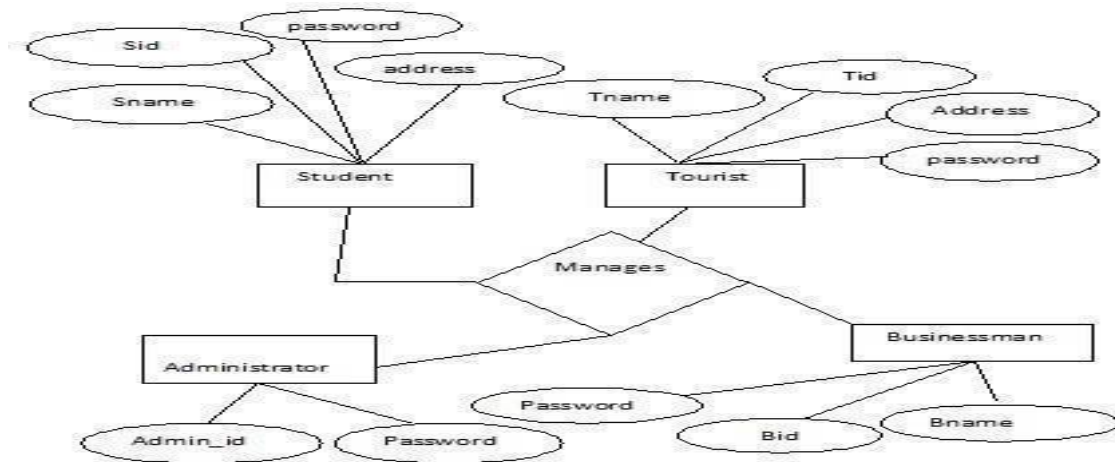


Figure 5

8.4. UML DIAGRAMS

UML is simply another graphical representation of a common semantic model. UML provides a comprehensive notation for the full lifecycle of object-oriented development.

ADVANTAGES

- To represent complete systems (instead of only the software portion) using object oriented concepts
- To establish an explicit coupling between concepts and executable code
- To take into account the scaling factors that are inherent to complex and critical systems
- To creating a modeling language usable by both humans and machines

UML defines several models for representing systems

The class model captures the static structure

The state model expresses the dynamic behavior of objects

The use case model describes the requirements of the user

The interaction model represents the scenarios and messages flows

The implementation model shows the work units

The deployment model provides details that pertain to
processallocation

8.4.1. USE CASE DIAGRAM

Use case diagrams overview the usage requirement for system. They are useful for presentations to management and/or project stakeholders, but for actual development you will find that use cases provide significantly more value because they describe “the meant” of the actual requirements. A use case describes a sequence of action that provides something of measurable value to an action and is drawn as a horizontal ellipse.

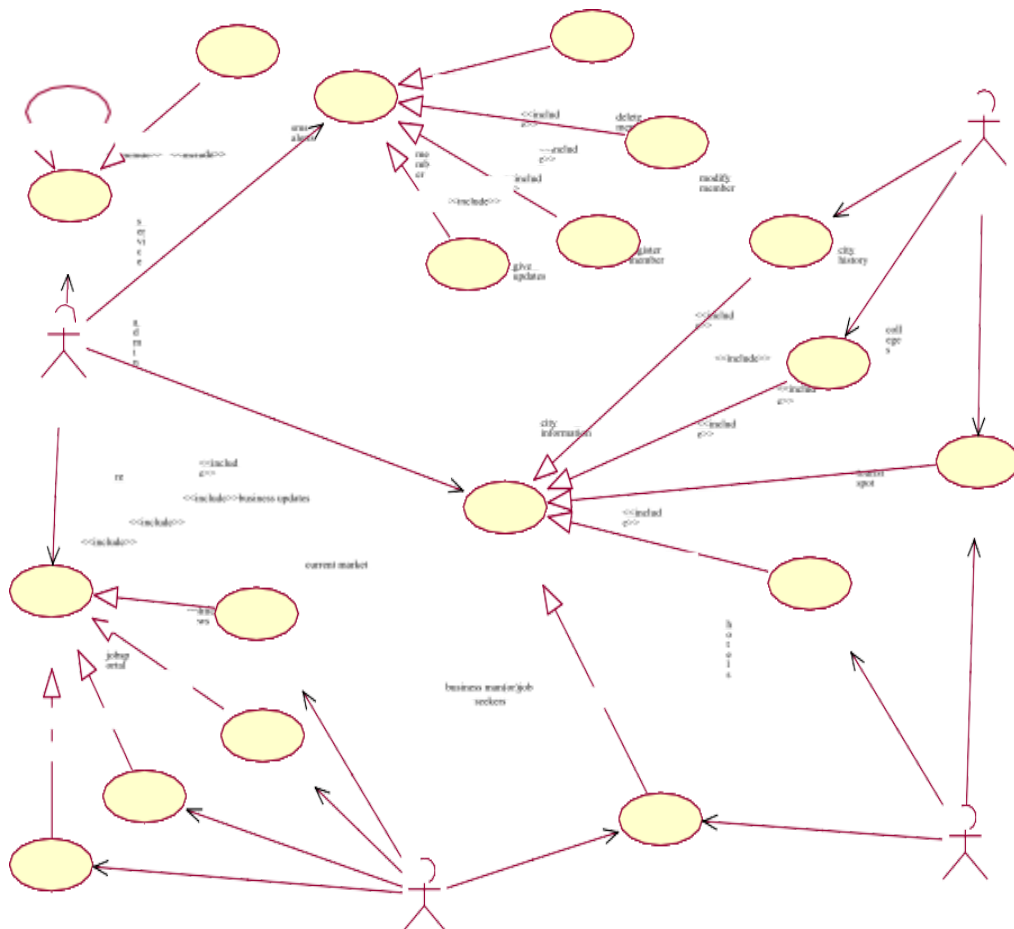


Figure 6

8.4.2. CLASS DIAGRAM

Class diagrams are the most common diagrams used in UML. Class diagram consists of classes, interfaces, associations, and collaboration. Class diagrams basically represent the object-oriented view of a system, which is static in nature. Class diagram represents the object orientation of a system.

