DECISION BASED RECCOMENDATION SYSTEM FOR SMART CITY

Submitted in partial fulfillment of the

requirements for the award of

Bachelor of Engineering degree in Computer Science and Engineering

By

NIRANJAN N (Reg.No - 39110702)

PRASATH S (Reg.No – 39110793)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC | 12B Status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI - 600119

APRIL- 2023

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **NIRANJAN N (39110702) and PRASATH S (39110793)** who carried out the Project Phase-2 entitled **"DECISION BASED RECOMMENDATION SYSTEM FOR SMART CITY"** under my supervision from JANUARY 2023 to APRIL 2023

Internal Guide

Ms. D.DEEPA, M.E., (Ph.D.)

Head of the Department
Dr. L. LAKSHMANAN, M.E., Ph.D.

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF CSE

Submitted for Viva voce Examination held on 20.04.2023

Internal Examiner **External Examiner**

DECLARATION

I, **NIRANJAN N (Reg.no - 39110702),** hereby declare that the Project Phase-2 Report entitled ”**Decision based recommendation system for smart city”** done by me under the guidance of **Ms. D.DEEPA, M.E., (Ph.D)** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

Niranjan

DATE: 20.04.2023 **NIRANJAN N**

PLACE: CHENNAI **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D**, **Professor & Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D., Professor & Head,** Department of Computer Science and Engineering for providing me with necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms. D. DEEPA,M.E., (Ph.D)**, Department of Computer Science and Engineering, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

The web-based product described is a comprehensive city portal that serves as a one-stop destination for all visitors looking for information related to a particular city. It offers a range of services such as hotel booking, ticket booking, transportation facilities, business-related information, marketing details, city news, and shopping details. The portal facilitates communication between users, experts, and the general public through various means such as chat, polls, and mail. This helps users save valuable time, which is economically viable. The system also provides a registration form for users who wish to avail of its services. The registration forms are categorized based on the type of user, such as students, businessmen, tourists, and jobseekers. For example, students can use the portal to download study materials and find information about coaching centers and college institutions. This ensures that user information is protected, and users can safely access the services provided by the portal. Overall, this city portal offers a range of services that cater to the needs of different types of users. It provides a convenient and efficient way to access information related to a particular city, which can help visitors save time and effort.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The project titled "A COMPLETE CITY GUIDE USING DATABASE" is a web-based platform that aims to provide a comprehensive city guide for users who are visiting a particular city. This platform contains all the essential information about the city, including places to be visited, site maps, route maps, business environment, job portal, information about organizations that provide transport and hospitality services, and the complete history of the city.

The platform is designed to be user-friendly, and anyone with general knowledge about the internet can use it. Initially, all users will be considered anonymous users. However, if a user needs any service, they will have to register on the platform. Once registered, the user will be treated as a registered user and will have access to a wide range of services offered by the platform.

The platform can be accessed by an unlimited number of users, and 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 on the platform with limited services and provide extra services to registered users. The platform also tracks all transaction details of the customer and confirms the end user's identity to ensure that only authorized users receive support.

The platform also maintains the history of each customer and their related information. This information can be used to provide customized services to the customer based on their previous interactions with the platform. The platform's database is designed to be scalable, ensuring that it can handle a large number of users and data without compromising the platform's performance.

CHAPTER 2

LITERATURE SURVEY

Li et al. (2019) Privacy and Security of Mobile Devices and Services. The study presented three novel approaches for satisfying the required privacy of mobile devices within smart cities. The first approach utilized the concept of foggy dummies to protect the privacy of the user; the second incorporated a blind third party where a trust relationship is developed to protect the user from the server provider; the third approach used the concept of a double foggy cache to solve the trust issue between peers with a traditional cooperation approach. Privacy-preserving authentication (PPA) protocols for mobile services have emerged as a promising cryptographic approach to provide authentication and privacy protection features for smart cities. The research presented in analyzed the PPA protocol suitability for mobile services within a typical mobile service application in a smart city context.

Abosaq et al (2019) Smart City Infrastructure research have argued that privacy can be easily compromised due to the high levels of interaction between people, devices and sensors, thus highlighting the need for this data to be fully protected A number of articles focused on smart city infrastructure and ways to overcome security and privacy issues within smart cities

Ainane et al. (2018); Smart City Infrastructure research have argued that privacy can be easily compromised due to the high levels of interaction between people, devices and sensors, thus highlighting the need for this data to be fully protected A number of articles focused on smart city infrastructure and ways to overcome security and privacy issues within smart cities

Chatterjee et al. (2017) Smart City Infrastructure research Studies have shown that the abundance of interactions between people, devices, and sensors in smart cities can pose a significant threat to privacy. As a result, it is crucial to ensure that the data collected is adequately protected. Multiple articles have focused on addressing security and privacy concerns in smart city infrastructure. They suggest using encryption, access controls, and privacy-enhancing technologies like differential privacy and anonymization to safeguard personal information. Other measures such as network isolation, intrusion detection systems, and regular vulnerability assessments can also help strengthen security.

Ferraz et al 2014 Smart City Infrastructure . research paper is based on Smart cities to use interconnected systems, devices, and sensors to gather and analyze data to improve quality of life, efficiency, and sustainability. However, this increase in data collection raises concerns about privacy and security. To address these concerns, multiple articles recommend using encryption, access controls, and privacy-enhancing technologies like differential privacy and anonymization to protect personal information. Additionally, measures such as network isolation, intrusion detection systems, and regular vulnerability assessments can enhance security. Public education and awareness are also crucial in promoting privacy and security in smart cities.

Alamaniotis et al. 2017 Smart Power System presented an intelligent methodology for enhancing privacy within smart power systems. The proposed methodology utilized demand patterns for several consumers connected to the power grid to provide a new consumption pattern.

Sanduleac et al 2016 Smart Power System addressed two main aspects of smart city implementation; namely: (i) multi-energy streams when different utilities serve different energy networks in the city such as electricity, gas, and heat (ii) the issue of engaging the citizens by sharing their private energy data profile, as an alternative to implementations that fail to progress from small pilots to large deployment.

Alromaihi et al. 2018 Smart Power System identified the main security and privacy challenges in designing IoT architecture in the context of healthcare applications, highlighting the increased use of sensors for medicine and healthcare applications over the last decade. The study identified key threats from personal health related data captured via sensors for e.g. heart rate and also blood pressure and the importance of an integrated security solution for the entire system.

Beltran et al. 2017 Protocols to Improve Security and Privacy This aspect of the literature has focused on encryption algorithms to build in security to smart city systems. As smart cities face a number of challenges connected to security and privacy, some studies proposed various frameworks, models and algorithms to improve these issues

Stromire et al (2018) Protocols to Improve Security and Privacy proposed to integrate an end-to-end cryptography system into smart city solutions at a foundation level. During any data breach, nothing about the data would be revealed by applying this system.

Dewi Rosadi et al. 2018 Operational Vulnerabilities for Smart Cities research discusses about 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

Belanche-Gracia et al. 2015; Use and Adoption of Smart Services by Citizens The study explored how specific technologies (smart bin, smart parking), and data usage (predictive policing, social media monitoring) may produce various privacy concerns.

2.1 INFERENCE FROM LITERATURE SURVEY

This literature survey highlights the various privacy and security concerns associated with the implementation of smart city infrastructure. Many studies propose different approaches and technologies to protect user privacy, such as foggy dummies, blind third party, and double foggy cache. Additionally, privacy-preserving authentication protocols, encryption algorithms, and privacy-enhancing technologies like differential privacy and anonymization are recommended to safeguard personal information. Some studies also emphasize the importance of public education and awareness, integrated security solutions, and end-to-end cryptography systems. The survey also identifies specific technologies and data usage that may produce various privacy concerns in smart cities, such as smart bin, smart parking, predictive policing, and social media monitoring.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENTS

- System - Pentium-IV
- Speed - 2.4GHZ
- Hard disk - 40GB
- RAM - 512MB

SOFTWARE REQUIREMENTS

- Operating System - Windows XP
- Coding language - Java
- IDE – NetBeans
- Database - MYSQL

3.2 REQUIREMENTS IN DECISION BASED RECOMMENDATION SYSTEM IN SMART CITY

- A decision-based recommendation system for a smart city requires several key requirements to be effective. Here are some of the key requirements:

- Data Collection: The system needs to collect data from various sources, including IoT sensors, social media, and other sources to have enough data to make accurate recommendations.

- Data Processing: The system needs to process the collected data to identify patterns and make sense of the data. This includes using data mining, machine learning, and other analytical techniques.

- Decision Models: The system needs to have a set of decision models that can interpret the processed data and make recommendations. These models can be rule-based, model-based, or a combination of both.

- User Feedback: The system needs to incorporate feedback from users to improve its recommendations. This can be done through surveys, user ratings, or other means.

- Integration: The system needs to be integrated with other systems in the smart city ecosystem, including transportation, energy, and waste management systems.

- Security: The system needs to be secure to protect user data and prevent unauthorized access. This includes using encryption, access controls, and other security measures.

- Real-Time Processing: The system needs to be able to process data in real-time to provide timely recommendations.

- Scalability: The system needs to be scalable to handle a large volume of data and users.

- User Interface: The system needs to have a user-friendly interface that allows users to interact with the system easily.

- Cost-effectiveness: The system needs to be cost-effective, with a low cost of ownership and maintenance, to ensure widespread adoption.

Overall, a decision-based recommendation system for a smart city needs to be robust, secure, and scalable, with a user-friendly interface and a low cost of ownership. It also needs to be integrated with other systems in the smart city ecosystem and incorporate user feedback to improve its recommendations over time.