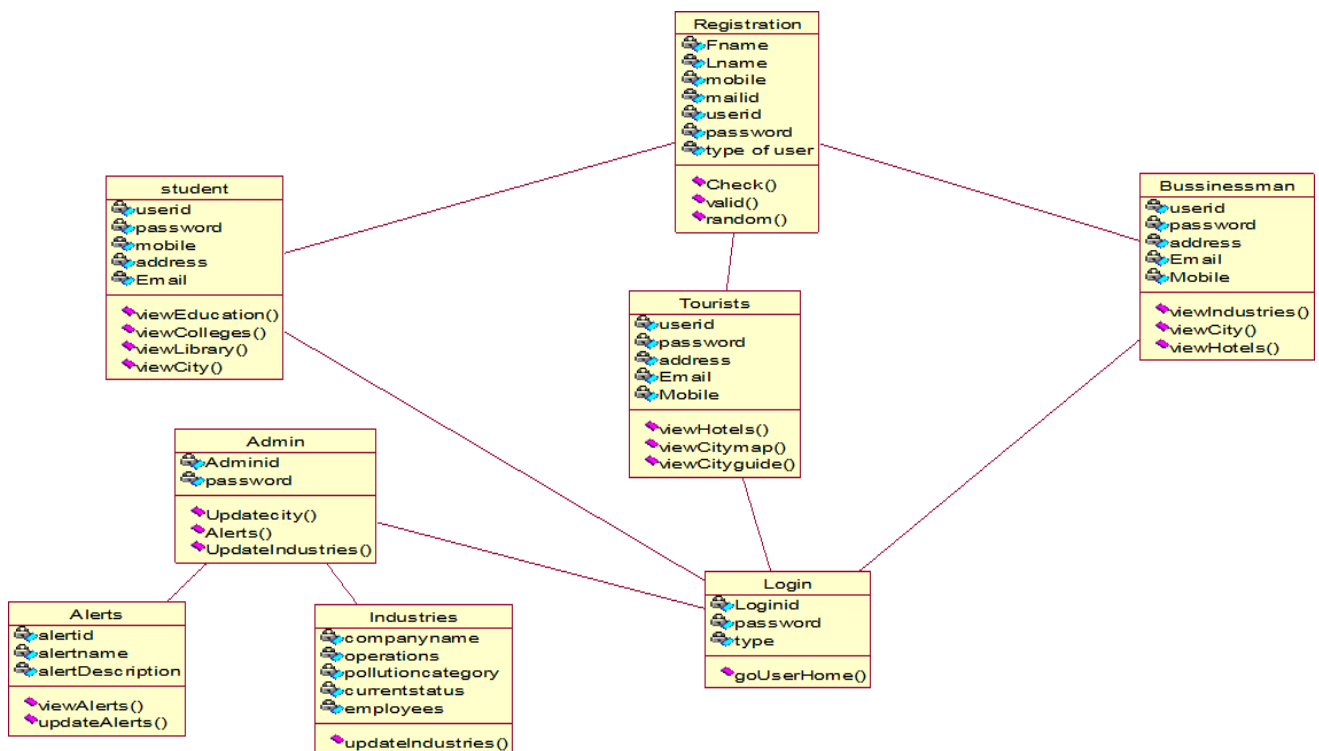


Figure 7

8.4.3. SEQUENCE DIAGRAM

Sequence diagram model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and commonly used for both analysis and design purpose. Sequence diagram are the most popular UML artifact for dynamic modeling, which focuses on identifying the behavior within your system.

Sequence Diagram of the User

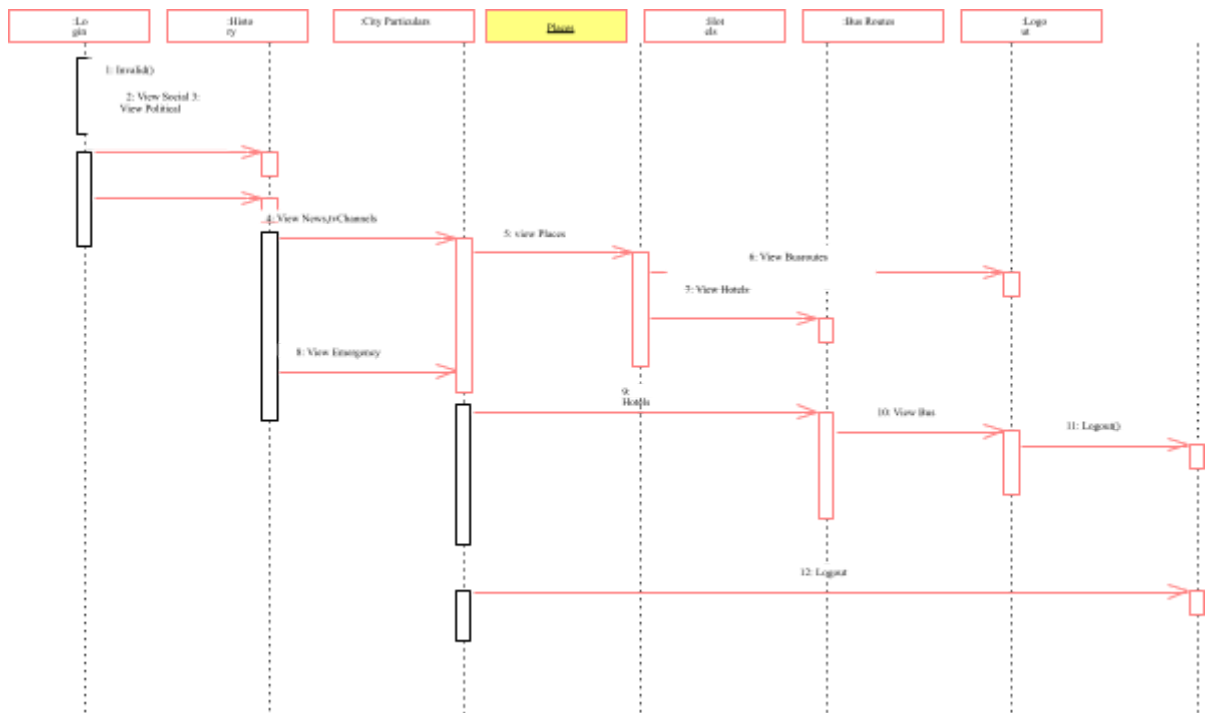


Figure 8

Admin Site Update Sequence Diagram

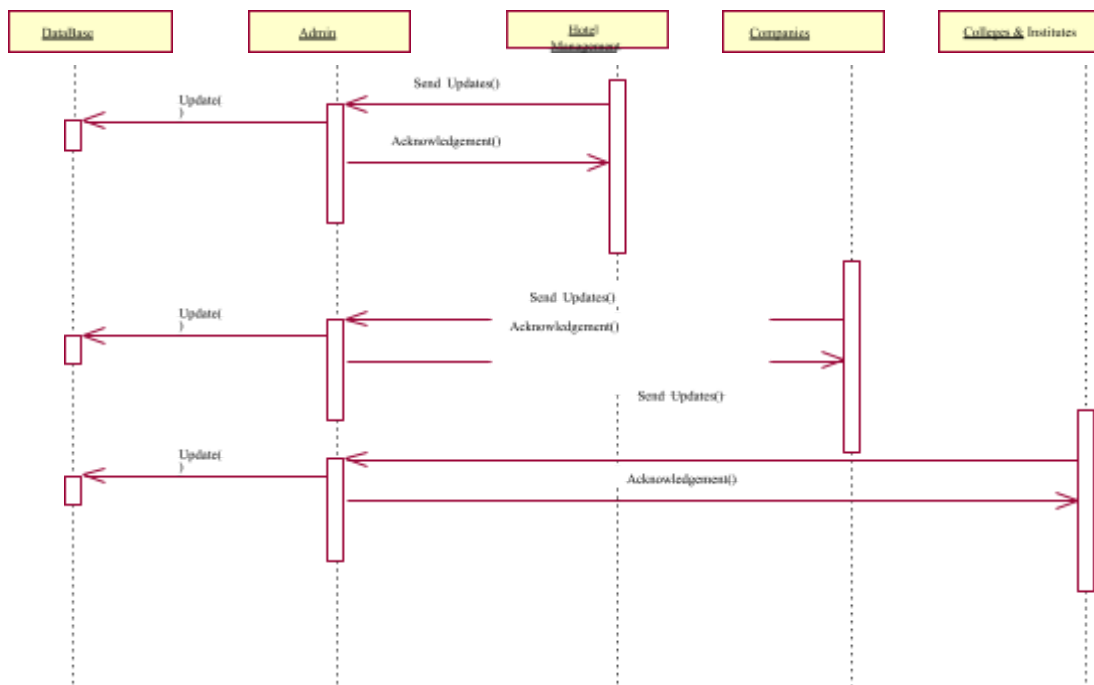


Figure 9

8.4.4. ACTIVITY DIAGRAM

Activity diagram are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. The activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. Activity diagram consist of Initial node, activity final node and activities in between.