3.3 LIBRARIES REQUIRED IN SMART CITY

There are several libraries and tools that may be required in a decision-based recommendation system for a smart city. Here are some of the commonly used ones:

Python: Python is a popular programming language for data science and machine learning, and is often used for building recommendation systems.

NumPy: NumPy is a Python library for scientific computing and is often used for numerical calculations in machine learning and data science applications.

Pandas: Pandas is a Python library for data manipulation and analysis. It is often used for data preprocessing and cleaning in recommendation systems.

Scikit-learn: Scikit-learn is a Python library for machine learning and is often used for building machine learning models in recommendation systems.

TensorFlow: TensorFlow is a popular machine learning framework for building deep learning models. It is often used for building recommendation systems that use neural networks.

PyTorch: PyTorch is another popular machine learning framework for building deep learning models. It is often used for building recommendation systems that use neural networks.

Apache Spark: Apache Spark is a distributed computing framework that is often used for big data processing and machine learning. It can be used for building recommendation systems that process large volumes of data.

Flask: Flask is a Python web framework that is often used for building web applications. It can be used for building the user interface of a recommendation system.

Dash: Dash is another Python web framework that is often used for building interactive web applications. It can be used for building the user interface of a recommendation system.

Matplotlib: Matplotlib is a Python library for data visualization and is often used for visualizing data in recommendation systems.

These libraries and tools are just a few examples of what may be required in a decision-based recommendation system for a smart city. The specific requirements may vary depending on the use case and the type of recommendation system being built.

3.4 FEATURES OF JAVA

Java is a popular programming language and has a wide range of features that can be useful in building a decision-based recommendation system for a smart city. Here are some of the features of Java that can be beneficial:

Platform Independence: Java is a platform-independent language, meaning that Java programs can run on any platform that has a Java Virtual Machine (JVM) installed. This can be advantageous in building recommendation systems that need to be deployed on multiple platforms.

Object-Oriented Programming: Java is an object-oriented programming language, which means it provides support for creating and using objects in programming. This can be useful in building complex recommendation systems that require the use of multiple objects.

Garbage Collection: Java has automatic memory management through a garbage collector, which frees the programmer from the task of explicitly deallocating memory. This can simplify the process of building and maintaining recommendation systems.

Multithreading: Java provides built-in support for multithreading, which can be beneficial in building recommendation systems that need to handle multiple tasks simultaneously.

Large Standard Library: Java has a large standard library, which provides many built-in functions and classes that can be useful in building recommendation systems. This can help reduce the amount of custom code needed to build the system.

Security: Java has built-in security features, including sandboxing and access controls, which can be useful in building recommendation systems that need to handle sensitive data or interact with external systems.

Scalability: Java is scalable, meaning it can be used to build systems that can handle large amounts of data or traffic. This can be beneficial in building recommendation systems for smart cities that need to handle large volumes of data.

3.5 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

FUNCTIONAL REQUIREMENTS

Functional requirements for a decision-based recommendation system for a smart city could include:

Data Collection and Analysis: The system must be able to collect data from various sources such as sensors, social media, and other sources, and analyze that data to extract meaningful insights.

Recommendation Engine: The system must have a recommendation engine that can generate personalized recommendations based on user preferences, behavior, and historical data.

User Profiles: The system must allow users to create profiles that contain their personal preferences, interests, and other relevant information that can be used to generate personalized recommendations.

Location-based Services: The system must be able to provide location-based services such as finding nearby restaurants, tourist attractions, and other points of interest.

Real-time Alerts: The system must be able to provide real-time alerts to users based on their location and other relevant information, such as traffic congestion, weather conditions, and other potential hazards.

User Feedback and Ratings: The system must allow users to provide feedback and ratings on the recommendations they receive, which can be used to improve the accuracy and relevance of future recommendations.

Integration with Third-party Services: The system must be able to integrate with other services such as transportation services, hotel booking services, and other relevant services to provide a seamless user experience.

Scalability: The system must be able to handle large amounts of data and user traffic, and should be designed with scalability in mind.

Security: The system must be designed with security in mind, including authentication and authorization mechanisms, data encryption, and other security measures to protect user data and ensure the privacy of user information.

NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements for a decision-based recommendation system for a smart city would typically include:

Performance: The system should be able to provide recommendations quickly and efficiently, even when there is a large amount of data to process. It should be able to handle a high volume of requests without slowing down or crashing.

Scalability: The system should be able to scale up or down as needed to accommodate changes in the volume of data or user traffic. It should be able to handle more users or data without requiring significant changes to the system architecture.

Security: The system should be secure and protect user data and privacy. It should use encryption and other security measures to prevent unauthorized access or data breaches.

Usability: The system should be easy to use and intuitive for users. It should have a simple and user-friendly interface, and users should be able to easily understand how to use it.

Reliability: The system should be reliable and consistently provide accurate recommendations. It should be able to handle errors or exceptions gracefully and recover from them quickly.

Maintainability: The system should be easy to maintain and update. It should have a modular and well-organized architecture that makes it easy to modify or add new features.

Compatibility: The system should be compatible with different platforms and devices. It should be able to work seamlessly with different operating systems, browsers, and mobile devices.

Accessibility: The system should be accessible to all users, including those with disabilities. It should comply with accessibility guidelines and be easy to use for users with different needs.

CHAPTER 4

DISCRIPTION OF PROPOSED SYSTEM

4.1 TECHNIQUES USED IN PROPOSED SYSTEM

Collaborative Filtering: Collaborative filtering is a technique that involves analyzing user behavior and preferences to identify patterns and make personalized recommendations. This algorithm looks for users who have similar preferences and recommends items based on what they like.

Content-Based Filtering: Content-based filtering involves recommending items based on their characteristics, such as their genre, type, or description. This algorithm looks for items that are similar to the ones that the user has already liked or viewed.

Matrix Factorization: Matrix factorization is a technique that involves breaking down a large matrix into smaller matrices to identify latent features or patterns. This algorithm is useful in situations where there is a lot of data and it is difficult to identify patterns using other techniques.

Deep Learning: Deep learning is a subset of machine learning that involves training neural networks to make predictions or recommendations. This algorithm is useful for complex recommendation systems that require a lot of data and processing power.

Hybrid Methods: Hybrid methods involve combining two or more algorithms to make more accurate recommendations. For example, a system could use collaborative filtering and content-based filtering to provide more accurate and personalized recommendations.

4.2 PROPOSED SYSTEM

The proposed system is designed to provide an online platform that offers information about a specific city to users who intend to visit it. The system provides an easy-to-use interface, which makes it user-friendly and accessible to everyone. It also has various controls that enhance the user experience, such as a rich user interface.

Moreover, the system makes project management easier and more flexible. The integration of a database approach ensures that the system is efficient, reliable and easily maintainable. It enables the storage of city information files in a centralized database that can be managed by the system.

One of the main benefits of the proposed system is that it can be accessed over an intranet, which is a private network that allows users to access information securely. This means that users can access the system from anywhere within the intranet without worrying about security issues.

In summary, the proposed system is a comprehensive solution that provides online city information to users, making the process of planning a trip more straightforward and more accessible. It incorporates a user-friendly interface, flexible project management, database integration, and intranet access, which make it an efficient and reliable system.

ADVANTAGES OF PROPOSED SYSTEM

There are several advantages of the proposed system, which are outlined below:

- Easy Accessibility: The proposed system is accessible through an intranet, which means that users can access it from anywhere within the network.

This makes it convenient for users to access information about the city they intend to visit without any hassle.

- User-Friendly Interface: The system is designed with a user-friendly interface that enables users to navigate easily and find the information they need quickly. This feature is particularly useful for first-time visitors who may not be familiar with the city.

- Rich User Interface: The system also provides a rich user interface that includes various controls that enhance the user experience. This makes the system visually appealing and easier to use.

- Centralized Database: The system utilizes a centralized database that stores all the city information files. This makes it easy to manage and maintain the information, ensuring that it is always up-to-date and accurate.

- Efficient Project Management: The system makes project management more flexible and easier. This means that the project team can allocate tasks and monitor progress efficiently, ensuring that the project is completed on time and within budget.

- Additional Services: The system also provides additional services to registered users, such as online bookings, travel guides, and recommendations for places to visit. This makes the system a comprehensive platform that offers all the information and services that a user might need when planning a trip.

4.3 SYSTEM ARCHITECTURE

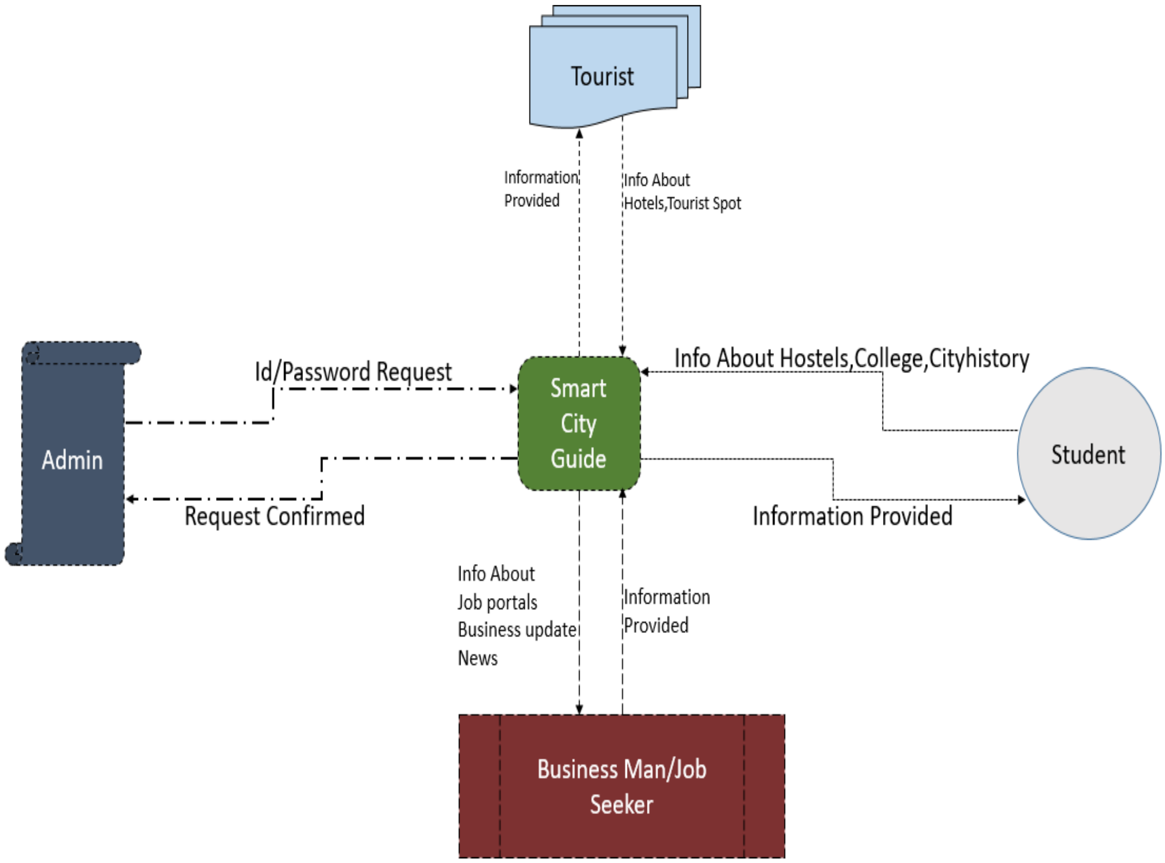


FIGURE 4.3.1: SYSTEM ARCHITECTURE

A decision-based recommendation system for smart cities is a system that provides personalized recommendations to users based on their preferences and needs, with the goal of enhancing their urban experience. The system architecture of this recommendation system involves several components, which are discussed below:
User Interface: The user interface is the front-end component of the system that allows users to interact with the system. It provides an easy-to-use interface that enables users to input their preferences and receive recommendations.

Data Collection: The data collection component collects data about the users and the city. This data includes user profiles, historical user data, and real-time data from sensors and other sources. The data collection component is responsible for gathering and storing this data in a centralized database.

Data Processing: The data processing component is responsible for analyzing the collected data to identify patterns and trends. This component uses machine learning algorithms to process the data and generate recommendations for users.

Decision Engine: The decision engine is the core component of the system that makes decisions based on the processed data. It uses a rule-based system to generate recommendations for users based on their preferences and needs.

Recommendation Engine: The recommendation engine is responsible for generating personalized recommendations for users based on the decisions made by the decision engine. The recommendation engine uses collaborative filtering and content-based filtering techniques to provide recommendations that are relevant to the user.

Output Interface: The output interface is the component that delivers the recommendations to the user. It presents the recommendations in a user-friendly format, such as a list or a map, and allows users to interact with the recommendations.

In summary, the system architecture of a decision-based recommendation system for smart cities involves several components, including the user interface, data collection, data processing, decision engine, recommendation engine, and output interface. These components work together to provide personalized recommendations to users, with the goal of enhancing their urban experience.

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 MODULES

- Administrator Module
- Tourist Module
- Jobseeker Module
- Businessman Module
- Student module

ADMINISTRATOR MODULE

The administrator module is a crucial component of the decision-based recommendation system for smart cities, as it enables the system administrator to manage the system and ensure its smooth operation. The administrator module is used in several ways in the system, as outlined below:

User Management: The administrator module allows the system administrator to manage the user accounts and profiles. The administrator can create, modify or delete user accounts, and also manage user preferences and historical data. This ensures that the user data is up-to-date and accurate, and that the system can provide relevant and personalized recommendations to users.

Data Management: The administrator module enables the system administrator to manage the data collected by the system. This includes the real-time data collected from sensors and other sources, as well as the historical user data. The administrator can ensure that the data is stored securely and can be accessed efficiently, which ensures that the system can process the data and generate recommendations quickly and accurately.

System Configuration: The administrator module also enables the system administrator to configure the system settings and parameters. This includes setting the rules and thresholds used by the decision engine, configuring the recommendation engine, and adjusting the user interface. This ensures that the system is customized to the needs of the users and is optimized for performance and accuracy.

Performance Monitoring: The administrator module allows the system administrator to monitor the performance of the system in real-time. The administrator can track the system metrics, such as response time and recommendation accuracy, and take appropriate actions to optimize the system performance.

System Maintenance: The administrator module also facilitates system maintenance activities, such as backups, updates, and troubleshooting. The administrator can ensure that the system is maintained properly, and any issues that arise are resolved quickly and efficiently.

TOURIST MODULE

The tourist module is an essential component of the decision-based recommendation system for smart cities, as it provides personalized recommendations to tourists based on their preferences and interests. The tourist module is used in several ways in the system, as outlined below:

Profile Creation: The tourist module allows tourists to create their profiles, which includes information such as their name, age, gender, and interests. This information is used by the system to generate personalized recommendations.

Recommendation Generation: The tourist module enables the system to generate recommendations for tourists based on their profile and preferences. The system uses machine learning algorithms to process the data collected from the tourist and other sources, such as sensor data, to generate recommendations for tourist attractions, restaurants, and other points of interest.

Interactive Map: The tourist module provides an interactive map that shows the tourist attractions and other points of interest in the city. The map allows tourists to explore the city and discover new places based on their interests and preferences.

Real-time Information: The tourist module provides real-time information about the tourist attractions and other points of interest, such as current events and special offers. This enables tourists to make informed decisions about where to go and what to do in the city.

Feedback and Ratings: The tourist module allows tourists to provide feedback and ratings on the tourist attractions and other points of interest. This feedback is used by the system to improve the recommendations provided to other tourists in the future.

JOB-SEEKER MODULE

The job-seeker module is an important component of the decision-based recommendation system for smart cities, as it helps job seekers find suitable employment opportunities based on their qualifications and preferences. The job-seeker module is used in several ways in the system, as outlined below:

Profile Creation: The job-seeker module allows job seekers to create their profiles, which includes information such as their name, qualifications, work experience, and job preferences. This information is used by the system to generate personalized recommendations for job openings.

Job Recommendation Generation: The job-seeker module enables the system to generate recommendations for job openings based on the job seeker's profile and preferences. The system uses machine learning algorithms to process the data collected from the job seeker and other sources, such as job postings and company data, to generate relevant job recommendations.

Application Tracking: The job-seeker module allows job seekers to track the status of their job applications. This includes information such as the status of the application, the employer's response, and any updates or feedback. This enables job seekers to stay informed and take appropriate actions, such as following up with the employer.

Career Development Resources: The job-seeker module provides career development resources, such as job search tips, resume writing guidance, and interview preparation advice. These resources help job seekers improve their job search skills and increase their chances of finding suitable employment.

Networking Opportunities: The job-seeker module provides networking opportunities for job seekers, such as job fairs, networking events, and online communities. These opportunities help job seekers connect with potential employers and expand their professional network.

Businessman Module

The business man module is an essential component of the decision-based recommendation system for smart cities, as it helps business owners and entrepreneurs find opportunities to grow their businesses and increase their profitability. The business man module is used in several ways in the system, as outlined below:

Business Profile Creation: The business man module allows business owners and entrepreneurs to create their business profiles, which includes information such as their business name, location, industry, and products or services offered. This information is used by the system to generate personalized recommendations for business opportunities.

Business Recommendation Generation: The business man module enables the system to generate recommendations for business opportunities based on the business owner's profile and preferences. The system uses machine learning algorithms to process the data collected from the business owner and other sources, such as market trends and customer data, to generate relevant business recommendations.

Financial Analysis: The business man module provides financial analysis tools, such as profit and loss statements and cash flow analysis, to help business owners and entrepreneurs make informed decisions about their businesses. This information helps them identify areas for improvement and develop strategies for increasing profitability.

Networking Opportunities: The business man module provides networking opportunities for business owners and entrepreneurs, such as business events, trade shows, and online communities. These opportunities help business owners connect with potential partners, investors, and customers, and expand their professional network.

Marketing Strategies: The business man module provides marketing strategies and tools, such as social media marketing and email marketing, to help business owners and entrepreneurs reach their target audience and increase their visibility. This helps them attract new customers and increase their revenue.

STUDENT MODULE

The student module is an important component of the decision-based recommendation system for smart cities, as it helps students find suitable educational opportunities based on their interests and qualifications. The student module is used in several ways in the system, as outlined below:

Profile Creation: The student module allows students to create their profiles, which includes information such as their name, educational qualifications, interests, and career goals. This information is used by the system to generate personalized recommendations for educational opportunities.

Course Recommendation Generation: The student module enables the system to generate recommendations for courses and educational programs based on the student's profile and preferences. The system uses machine learning algorithms to process the data collected from the student and other sources, such as course catalogs and student feedback, to generate relevant course recommendations.