User Activity Diagram

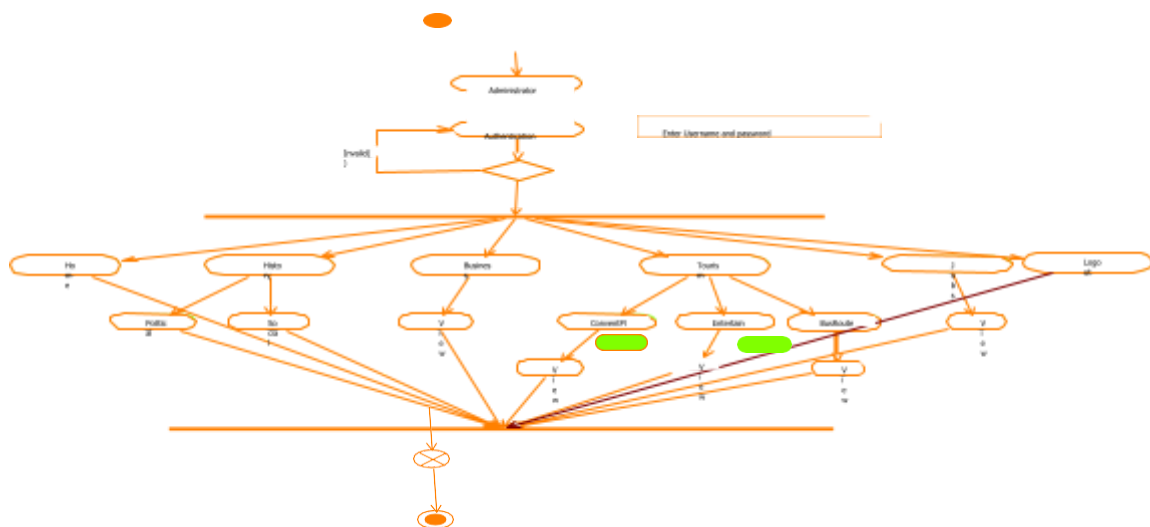


Figure 10

Administrator activity diagram

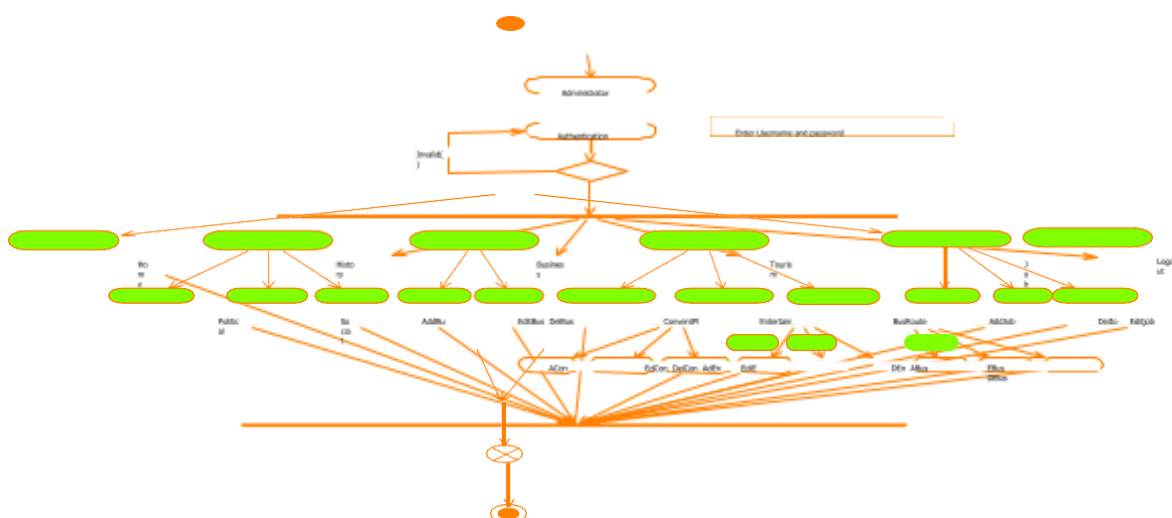


Figure 11

8.4.5. COLLABORATION DIAGRAM

Collaboration diagram is another form of interaction diagram. It represents the structural organization of a system and the messages sent/received. Structural organization consists of objects and links. The purpose of collaboration diagram is similar to sequence diagram.

Collaboration Diagram of the User

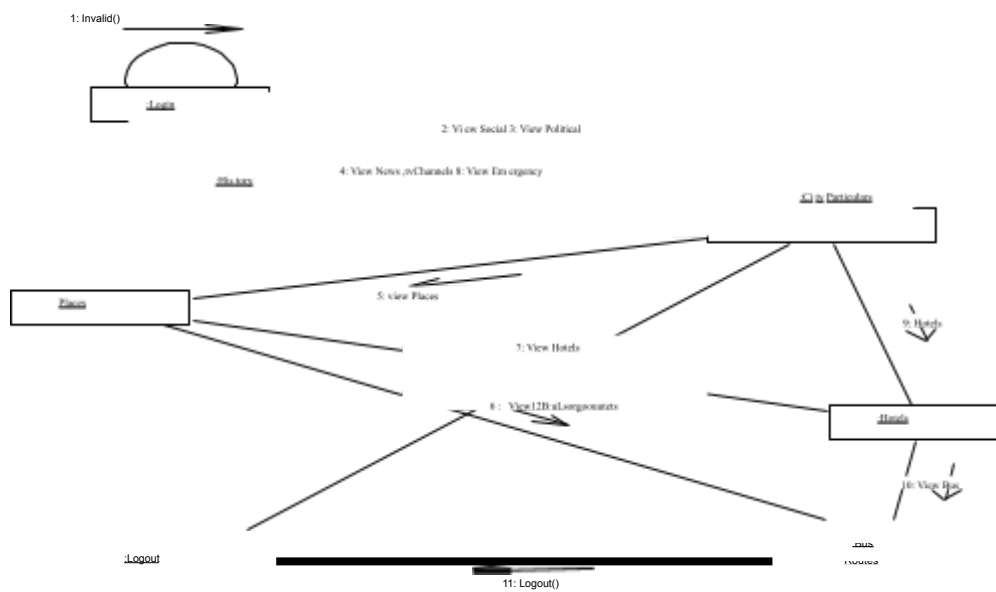


Figure 12

9. IMPLEMENTATION

9.2. SAMPLE CODE:

```
package

com.example.cityguide; import

android.content.Context;

import

androidx.test.platform.app.InstrumentationRegistry;

import

androidx.test.ext.junit.runners.AndroidJUnit4;

import org.junit.Test;

import org.junit.runner.RunWith;

import static org.junit.Assert.*;

/**

 * Instrumented test, which will execute on an Android device.

 *

 * @see <a href="http://d.android.com/tools/testing">Testing documentation</a>

 */

@RunWith(AndroidJUnit4.c

lass)

public class ExampleInstrumentedTest

{ @Test
```

```

public void useAppContext() {

// Context of the app under test.

Context                                appContext
= InstrumentationRegistry.getInstrumentation().getTargetContext();

assertEquals("com.example.cityguide", appContext.getPackageName());

}

}