Application Tracking: The student module allows students to track the status of their course applications. This includes information such as the status of the application, the institution's response, and any updates or feedback. This enables students to stay informed and take appropriate actions, such as following up with the institution.

Scholarship and Financial Aid Information: The student module provides information about scholarships and financial aid opportunities that are available to students. This information helps students access financial support that can help them achieve their educational goals.

Career Development Resources: The student module provides career development resources, such as job search tips, resume writing guidance, and interview preparation advice. These resources help students improve their employability and increase their chances of finding suitable employment after graduation.

5.2 STEPS IN DECISION BASED RECOMMENDATION SYSTEM FOR SMART CITY

The decision-based recommendation system for smart city typically involves several steps, which are outlined below:

Data Collection: The system collects data from various sources, such as city databases, social media, and user feedback. The data includes information about the city's infrastructure, services, events, and other relevant factors.

Data Processing: The system processes the collected data using machine learning algorithms and data analytics tools. The processing helps to identify patterns, trends, and relationships in the data, which can be used to generate recommendations.

User Profile Creation: The system allows users to create their profiles, which include information such as their interests, preferences, and demographics. The profile creation helps to personalize the recommendations for each user.

Recommendation Generation: The system generates recommendations based on the user's profile and the processed data. The recommendations can be in the form of suggested tourist attractions, educational courses, job opportunities, business ventures, or other relevant information.

Feedback Collection: The system collects feedback from the users about the recommendations. The feedback can include ratings, reviews, and comments. The feedback helps to improve the accuracy and relevance of the recommendations.

Continuous Improvement: The system continuously improves based on the feedback and new data. The system updates the recommendations, algorithms, and other components to ensure that they remain relevant and effective.

In summary, the decision-based recommendation system for smart city involves data collection, processing, user profile creation, recommendation generation, feedback collection, and continuous improvement. The system is designed to provide personalized and relevant recommendations for various users, such as tourists, students, job seekers, business owners, and others. The system aims to enhance the quality of life in the city by providing useful information and opportunities for growth and development.

5.3 SYSTEM DESIGN

The design of a decision-based recommendation system for a smart city involves several key components that need to be considered. Here is an overview of the system design for such a recommendation system:

Data Collection: The first step is to identify the data sources that will be used to build the recommendation system. These may include sensor data, social media data, user profiles, and other sources of data that can provide insight into the needs and preferences of city residents.

Data Preprocessing: Once the data has been collected, it needs to be preprocessed to remove noise, missing data, and other inconsistencies. This may involve data cleaning, data transformation, and data normalization.

Machine Learning: The next step is to use machine learning algorithms to build a recommendation model based on the preprocessed data. This may involve techniques such as collaborative filtering, content-based filtering, or hybrid approaches.

Decision Making: Once the recommendation model has been built, the system needs to be able to make decisions based on the recommendations. This may involve integrating the recommendation model with other decision-making systems in the smart city, such as traffic management or emergency response systems.

User Interface: The recommendation system needs to provide a user interface that allows city residents to access and interact with the system. This may involve a web-based interface or a mobile app that can be used to access recommendations and provide feedback.

Deployment: Finally, the system needs to be deployed in a production environment, which may involve integrating the system with existing infrastructure and ensuring that it is scalable, reliable, and secure.

5.4 DATASETS

Dataset implementation is a crucial component of the decision-based recommendation system for smart city. The system requires large and diverse datasets to generate accurate and relevant recommendations for various users. The implementation of the dataset typically involves the following steps:

Dataset Identification: The system identifies relevant datasets from various sources, such as government databases, social media platforms, and other public sources. The datasets can include information about the city's infrastructure, services, events, demographics, and other relevant factors.

Dataset Collection: The system collects the identified datasets using data scraping tools, APIs, or other data collection methods. The collection process can involve cleaning and preprocessing the data to remove duplicates, errors, and other irrelevant information.

Dataset Integration: The system integrates the collected datasets into a centralized database, which can be accessed by various components of the system. The integration process can involve merging, matching, and transforming the data to ensure compatibility and consistency.

Dataset Storage: The system stores the integrated dataset in a secure and scalable storage system, such as a cloud-based database or a data warehouse. The storage system enables efficient retrieval and processing of the data.

Dataset Processing: The system processes the stored dataset using various data analytics tools, such as machine learning algorithms, data mining, and statistical analysis. The processing helps to identify patterns, trends, and relationships in the data, which can be used to generate recommendations.

Dataset Updating: The system continuously updates the dataset to ensure that it remains current and relevant. The updating process can involve adding new data, removing outdated data, and updating existing data.

5.5 INPUT AND OUTPUT DESIGN

INPUT DESIGN

Input design is an important aspect of the decision-based recommendation system for smart city. It involves the design and development of user interfaces that enable users to input their preferences, needs, and other relevant information. The input design typically involves the following steps:

User Profiling: The system collects information about the user's preferences, needs, and interests through various sources, such as social media, surveys, and user interactions. The user profiling helps to create a personalized user profile that captures the user's preferences and needs.

User Interface Design: The system designs a user interface that enables users to input their preferences, needs, and other relevant information. The user interface can include various input fields, such as drop-down menus, checkboxes, radio buttons, and text boxes.

Input Validation: The system validates the user inputs to ensure that they are accurate and complete. The validation process can involve checking for errors, duplicates, and missing information.

Input Preprocessing: The system preprocesses the user inputs to transform them into a format that can be used by the recommendation engine. The preprocessing can involve cleaning, filtering, and transforming the input data.

Input Integration: The system integrates the user inputs with the dataset to generate personalized recommendations. The integration process can involve matching the user inputs with the relevant data in the dataset.

Input Updating: The system continuously updates the user inputs to ensure that the recommendations remain relevant and accurate. The updating process can involve soliciting feedback from users, updating user profiles, and reprocessing the input data.

OUTPUT DESIGN

Output design is an important component of the decision-based recommendation system for smart city. It involves the design and development of user interfaces that enable users to view and interact with the recommendations generated by the system. The output design typically involves the following steps:

Recommendation Generation: The system generates recommendations based on the user inputs and other relevant data. The recommendations can include information about tourist attractions, job opportunities, businesses, and educational institutions.

Recommendation Presentation: The system presents the recommendations to the user in a user-friendly and intuitive manner. The presentation can include various types of visualizations, such as graphs, charts, and maps, as well as text descriptions and multimedia content.

Recommendation Filtering: The system filters the recommendations based on the user's preferences, needs, and other relevant factors. The filtering process can involve applying various algorithms and techniques, such as collaborative filtering and content-based filtering.

Recommendation Ranking: The system ranks the recommendations based on their relevance and importance to the user. The ranking can involve applying various metrics, such as popularity, user ratings, and sentiment analysis.

Recommendation Personalization: The system personalizes the recommendations based on the user's profile and other relevant data. The personalization can involve adapting the recommendations to the user's preferences, needs, and interests.

Recommendation Feedback: The system solicits feedback from the user about the recommendations. The feedback can involve asking the user to rate the recommendations, provide comments, or suggest improvements

.CHAPTER 6

RESULTS AND DISCUSSION

Result and discussion in the decision-based recommendation system for smart city involves evaluating the performance and effectiveness of the system. This can involve analyzing various metrics and indicators, such as user satisfaction, recommendation accuracy, and system efficiency.

To evaluate the performance of the system, various methods can be used, including user surveys, user testing, and data analysis. User surveys can be conducted to gather feedback and opinions from the users about the system's recommendations and user interface. User testing can be done to assess how users interact with the system and to identify any usability issues or bugs. Data analysis can be used to evaluate the system's performance based on various metrics, such as recommendation accuracy and response time.

The results and discussion of the evaluation can provide insights into the strengths and weaknesses of the system and suggest areas for improvement. For example, if the system's recommendation accuracy is found to be low, further analysis can be done to identify the causes and develop solutions. If the user feedback indicates that the user interface is difficult to use, changes can be made to improve the usability of the system.

In addition to evaluating the system's performance, the results and discussion can also provide insights into the impact of the system on the users and the city. For example, if the system is found to be effective in recommending job opportunities, this can have a positive impact on the employment rate and the overall economic development of the city.

In conclusion, result and discussion in the decision-based recommendation system for smart city involves evaluating the performance and impact of the system. Various methods can be used to gather feedback and analyze data to assess the system's effectiveness and identify areas for improvement. The results and discussion can provide insights into the strengths and weaknesses of the system and suggest ways to enhance its functionality and impact.

CHAPTER 7

CONCLUSION

The decision-based recommendation system for smart city is a promising technology that can offer several benefits to users and city management. The system's primary objective is to provide personalized recommendations to users based on their preferences and needs. This means that the system can provide more relevant and useful information to users compared to traditional information sources.

For example, tourists can receive recommendations for the best places to visit, eat, and stay based on their interests and budget. Job-seekers can receive recommendations for job opportunities based on their skills and experience. Business owners can receive recommendations for the best locations to start a business based on market demand and competition. Students can receive recommendations for courses and programs based on their academic performance and career goals. Administrators can receive recommendations for policy and resource allocation based on data analysis and citizen feedback.

The system's architecture involves several components, including data collection and preprocessing, recommendation engine, and user interface. The data collection and preprocessing stage involves gathering data from various sources, such as user profiles, city databases, and external APIs, and preparing the data for analysis. The recommendation engine uses various algorithms and techniques, such as collaborative filtering and content-based filtering, to generate recommendations. The user interface provides a platform for users to interact with the system and receive recommendations.

The implementation of the system involves several steps, including dataset creation, input and output design, and user testing. The dataset creation involves identifying relevant data sources and collecting data in a structured format. The input and output design involve designing the user interface and defining the system's inputs and outputs. The user testing involves evaluating the system's performance and user experience.

The evaluation of the system's performance can involve analyzing various metrics, such as recommendation accuracy, user satisfaction, and system efficiency. The results of the evaluation can provide insights into the system's strengths and weaknesses and suggest areas for improvement. For example, if the recommendation accuracy is low, the system's algorithms can be adjusted or improved. If the user interface is difficult to use, the design can be updated to improve usability.

In conclusion, the decision-based recommendation system for smart city is a powerful tool that can enhance the user experience, promote economic development, and improve city management. With further research and development, the system has the potential to become an integral part of smart city infrastructure and contribute to the creation of more livable, sustainable, and prosperous cities.

FUTURE WORK

There are several potential areas for future work in the development and implementation of decision-based recommendation systems for smart cities. Some of these areas include:

Integration with emerging technologies: As new technologies such as artificial intelligence, blockchain, and Internet of Things continue to evolve, decision-based recommendation systems can integrate these technologies to provide more accurate, efficient, and secure recommendations.

Personalization and adaptation: The future of recommendation systems lies in creating personalized and adaptive recommendations based on user behavior, feedback, and preferences. As more data becomes available, systems can use machine learning and deep learning algorithms to generate more personalized recommendations.

Multi-objective optimization: Decision-based recommendation systems can be expanded to optimize multiple objectives simultaneously, such as maximizing economic development, minimizing traffic congestion, and reducing carbon emissions. This can involve developing multi-criteria decision-making models that consider various trade-offs and preferences.

Collaborative decision-making: Involving citizens, stakeholders, and experts in the decision-making process can lead to more transparent, accountable, and equitable outcomes. Future work can focus on developing collaborative decision-making frameworks that integrate citizen feedback and preferences.

Real-time data analytics: Decision-based recommendation systems can be enhanced by integrating real-time data analytics and visualization tools that enable city managers and users to monitor and respond to changing conditions quickly.

Integration with smart transportation: Smart transportation systems, such as intelligent traffic management and connected vehicles, can be integrated with decision-based recommendation systems to provide more accurate and efficient recommendations for commuting, parking, and public transportation.

REFERENCES

- Appadurai A., (1996). Modernity at Large. Cultural Dimensions of Globalization, University of Minnesota Press, Minneapolis.
- Arato A., Cohen J. L., (1992). Civil Society and Political Theory, MIT Press, Cambridge (MA).
- Atkinson R., Blandy S., (2006).Gated Communities, Routledge, New York.
- Baumgardner J.R., (1988) The Division of Labor, Local Markets, And Worker Organization, in <<Journal of Political Economy>>, XCVI, 3, pp. 509-527.
- Becker, G.S., Human Capital (1964) A Theoretical and Empirical Analysis, with Special Reference to Education, New York, Columbia University Press (2008) trad. It. Il capitale umano, Roma-Bari, Laterza.
- Bonomi A., (2013). Il capitalismo in-finito, Torino, Einaudi.
- Bonomi A., Borgna E., (2011). Elogio della depressione, Torino, Einaudi.
- Bonomi A., Masiero R., (2014). Dalla smart city alla smart land, Marsilio.
- Dematteis G., (2012). Le grandi città italiane. Società e territori da ricomporre, Marsilio, Venezia.
- Farias I., Bender T., (2012). Urban Assemblages: How Actor-Network Theory Changes Urban Studies, Routledge, London.

APPENDIX

(A) SOURCE CODE

```
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 imagevew;
   TextView name;
   DatabaseReference rootref;
```

```
    FirebaseAuth mauth;
    FirebaseUser 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);
        imagevew = (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_home);
        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()).transform(new
circle(getApplicationContext())).into(imagevew);
```

```
                name.setText("Prasath");
                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;
}
```

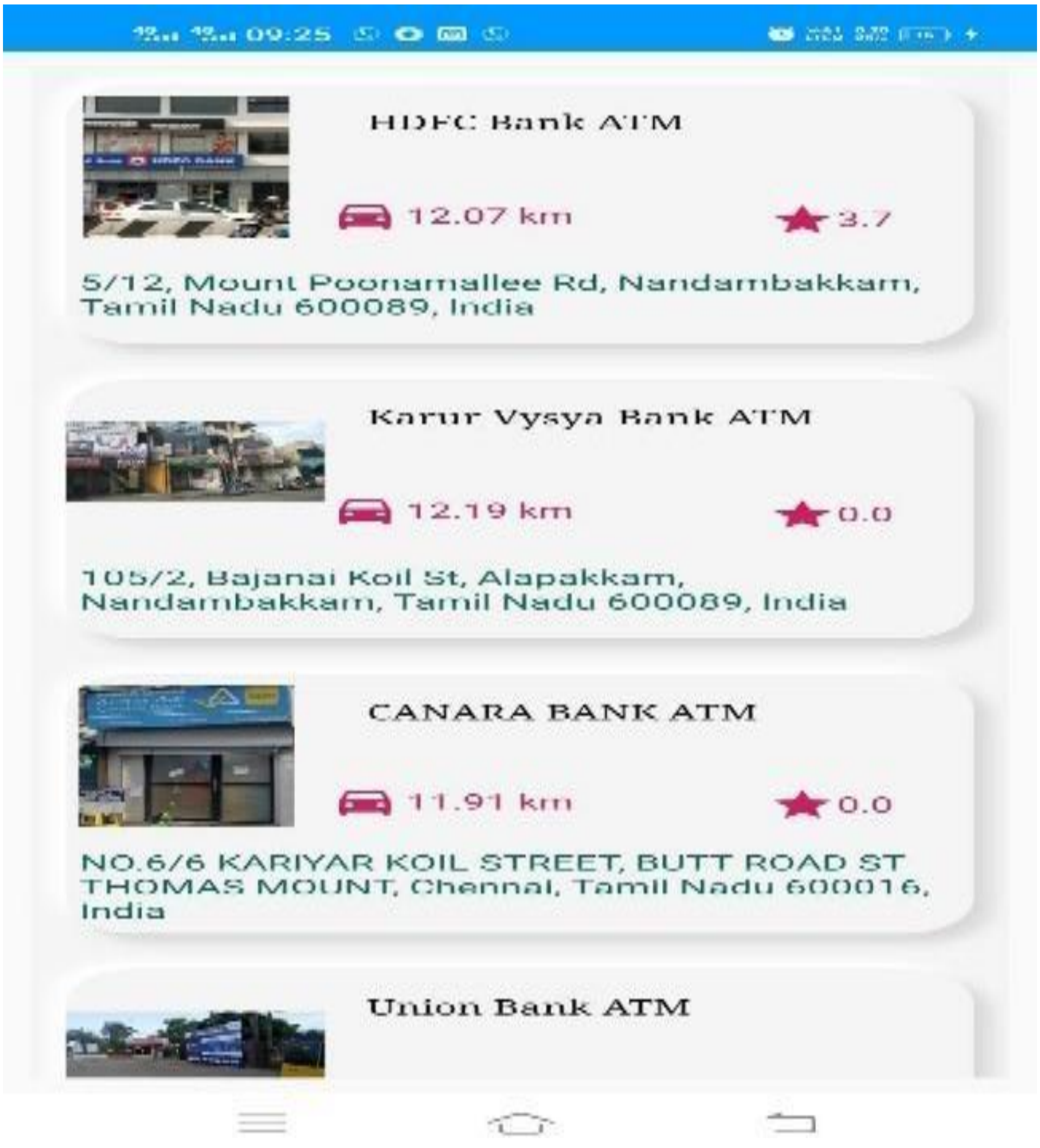
```
    toolbar.setTitle(title);
}

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(true);
            break;

        case R.id.nav_category:
            fragment = new fragmentCategory();
            navigationView.getMenu().getItem(1).setChecked(true);
            break;

        case R.id.nav_Location:
            fragment = new fragmentLocation();
            navigationView.getMenu().getItem(2).setChecked(true);
            break;
        case R.id.nav_save:
            fragment = new fragmentSaved();
            navigationView.getMenu().getItem(3).setChecked(true);
            break;
        case R.id.nav_profile:
            fragment = new fragmentProfile();
```

```
                navigationView.getMenu().getItem(4).setChecked(true);
                break;
            case R.id.nav_logout:
                logOut();


                navigationView.getMenu().getItem(5).setChecked(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 **/
        setScreenTitle(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;
    }







}
```

(B) OUTPUT SCREENSHOTS



HDFC Bank ATM

12.07 km

3.7

5/12, Mount Poonamallee Rd, Nandambakkam, Tamil Nadu 600089, India

Karur Vysya Bank ATM

12.19 km

0.0

105/2, Bajanai Koil St, Alapakkam, Nandambakkam, Tamil Nadu 600089, India

CANARA BANK ATM

11.91 km

0.0

NO.6/6 KARIYAR KOIL STREET, BUTT ROAD ST THOMAS MOUNT, Chennai, Tamil Nadu 600016, India