```

```

<?xml                                version="1.0"                                encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.cityguide">
<uses-permission
android:name="android.permission.INTERNET"></uses- permission>
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE"
/>
<uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE"
/>
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"
/>
<uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<application
android:allowBackup="true"

```

android:icon="@drawable/icon"

android:label="@string/app_name"

android:roundIcon="@mipmap/ic_launcher_round"

```
android:supportsRtl="true"
android:theme="@style/Theme.CityGu
ide">
```

```
<meta-data
android:name="com.google.android.gms.ver
sion"
android:value="@integer/google_play_services_version" />
```

```
<meta-data
android:name="com.google.android.geo.API_KEY"
android:value="AIzaSyBfbv99FCA24Mw6i9iE9vfMGCbJNULfmrM"/>
```

```
<activity
android:name=".activity.SplashActivity"
android:theme="@style/Theme.AppCompat.NoActionBar">
```

```
<intent-filter>
```

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

```
<category                android:name="android.intent.category.LAUNCHER"                />
```

```
</intent-filter>
```

```
</activity>
```

```
<activity
android:name=".sign.SignInActivity"
android:theme="@style/Theme.AppCompat.Light.NoActionBar"/>
```

```
<activity
android:name=".sign.signupActivity"
android:theme="@style/Theme.AppCompat.Light.NoActionBar"/>
```

```
<activity
android:name=".sign.UserActivity"
android:theme="@style/Theme.AppCompat.Light.NoActionBar"/>
```

```
<activity  
    android:name=".activity.MainActivity"  
    android:theme="@style/AppTheme.NoActionBar"/>
```

```
<activity  
    android:name=".activity.categoryActivity"  
    android:theme="@style/AppTheme.NoActionBar"  
/>
```

```
<activity                                android:name=".activity.detailActivity"
```

```
android:theme="@style/AppTheme.NoActionBar"  
</>
```

```
<activity  
    android:name=".activity.viewActivity"  
    android:theme="@style/AppTheme.NoActionBar"/>
```

```
</application>
```

```
</manifest>
```

Category activity.java

```
package com.example.cityguide.activity;  
  
import android.content.Intent;  
  
import android.net.Uri;  
  
import android.os.Bundle;  
  
import android.util.Log;  
  
import  
  
    android.widget.Toast;  
  
  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import  
  
    androidx.recyclerview.widget.DefaultItemAnimator;  
  
import  
  
    androidx.recyclerview.widget.LinearLayoutManager;  
  
import androidx.recyclerview.widget.RecyclerView;
```



```
import com.android.volley.Request;
```

```
import com.android.volley.Response;

import com.android.volley.VolleyError;

import

com.android.volley.toolbox.JsonObjectRequest;

import com.example.cityguide.MySingleton;

import com.example.cityguide.R;

import

com.example.cityguide.adapters.categoryAdapter

; import

com.example.cityguide.modals.details_categories

;


import org.json.JSONArray;

import

org.json.JSONObject;


import java.util.ArrayList;


public class categoryActivity extends AppCompatActivity {

    RecyclerView recyclerView;

    String longitude, latitude;

    categoryAdapter adapter;
```

```
ArrayList<details_categories> arrayList = new
```

```
ArrayList<>(); String query;
```

```
String    apiKey="AlzaSyAxrj_oW36wimhP-0Y2kvQSkK2CG6mTmKc";
```

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
  
    setContentView(R.layout.category_recycle);  
    recyclerView = (RecyclerView) findViewById(R.id.recycle);  
    recyclerView.setLayoutManager(new LinearLayoutManager(this));  
    recyclerView.setHasFixedSize(true);  
    recyclerView.setItemAnimator(new DefaultItemAnimator());
```

```
    Intent i=getIntent();  
    query=i.getStringExtra("query");  
    fetchData();  
}
```

```
private void fetchData() {
```

```
    MainActivity a = new MainActivity();  
    latitude = a.getLatitude();  
    longitude = a.getLongitude();  
    String url = "https://maps.googleapis.com/maps/api/place/textsearch/json?";  
    String lat=a.getLatitude()+" "+a.getLongitude();  
    Uri baseUrl= Uri.parse(url);
```

```

Uri.Builder uriBuilder = baseUrl.buildUpon();

uriBuilder.appendQueryParameter("ll",lat);

uriBuilder.appendQueryParameter("query",query);

uriBuilder.appendQueryParameter("key",apiKey);

Log.d("urlll",uriBuilder.toString());

JsonObjectRequest jsonObjectRequest = new
JsonObjectRequest(Request.Method.GET, uriBuilder.toString(), null, new
Response.Listener<JSONObject>() {

@Override

public void onResponse(JSONObject response) {

try {

JSONArray

jsonArray=response.getJSONArray("results"); for(int

i=0;i<jsonArray.length();i++){

JSONObject jsonObject=jsonArray.getJSONObject(i);

String address="null";

if(jsonObject.has("formatted_address")) {

address = jsonObject.getString("formatted_address");

}

double lat=0,lon=0;

if(jsonObject.has("geometry")) {

JSONObject jsonObject1 = jsonObject.getJSONObject("geometry");

JSONObject jsonObject2 = jsonObject1.getJSONObject("location");

```

```
lat =  
jsonObject2.getDouble("lat"); lon  
= jsonObject2.getDouble("lng");  
}  
String resname=null;  
if(jsonObject.has("name"))  
{  
resname = jsonObject.getString("name");  
}  
String placeid=null;  
if(jsonObject.has("place_id")) {  
placeid = jsonObject.getString("place_id");  
}  
double rating=0;  
if(jsonObject.has("rating")  
) {  
rating = jsonObject.getDouble("rating");  
}  
String photoref="null";  
if(jsonObject.has("photos")) {  
JSONArray jsonArray1 = jsonObject.getJSONArray("photos");  
  
JSONObject jsonObject3 = jsonArray1.getJSONObject(0);
```

```
photoref = jsonObject3.getString("photo_reference");  
}
```

```

Log.d("S",resname+"/"+address+"/"+lat+"/"+lon+"/"+photoref+"/"+placeid+"/"+
+rating);

// fetchPhoto(resname,address,lat,lon,photoref,placeid,rating);

details_categories                                details=new
details_categories(resname,address,lat,lon,photoref,placeid,rating);

arrayList.add(details);

}

} catch (Exception e) {

Toast.makeText(getApplicationContext(),
e.getMessage()+"ss", Toast.LENGTH_SHORT).show();

}

adapter = new categoryAdapter(categoryActivity.this, arrayList);

recyclerView.setAdapter(adapter);

}

}, new Response.ErrorListener()

{ @Override

public void onErrorResponse(VolleyError error) {

Toast.makeText(getApplicationContext(),          "Failed:Check
Internet", Toast.LENGTH_SHORT).show();

}

});

```



```
MySingleton.getInstance(this).addToRequestQueue(jsonObjectRequest);  
  