Union Bank ATM

FIGURE B.1 Nearby ATM

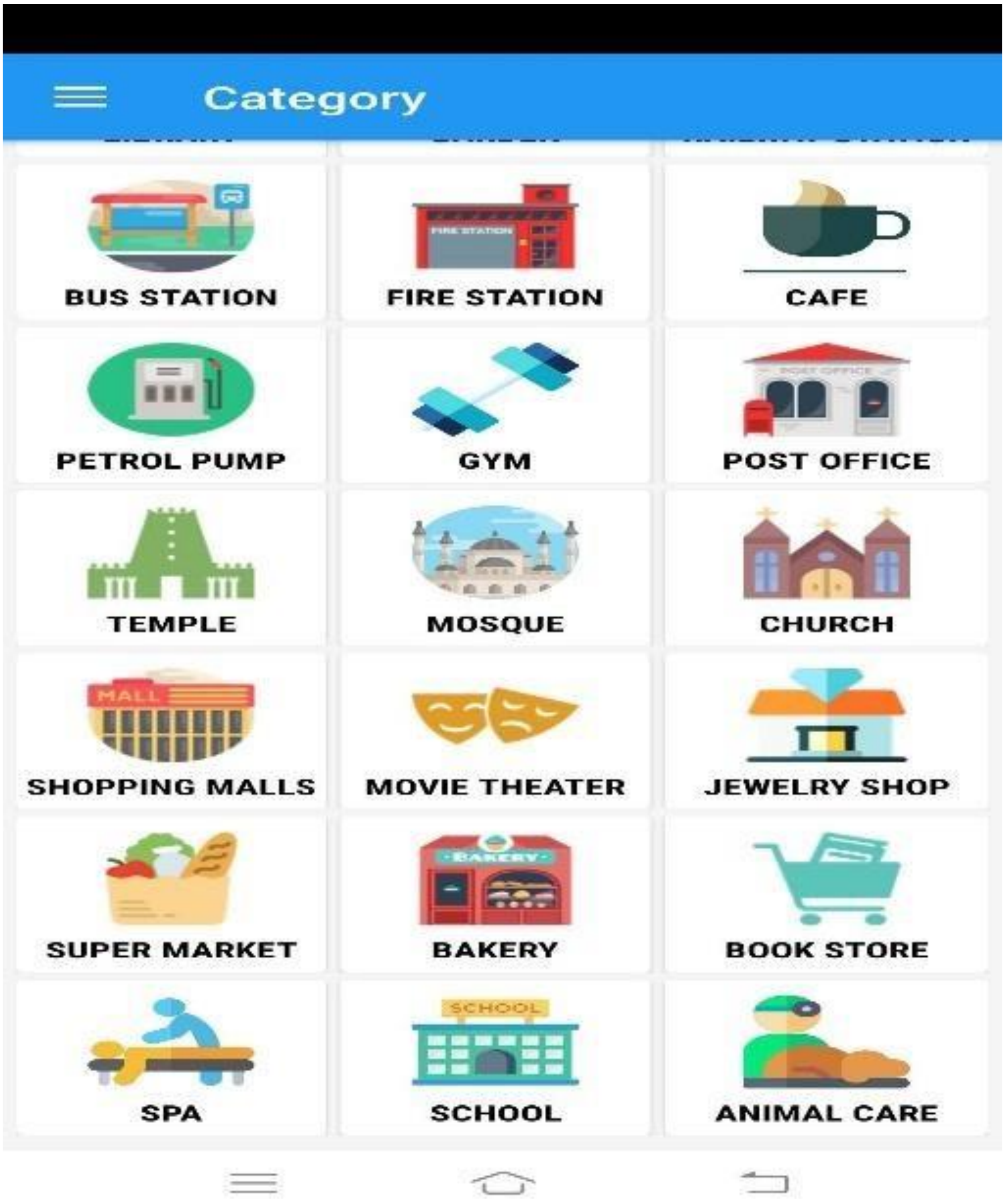


FIGURE B.2 Category Menu

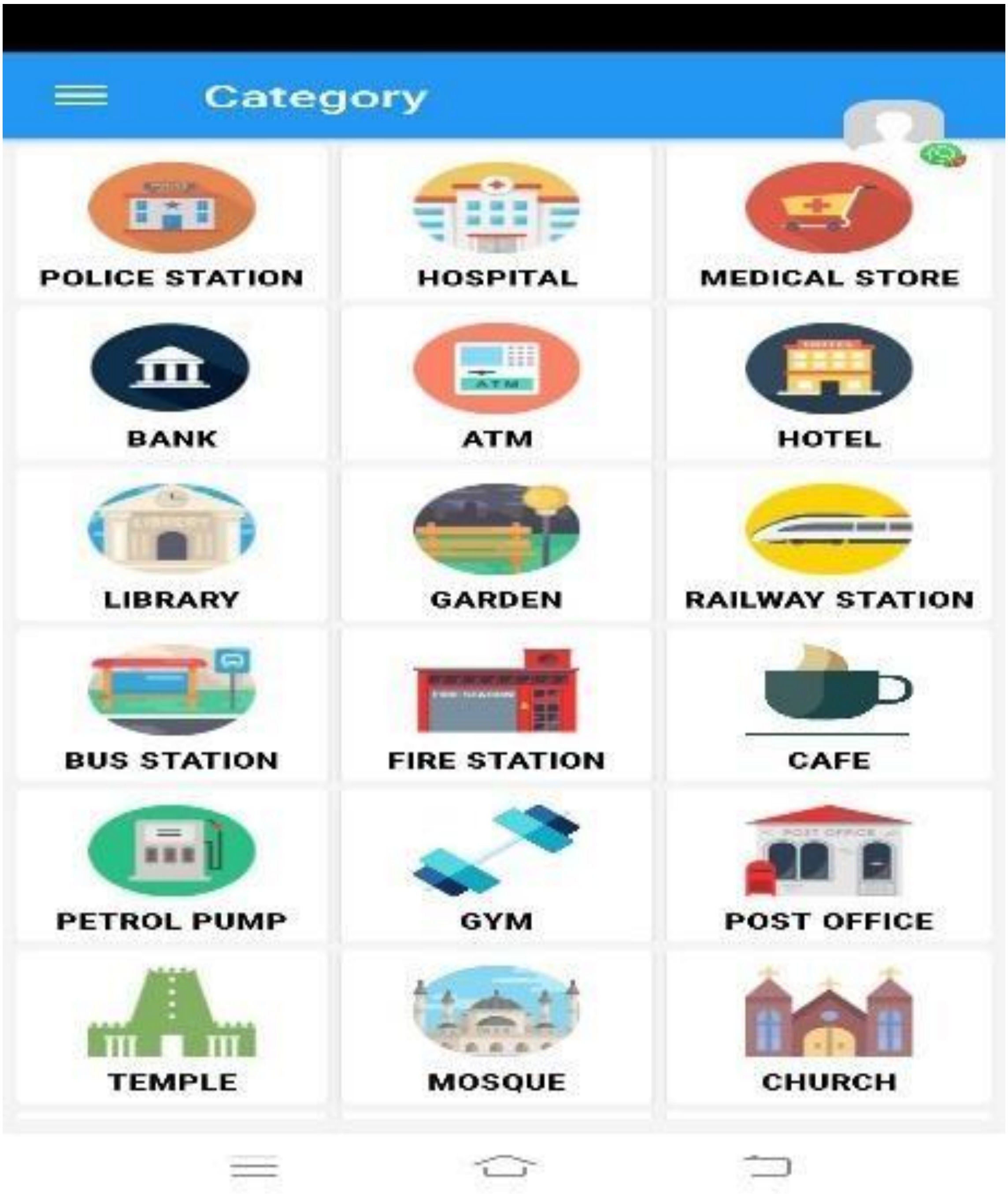


FIGURE B.3 Category Menu 2

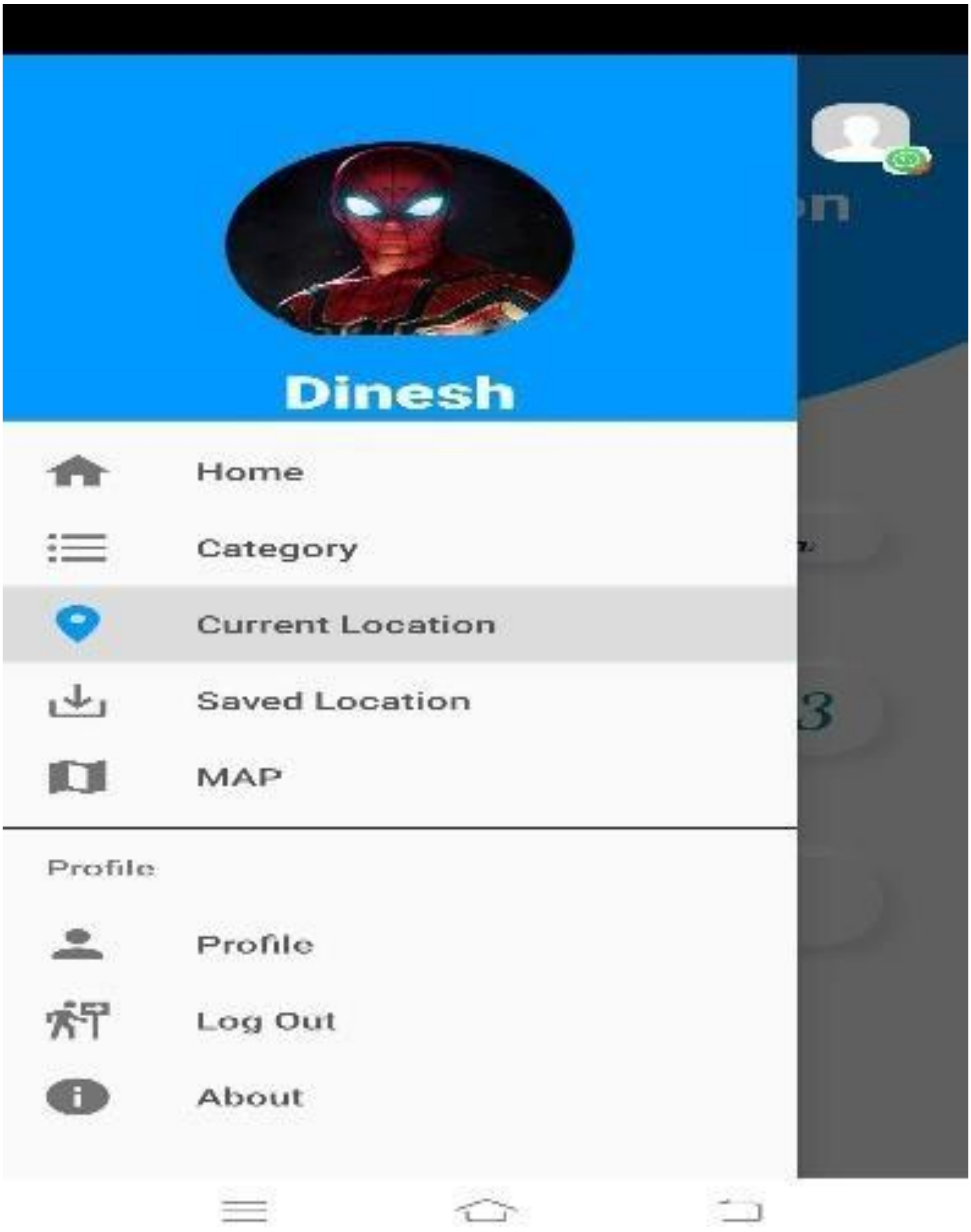
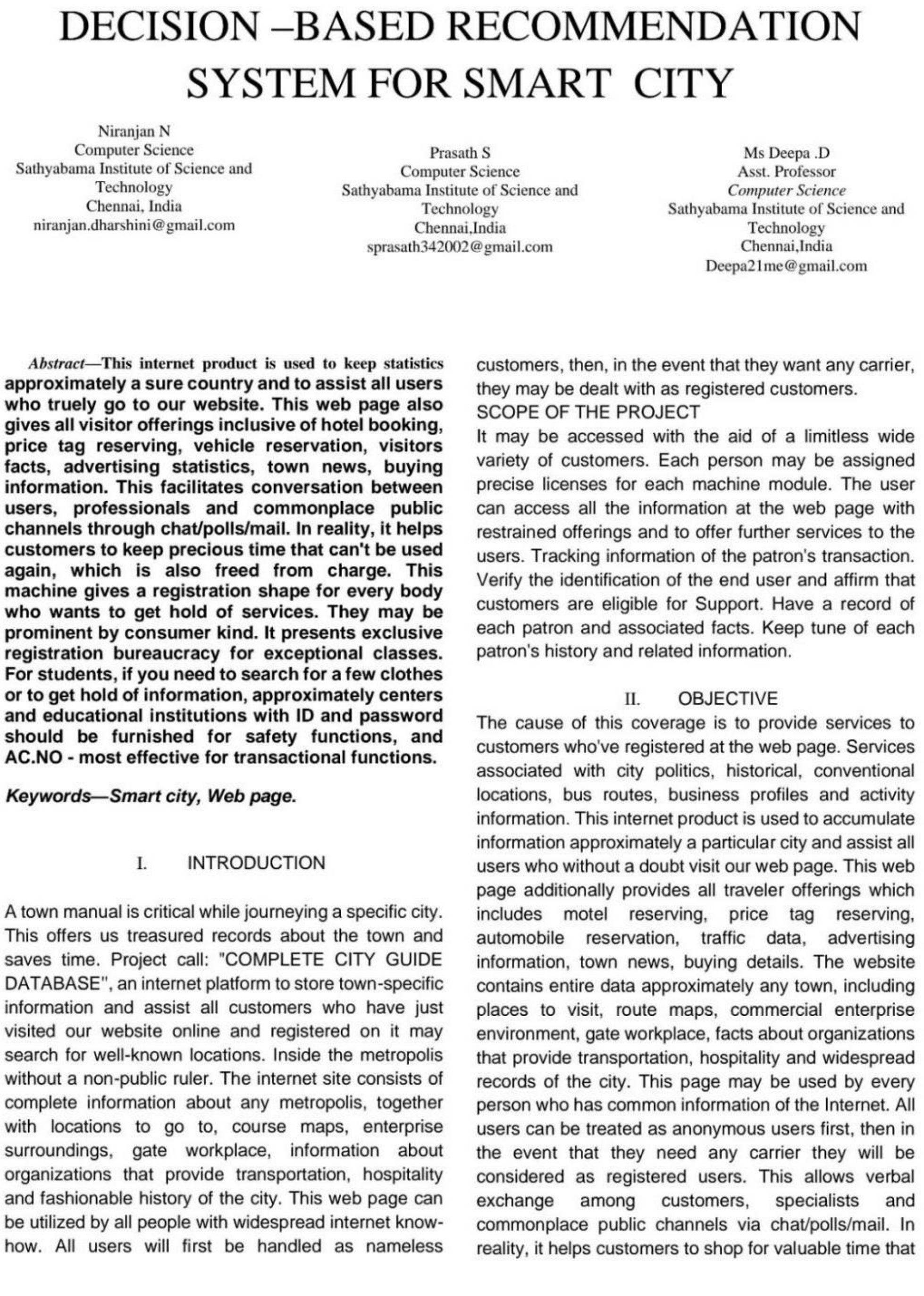


FIGURE B.4 Menu

(C) RESEARCH PAPER



DECISION –BASED RECOMMENDATION SYSTEM FOR SMART CITY

Niranjan N
Computer Science
Sathyabama Institute of Science and Technology
Chennai, India
niranjan.dharshini@gmail.com

Prasath S
Computer Science
Sathyabama Institute of Science and Technology
Chennai,India
sprasath342002@gmail.com

Ms Deepa .D
Asst. Professor
Computer Science
Sathyabama Institute of Science and Technology
Chennai,India
Deepa21me@gmail.com

Abstract—**This internet product is used to keep statistics approximately a sure country and to assist all users who truely go to our website. This web page also gives all visitor offerings inclusive of hotel booking, price tag reserving, vehicle reservation, visitors facts, advertising statistics, town news, buying information. This facilitates conversation between users, professionals and commonplace public channels through chat/polls/mail. In reality, it helps customers to keep precious time that can't be used again, which is also freed from charge. This machine gives a registration shape for every body who wants to get hold of services. They may be prominent by consumer kind. It presents exclusive registration bureaucracy for exceptional classes. For students, if you need to search for a few clothes or to get hold of information, approximately centers and educational institutions with ID and password should be furnished for safety functions, and AC.NO - most effective for transactional functions.**

Keywords—Smart city, Web page.

I. INTRODUCTION

A town manual is critical while journeying a specific city. This offers us treasured records about the town and saves time. Project call: "COMPLETE CITY GUIDE DATABASE", an internet platform to store town-specific information and assist all customers who have just visited our website online and registered on it may search for well-known locations. Inside the metropolis without a non-public ruler. The internet site consists of complete information about any metropolis, together with locations to go to, course maps, enterprise surroundings, gate workplace, information about organizations that provide transportation, hospitality and fashionable history of the city. This web page can be utilized by all people with widespread internet know-how. All users will first be handled as nameless customers, then, in the event that they want any carrier, they may be dealt with as registered customers.