}  
  
}
```

Detailactivity.java

```
package com.example.cityguide.activity;  
  
import android.os.Bundle;  
  
import  
  
android.view.LayoutInflater;  
  
import android.widget.ImageView;  
  
import android.widget.TextView;  
  
  
import androidx.annotation.NonNull;  
  
import  
  
androidx.appcompat.app.AppCompatActivity;  
  
import androidx.databinding.DataBindingUtil;  
  
import androidx.fragment.app.Fragment;  
  
import  
  
androidx.fragment.app.FragmentManager;  
  
import androidx.lifecycle.Lifecycle;  
  
import androidx.viewpager2.adapter.FragmentStateAdapter;
```

```
import com.bumptechnology.glide.Glide;
```

```
import com.example.cityguide.R;

import com.example.cityguide.adapters.categoryAdapter;

import

com.example.cityguide.databinding.ActivityDetailBinding;

import com.example.cityguide.fragments.fragmentInfo;

import com.example.cityguide.fragments.fragmentPhoto;

import com.example.cityguide.fragments.fragmentReview;

import

com.google.android.material.appbar.CollapsingToolbarLayout;

import com.google.android.material.tabs.TabLayoutMediator;


import

java.util.ArrayList;

import java.util.List;


public class detailActivity extends AppCompatActivity {

    ImageView image;

    private ActivityDetailBinding binding;

    private detailActivity activity;

    private viewPagerAdapter adapter;

    CollapsingToolbarLayout text;

    protected void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);
```

```
binding=DataBindingUtil.setContent View(this,R.layout.activity_detail );
```

```
activity=this;

image=findViewById(R.id.image);

text=findViewById(R.id.toolbar);

categoryAdapter a=new
categoryAdapter();

Glide.with(getApplicationContext()).load(a.getphotoiurl()).into(image);

text.setTitle(a.getpplacement());


initView();

}


private void initView() {

adapter=new
viewPagerAdapter(activity.getSupportFragmentManager(),activity.getLife
cycle ());

adapter.addFragement(new fragmentInfo(),"INFO" );

adapter.addFragement(new fragmentPhoto(),"PHOTO");

adapter.addFragement(new fragmentReview(),"REVIEWS" );

binding.viewPager.setAdapter(adapter);

binding.viewPager.setOffscreenPageLimit(1);
```

new

```
TabLayoutMediator(binding.tabLayout, binding.viewP  
ager, (tab, position) -> {  
tab.setText(adapter.fragmentTitleList.get(position));  
}).attach();
```

```
for(int i=0;i<binding.tabLayout.getTabCount();i++){
```

```
TextView tv = (TextView) LayoutInflater.from(activity)  
.inflate(R.layout.custom_tab, null);
```

```
binding.tabLayout.getTabAt(i).setCustomView(tv);  
  
}  
  
}
```

```
class viewPagerAdapter extends FragmentStateAdapter {  
private final List<Fragment> fragmentList=new ArrayList<>();  
private final List<String> fragmentTitleList=new  
ArrayList<>();
```

```
public viewPagerAdapter(@NonNull FragmentManager
```

```
fragmentManager, @NonNull Lifecycle lifecycle) {
```

```
super(fragmentManager, lifecycle);
```

```
}
```

```
public void addFragement(Fragment fragment,String title){
```

```
fragmentList.add(fragment);
```

```
fragmentTitleList.add(title);
```

```
}
```

```
@NonNull
```

```
@Overrid
```

```
e
```

```
public Fragment createFragment(int position)
```

```
{ return fragmentList.get(position);
```

```
}
```

```
@Override
```

```
public int getItemCount()
```

```
{ return
```

```
fragmentList.size();
```

```
}}}
```

Mainactivity.java


```
package com.example.cityguide.activity;
```

```
import android.Manifest;

import

android.app.AlertDialog;

import

android.content.DialogInterface;

import android.content.Intent;

import

android.content.pm.PackageManager;

import android.location.Location;

import android.os.Bundle;

import android.util.Log;

import

android.view.MenuItem;

import android.view.View;

import

android.view.WindowManager;

import android.widget.ImageView;

import android.widget.TextView;


import androidx.annotation.NonNull;

import

androidx.appcompat.app.ActionBarDrawerToggle;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

```
import androidx.core.app.ActivityCompat;
```

```
import androidx.core.view.GravityCompat;
```

```
import
```

```
androidx.drawerlayout.widget.DrawerLayout;
```

```
import androidx.fragment.app.Fragment;
```

```
import androidx.fragment.app.FragmentTransaction;

import com.bumptech.glide.Glide;

import com.example.cityguide.R;
import com.example.cityguide.circle;

import
com.example.cityguide.fragments.fragmentCategory;
import com.example.cityguide.fragments.fragmentAbout;
import com.example.cityguide.fragments.fragmentMap;
import com.example.cityguide.fragments.fragmentHome;
import
com.example.cityguide.fragments.fragmentLocation;
import com.example.cityguide.fragments.fragmentProfile;
import com.example.cityguide.fragments.fragmentSaved;
import com.example.cityguide.modals.Users;
import com.example.cityguide.sign.SignInActivity;
import
com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
```

```
import
```

```
com.google.android.material.navigation.NavigationView;
```

```
import com.google.firebase.auth.FirebaseAuth;
```

```
import com.google.firebase.auth.FirebaseUser;
```

```

import com.google.firebase.database.DataSnapshot;

import com.google.firebase.database.DatabaseError;

import com.google.firebase.database.DatabaseReference;

import com.google.firebase.database.FirebaseDatabase;

import com.google.firebase.database.ValueEventListener;


public class MainActivity extends AppCompatActivity
implements NavigationView.OnNavigationItemSelectedListener {

    ImageView imageview;

    TextView name;

    DatabaseReference

    rootref; FirebaseAuth

    mAuth; FirebaseAuth

    firebaseUser;

    public static Location currentLocation;

    FusedLocationProviderClient

    fusedLocationProviderClient; private static final int

    REQUEST_CODE = 101; NavigationView

    navigationView; androidx.appcompat.widget.Toolbar

    toolbar;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

```

```
super.onCreate(savedInstanceState);
```

```
getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,  
WindowManager.LayoutParams.FLAG_FULLSCREEN);
```

```
setContentView(R.layout.activity_main
```

```
); mAuth = FirebaseAuth.getInstance();
```

```
toolbar = (androidx.appcompat.widget.Toolbar) findViewById(R.id.toolbar);
```

```
toolbar.setTitle("MY PLACE");
```

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

```
View hView = navigationView.getHeaderView(0);
```

```
imageView = (ImageView) hView.findViewById(R.id.imageView);
```

```
name = (TextView) hView.findViewById(R.id.menu_header_name);
```

```
fusedLocationProviderClient =
```

```
LocationServices.getFusedLocationProviderClient(this);
```

```
DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer);
```

```
ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
```

```
this, drawer, toolbar,
```

```
R.string.navigation_drawer_open, R.string.navigation_drawer_close);
```

```
drawer.setDrawerListener(toggle);
```

```
toggle.syncState();
```



```
navigationView.setNavigationItemSelectedListener(this);
```

```
//showing default fragment
```

```
displaySelectedFragment(R.id.nav_ho
```

```
me); getRef();
```

```
currentlocation();
```

```
}
```

```
@Override
```

```
public void onBackPressed() {
```

```
DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer);
```

```
if (drawer.isDrawerOpen(GravityCompat.START)) {
```

```
drawer.closeDrawer(GravityCompat.START);
```

```
} else {
```

```
super.onBackPressed();
```

```
}
```

```
}
```

```
void getRef() {
```

```
firebaseUser = FirebaseAuth.getInstance().getCurrentUser();
```

```
rootref =
```

```
FirebaseDatabase.getInstance().getReference("Users");
```

```
rootref.addValueEventListener(new ValueEventListener() {
```

@Override

```
public void onDataChange(@NonNull DataSnapshot
snapshot) { for (DataSnapshot dataSnapshot :
snapshot.getChildren()) { Users user =
dataSnapshot.getValue(Users.class);

Glide.with(getApplicationContext()).load(user.getImageUrl().toString()).tr
ansform(new circle(getApplicationContext())).into(imageview);

name.setText("Dinesh");

Log.d("name", firebaseUser.getEmail());

}

}
```

@Override

```
public void onCancelled(@NonNull DatabaseError error) {

}

});

}
```

@Override

```
public boolean onNavigationItemSelected(@NonNull MenuItem item) {
```

```
// item id is being passed into the method here
```

```
displaySelectedFragment(item.getItemId());
```

```
DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer);
```

```
drawer.closeDrawer(GravityCompat.START);
```

```
return true;
```

```
}
```

```
public void setScreenTitle(int item_id)
```

```
{ String title = "";
```

```
switch (item_id) {
```

```
case R.id.nav_home:
```

```
title = "Home";
```

```
break;
```

```
case
```

```
R.id.nav_category: title
```

```
= "Category"; break;
```

case

R.id.nav_Location: title

= "Location"; break;

case R.id.nav_save:

title = "Saved

Location"; break;

case

R.id.nav_profile:

title = "Profile";

break;

case R.id.nav_logout:

title = "Three";

break;

case R.id.nav_map:

title = "MAP";

break;

case R.id.nav_about:

title = "About";

```
break;
```

```
}
```

```
toolbar.setTitle(title);
```

```
}
```

```
public void displaySelectedFragment(int item_id) {
```

```
Fragment fragment = null;
```

```
switch (item_id) {
```

```
case R.id.nav_home:
```

```
fragment = new fragmentHome();
```

```
navigationView.getMenu().getItem(0).setChecked
```

```
d(true); break;
```

```
case R.id.nav_category:
```

```
fragment = new fragmentCategory();
```

```
navigationView.getMenu().getItem(1).setChecked
```

```
d(true); break;
```

```
case R.id.nav_Location:
```

```
fragment = new fragmentLocation();
```

```
navigationView.getMenu().getItem(2).setChecked  
d(true); break;
```

```
case R.id.nav_save:
```

```
fragment = new fragmentSaved();
```

```
navigationView.getMenu().getItem(3).setChecked  
d(true); break;
```

```
case R.id.nav_profile:
```

```
fragment = new fragmentProfile();
```

```
navigationView.getMenu().getItem(4).setChecked  
d(true); break;
```

```
case R.id.nav_logout:
```

```
logOut();
```

```
navigationView.getMenu().getItem(5).setChecked  
d(true); break;
```

```
case R.id.nav_map:
```

```
fragment = new fragmentMap();
```

```
break;
```

```
case R.id.nav_about:
```

```
fragment = new fragmentAbout();
```

```
break;
```

```
}
```

```
if (fragment != null) {
```

```
FragmentTransaction ft = getSupportFragmentManager().beginTransaction();
```

```
//this is where the id of the FrameLayout is being mentioned. Hence the fragment  
would be loaded into the framelayout
```

```
ft.replace(R.id.container, fragment);
```

```
ft.commit();
```

```
}
```

```
/** setting title to the screen
```

```
*/ setTitle(item_id);
```

```
}
```

```
private void logOut() {  
  
    AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);  
  
    builder.setMessage("Do you want to Log Out ?");  
  
    builder.setTitle("LOG OUT");  
  
  
    builder.setCancelable(false);  
  
    builder.setPositiveButton("Yes", new DialogInterface.OnClickListener() {  
  
        @Override  
  
        public void onClick(DialogInterface dialog, int which) {  
  
            outofScreen();  
  
        }  
  
    });  
  
    builder.setNegativeButton("No", new DialogInterface.OnClickListener() {  
  
        @Override  
  
        public void onClick(DialogInterface dialog,  
  
            int which) {  
  
            dialog.cancel();  
  
        }  
  
    }  
}
```



```
});
```

```
AlertDialog alertDialog = builder.create();
```

```
alertDialog.show();
```

```
}
```

```
private void outofScreen() {
```

```
    FirebaseAuth.getInstance().signOut();
```

```
    Intent intent = new Intent(MainActivity.this, SignInActivity.class);
```

```
    startActivity(intent);
```

```
}
```

```
void currentlocation(){
```

```
    if
```

```
    (ActivityCompat.checkSelfPermission(this,
```

```
    Manifest.permission.ACCESS_FINE_LOCATION)
```

```
    !=
```

```
    PackageManager.PERMISSION_GRANTED
```

```
    && ActivityCompat.checkSelfPermission(
```

```
    this,      Manifest.permission.ACCESS_COARSE_LOCATION)
```

```
    != PackageManager.PERMISSION_GRANTED) {
```

```
ActivityCompat.requestPermissions(this,  
new String[] {Manifest.permission.ACCESS_FINE_LOCATION},  
REQUEST_CODE);
```

```
return;
```

```
}
```

```
Task<Location> task = fusedLocationProviderClient.getLastLocation();
```

```
task.addOnSuccessListener(new OnSuccessListener<Location>() {
```

```
@Override
```

```
public void onSuccess(Location location) {
```

```
if (location != null) {
```

```
currentLocation = location;
```

```
Log.d("lan", String.valueOf(currentLocation.getLatitude()));
```

```
Log.d("lot", String.valueOf(currentLocation.getLongitude()));
```

```
}
```

```
}
```

```
});
```

```
}
```

```
@Override
```

```

public void onRequestPermissionsResult ( int requestCode, @NonNull String[]
permissions,

@NonNull int[] grantResults){

super.onRequestPermissionsResult(requestCode, permissions,

grantResults); switch (requestCode) {

case REQUEST_CODE:

if      (grantResults.length      >      0      &&      grantResults[0]      ==

PackageManager.PERMISSION_GRANTED) {

currentlocation();

}

break;


}

}

}

public String getLatitude(){

return String.valueOf(currentLocation.getLatitude());

}

public String getLongitude(){

return  String.valueOf(currentLocation.getLongitude());

```

```
}  
  
public Location  
getLoction(){ return  
currentLocation;  
}}
```

Splashactivity.java

```
import  
  
android.content.Intent;  
  
import android.os.Bundle;  
  
import android.os.Handler;  
  
import android.view.WindowManager;  
  
  
import androidx.appcompat.app.AppCompatActivity;  
  
  
import com.example.cityguide.R;  
  
import  
  
com.example.cityguide.sign.SignInActivity;  
  
import com.google.firebase.auth.FirebaseAuth;  
  
  
public class SplashActivity extends AppCompatActivity {  
  
  
private static int SPLASH_SCREEN_TIME_OUT=2000;  
  
FirebaseAuth mAuth;
```