SCOPE OF THE PROJECT

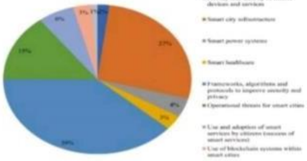
It may be accessed with the aid of a limitless wide variety of customers. Each person may be assigned precise licenses for each machine module. The user can access all the information at the web page with restrained offerings and to offer further services to the users. Tracking information of the patron's transaction. Verify the identification of the end user and affirm that customers are eligible for Support. Have a record of each patron and associated facts. Keep tune of each patron's history and related information.

II. OBJECTIVE

The cause of this coverage is to provide services to customers who've registered at the web page. Services associated with city politics, historical, conventional locations, bus routes, business profiles and activity information. This internet product is used to accumulate information approximately a particular city and assist all users who without a doubt visit our web page. This web page additionally provides all traveler offerings which includes motel reserving, price tag reserving, automobile reservation, traffic data, advertising information, town news, buying details. The website contains entire data approximately any town, including places to visit, route maps, commercial enterprise environment, gate workplace, facts about organizations that provide transportation, hospitality and widespread records of the city. This page may be used by every person who has common information of the Internet. All users can be treated as anonymous users first, then in the event that they need any carrier they will be considered as registered users. This allows verbal exchange among customers, specialists and commonplace public channels via chat/polls/mail. In reality, it helps customers to shop for valuable time that can not be again, which is also freed from charge. This device gives a registration form for absolutely everyone who desires to receive offerings. They can be outstanding by using person kind. It offers distinct registration forms for different categories. For students, in case you want to look for a few fabric or to obtain records about instructional facilities and establishments / For businessmen, to receive statistics about any type of commercial enterprise / For travelers, to acquire hospitality services / In the case of task seekers, to get hold of statistics about every job to be had, that with ID and password ought to offer for protection purposes, and AC.NO - best within the safety of the transaction.