@Override

protected void onCreate(Bundle savedInstanceState) {

```
super.onCreate(savedInstanceState);
```

```
getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,  
WindowManager.LayoutParams.FLAG_FULLSCREEN);
```

```
setContentView(R.layout.splash_screen);
```

```
mAuth= FirebaseAuth.getInstance();
```

```
new Handler().postDelayed(new Runnable()
```

```
{ @Override
```

```
public void run() {
```

```
if(mAuth.getCurrentUser() == null)
```

```
{
```

```
Intent i = new
```

```
Intent(SplashActivity.this,SignInActivity.class);
```

```
startActivity(i);
```

```
}else {
```

```
Intent intent=new Intent(SplashActivity.this,MainActivity.class);
```

```
startActivity(intent);
```

```
}
```

```
finish();
```

```
}
```

```
}, SPLASH_SCREEN_TIME_OUT);
```

```
}
```

```
}
```

Viewactivity.java

```
package com.example.cityguide.activity;
```

```
import
```

```
android.content.Intent;
```

```
import android.os.Bundle;
```

```
import android.util.Log;
```

```
import
```

```
android.view.WindowManager;
```

```
import
```

```
android.widget.ImageView;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

```
import androidx.databinding.DataBindingUtil;
```

```
import com.bumptech.glide.Glide;
```

```
import com.example.cityguide.R;
```

```
import com.example.cityguide.adapters.PostAdapter;
```

```
import
```

```
com.example.cityguide.adapters.categoryAdapter;
```

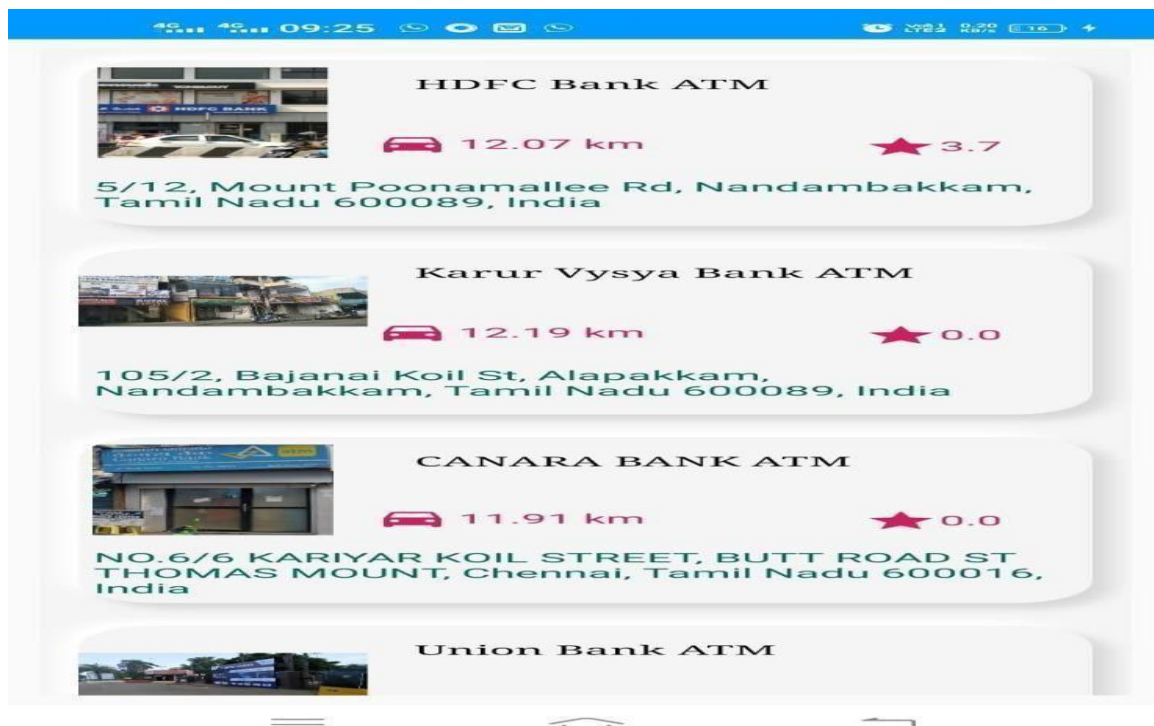
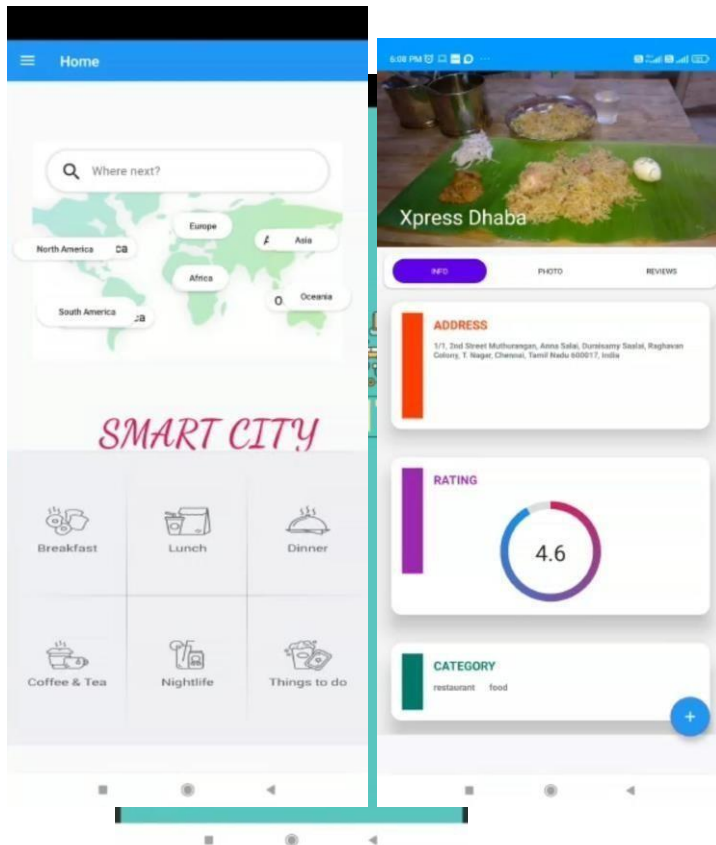


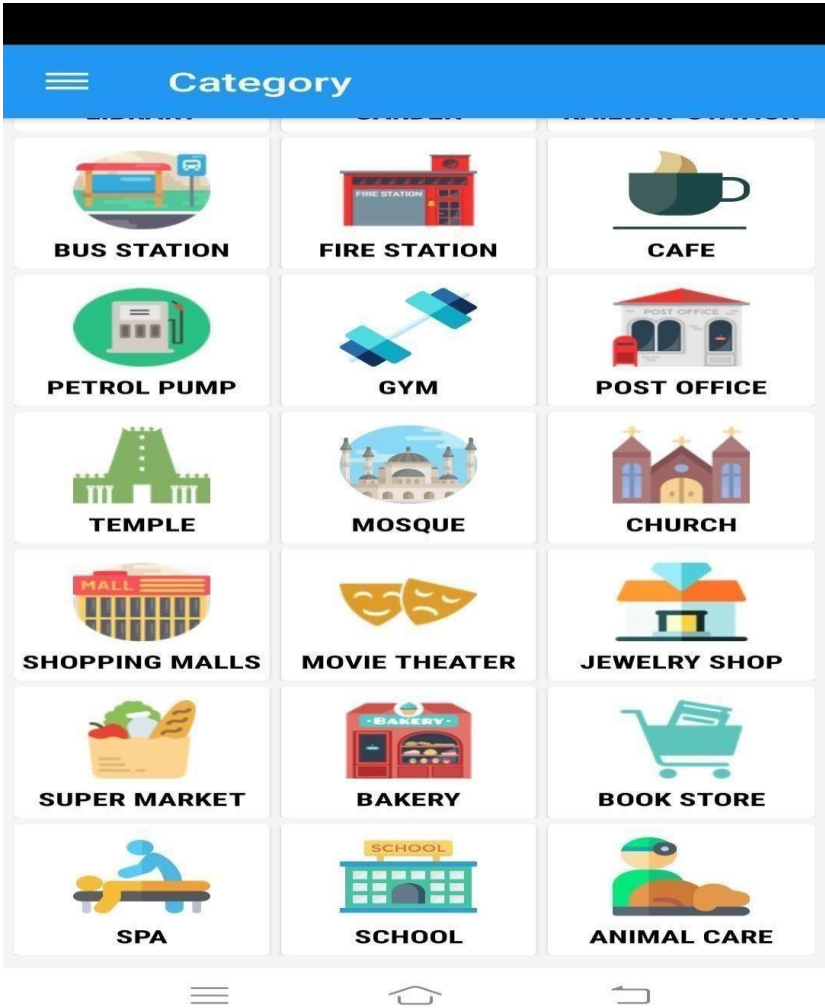
```
public class viewActivity extends AppCompatActivity {
```

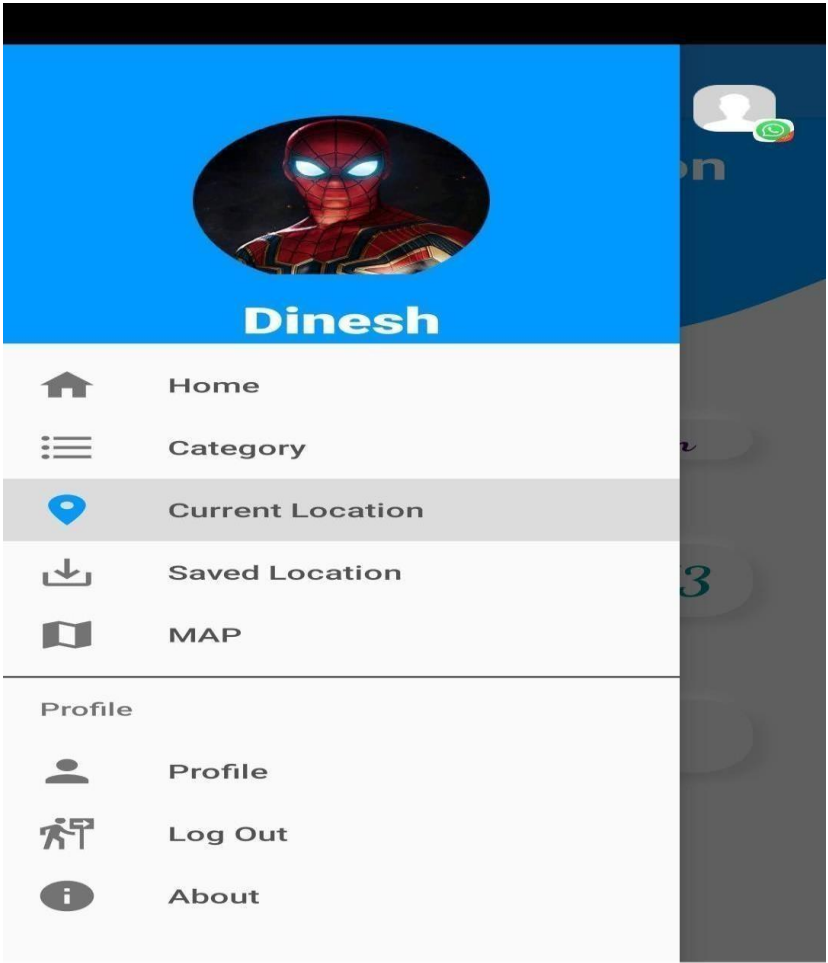
```
    ImageView viewphoto;
```

```
protected void onCreate(Bundle savedInstanceState) {  
  
    super.onCreate(savedInstanceState);  
  
    setContentView(R.layout.activity_viewphoto);  
  
  
    viewphoto=findViewById(R.id.i2);  
  
    PostAdapter a=new PostAdapter();  
  
    String url=a.getUrl();  
  
  
    Glide.with(getApplicationContext()).load(url).into(viewphoto);  
  
}  
  
}
```

9.2.OUTPUT SCREENSHOTS







10. CONCLUSION AND FUTURE

ENHANCEMENT CONCLUSION:

This Project in Java provides info regarding the various aspects of city such as tourism, institutes, industry, geographical maps, ATM locations, etc. The implementation of this project solves most of the problems a new visitor faces while coming to a new city such as: path finding, hotel searching, ticket booking, and more.

FUTURE SCOPE:

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 .

REFERENCES

1. Smart Cities, International Electrotechnical Commission. Available online: <https://www.iec.ch/smartcities/introduction.htm> (accessed on 7 August 2020).
2. Gandy, O.H., Jr.; Nemorin, S. Toward a political economy of nudge: Smart city variations. *Inf. Commun. Soc.* 2019, 22, 2112–2126. [CrossRef]
3. Sadowski, J.; Pasquale, F.A. The spectrum of control: A social theory of the smart city. *First Monday* 2015, 20. [CrossRef]
4. Inclusive Smart Cities: A European Manifesto on Citizen Engagement. Available online: https://eu-smartcities.eu/sites/default/files/2017-09/EIP-SCC%20Manifesto%20on%20Citizen%20Engagement%20%26%20Inclusive%20Smart%20Cities_0.pdf (accessed on 7 August 2020).
5. Ageing and Life-Course. Available online: <https://www.who.int/ageing/projects/age-friendly-environments/en> (accessed on 7 August 2020).
6. Van Zoonen, L. Privacy concerns in smart cities. *Gov. Inf. Q.* 2016, 33, 472–480. [CrossRef]
7. Lai, C.S.; Lai, L.L.; Lai, Q.H. *Smart Grids and Big Data Analytics for Smart Cities*, 1st ed.; Springer International Publishing: New York, NY, USA, 2020.
8. Camero, A.; Alba, E. Smart City and information technology: A review. *Cities* 2019, 93, 84–94.
9. Caird, S.P.; Hallett, S.H. Towards evaluation design for smart city development. *J. Urban Des.* 2019, 24, 188–209.

10. ISO/TR 37150:2014(en) Smart Community Infrastructures—Review of Existing Activities Relevant to Metrics.

Available online: <https://www.iso.org/obp/ui/#iso:std:iso:tr:37150:ed-1:v1:en> (accessed on 7 August 2020).

11. ISO/TS 37151:2015 Smart Community Infrastructures—Principles and Requirements for Performance

Metrics. Available online: <https://www.iso.org/standard/61057.html> (accessed on 7 August 2020).

12. Citykeys. Available online: <http://citykeys-project.eu> (accessed on 7 August 2020).

13. Hasija, S.; Shen, Z.-J.M.; Teo, C.-P. Smart city operations: Modeling challenges and opportunities. Manuf.

Serv. Oper. Manag. 2020, 22, 203–213.

14. Anthopoulos, L. Smart City Emergence: Cases From Around the World; Elsevier: Amsterdam, The Netherlands, 2019.

15. van Winden, W.; van den Buuse, D. Smart city pilot projects: Exploring the dimensions and conditions of

scaling up. J. Urban Technol. 2017, 24, 51–72. [CrossRef]