III. LITERATURE OVERVIEW

The literature focuses on numerous key large subjects related to clever towns, specifically: privacy and security of mobile gadgets and offerings; the metropolis's infrastructure and technical structure; electricity systems in clever cities; health care; security and privateness systems; algorithms and protocols; operational threats to smart cities; utility of blockchain answers in smart towns; social networks and clever cities. The parent beneath shows the studies topics and the proportion of articles underneath each awesome topic. Related subjects and hyperlinks are indexed inside the desk underneath. Subsections of the thing have sparked dialogue on every subject, with insight into the studies being executed below every important subject matter.



Privacy and Security of Mobile Devices and Services

Mobile devices are the spine of verbal exchange with the community infrastructure of smart cities, but they pose new challenges for security and person privateness, as sensitive data may be prone to 1/3-birthday party attacks. The examine affords three new tactics to cellular privacy in clever cities. The first method used the idea of a cloud monster to shield user privacy; the second includes a blind 1/3 birthday party, wherein a believe courting is hooked up to guard the user from the company; a 3rd technique is the idea of a double cloud cache to resolve the peer agree with hassle thru a conventional collaborative technique. Privacy Authentication (PPA) protocols have emerged as cell carrier protocols as a cryptographic technique to presenting authentication and privateness safety capabilities for smart states. The examine offered in Li et al. (2019) analyzed the suitability of the PPA protocol for cell services in an ordinary cell software in an city surroundings.

Smart City Infrastructure

Several chapters have targeted on smart urban infrastructure and how to triumph over security and privacy troubles in clever cities (Abosaq 2019; Ainane et al. 2018; Alandjani 2018; Antoine Picon 2019; Awad et al. 2019; Baryshev et al. 2016; Bernardus). And others, 2018; Chatterjee et al., 2017; from Amorim et al., 2019). Research argues that privateness can be without difficulty compromised due to the excessive level of interaction between humans, machines and sensors, to the extent that there's a want for complete safety of this information (Antoine Picon, 2019; Elmagrabi and Losavio, 2014). Research has diagnosed the advantages of an extra strategic consciousness on urban safety, moving privateness guidelines beyond statistics to prudent security policies (Efthymiopoulos 2015). A take a look at by using Ferraz and Ferraz (2014a) argues that records safety includes now not handiest confidentiality, privacy, integrity, availability, but also interoperability security, that is the concept of failing the entire city machine.

Smart Power System

Captive state structures are potentially essential to the overall protection and private infrastructure as 0.33 events related to the network can monitor account utilization and are expecting patron conduct. Wireless era, which specializes in the nature of many systems that provide and manage warmth and light in clever towns, can divulge networks to security vulnerabilities. Alamaniotis et al.(2017) affords a clever method for increasing privacy in smart grids. The proposed method used more than one purchaser demand fashions with a grid to create a new consumption version. In the take a look at by way of Sanduleac et al. (2016) considered primary components of smart city implementation; Namely: (i) multi-power flows where distinctive utilities serve exceptional power inside the metropolis networks such as electricity, gasoline and warmth (ii) problems to engage citizens by using sharing non-public power data as an alternative to put into effect those who cannot circulate from small. To installation huge pilot projects.

Smart Healthcare

The protection and privateness of health services and concepts in smart cities is a key element inside the basic minimal data disclosure and facts security infrastructure (Maria De Fuentes et al. 2018). A look at by way of Alromaihi et al. (2018) diagnosed the principle safety and privateness concerns in IoT architecture design in the context of healthcare programs, because of the growing use of sensors in medication and healthcare over the past decade. The take a look at of key threats to personal protection statistics received via sensors, for instance. Coronary heart charge as well as blood pressure and the importance of a entire protection answer for the whole machine.

Tables, fashions, algorithms and protocols to improve safety and privateness

As smart cities face a number of protection and privateness-related challenges, diverse studies have proposed special frameworks, models and algorithms to deal with those challenges (Al-Dubhani et al. 2018; Antonopoulos et al. 2017; Avgeru et al., 2016; : Beltran et al.). And others, 2017; Burange and Misalkar, 2015; Caliero et al., 2015). This literature targeted on encryption algorithms for capturing urban structures. Antonopoulos et al (2017) test algorithms for excessive-stage security functions the usage of the improvement of a wi-fi sensor network (WSN). Stromir and Potochny-Jones (2018) proposed to integrate an cease-to-cease cryptography system into urban answers at a fundamental level. In the occasion of any facts breach, none of this information will be disclosed.

Operational vulnerabilities for smart cities

Data in smart city applications can be capable of resist amendment, tampering, verification, valid get admission to, disclosure and destruction. Key protection and privateness requirements encompass confidentiality, integrity, availability, non-repudiation, access manage, and privacy (Dewi Rosadi et al. 2018). Privacy is a primary problem in clever towns and can be immediately linked to minimal cognizance of privacy by nearby governments and businesses in how they acquire and procedure non-public facts. They frequently do no longer provide the community with an possibility and consensus mechanism (Dewi Rosadi et al. 2018).

Use and acceptance by citizens of customer services (a success customer services)

Several studies highlight the significance of residents' perceived security and privateness of urban services (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). Perceived security and privateness have been discovered to seriously influence the use and recognition of clever offerings by means of citizens. Studies vary within the level of privateness relying on the kind of technology, statistics usage and vicinity. According to van Zoonen (2016), human beings in clever cities have 4 regions of problem, which range from low (anonymized statistics, purpose of carrier) to very excessive (personal records, surveillance). Research examines how positive technology (smart shopping carts, clever motors) and facts utilization (predictive police, social media tracking) can cause numerous privacy troubles.

IV. REQUIREMENT ANALYSIS

A. EXISTING SYSTEM:

- In the modern-day machine, someone who visits a sure town should collect from a person who's inside the metropolis or take the assist of a frontrunner within the city. Gather all of the statistics you need to go to the city. It's time smart and it receives ahead of time. In order to acquire each piece of records, we want to contact assist.

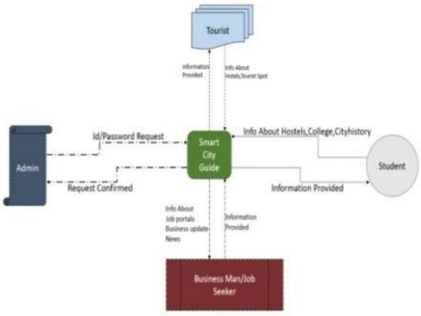
B. LIMITATIONS OF THE EXISTENCE OF THE SYSTEM:

- The current gadget is a manual machine. Here, the town records need to store its facts inside the shape of Excel spreadsheets or disk drives.
- • Wool isn't feasible if the information is presented on paper or on discs.
- • The manual machine offers us less security for stored records; some data can be lost from harm.
- • This device is restrained and much less person pleasant.

C. SYSTEM PROPOSAL:

- The proposed gadget offers online information about the styles of cities you'll visit. It additionally presents extra services to the consumer. The development of this new system includes the following sports that automate the complete manner with database integration in mind.
- • The application offers ease of use with numerous controls supplied with the aid of the Rich User interface.
- • The device makes standard mission control a lot less difficult and more flexible.
- • The intranet can be accessed via
- • City statistics files may be stored in a centralized database that can be saved all through the device

D. SYSTEM ARCHITECTURE

- 

E. SYSTEM REQUIREMENTS

1. HARDWARE REQUIREMENTS
   - System - Pentium-IV
   - Speed - 2.4GHZ
   - Hard disk - 40GB
   - RAM - 512MB

2. SOFTWARE REQUIREMENTS
   - Operating System - Windows XP
   - Coding language - Java
   - IDE – Android studio

3. MODULES
   - Administrator Module
   - Tourist Module
   - Jobseeker Module
   - Businessman Module
   - Student module

4. MODULES DESCRIPTION

Administrator Module

Admin module is the principle module that is accountable for doing simple operations associated with web page updates, product updates, service signals, and so forth. It shops information approximately the opposite four modules. Various programs inside the admin module are update alerts, enterprise updates, motel updates, resume view, site information replace. The info entered by means of the administrator consist of the entire history of the city, which includes politicians, political leaders and social networks, every enterprise which includes the pinnacle businesses inside the town and their statistics, every task which includes openings and the enterprise's activity profile, cases consisting of cases. Smartphone wide variety Common locations consisting of description, region, deal with and photo of the area, information, newspapers, town channels which includes what is to be had within the metropolis newspapers, records of students which includes faculties, establishments, dormitories, libraries ...., tourist information consisting of like traveller places, accommodations, maps, cheap books. Data entered by way of the administrator also can edit the information after it's been entered.

Tourist Module

The tourist module continues data about the location of the town, hotels of the city, accommodation on this city, etc., the consumer after being registered in the traveller is considered an regular consumer. Various programs inside the tourism module: theater view, motel view, town map view, device region view, health facility view, city history history, travel view, avenue experience angle.

Jobseeker Module

The Jobseeker module is a industrial module that requires some paid offerings. The searcher is going thru the office gate and finds the job. It then sends the e-mail to the administrator. Various software program packages are to be had in the task seek module: View Jobs, View City Information, View Institutions, View City Updates, Post Resume, View Alerts, View City Location.

Businessman Module

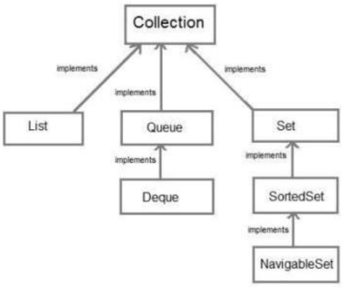
The commercial enterprise module consists of information approximately the numerous corporations of the town, the industry within the town, and the social and political impact of the city. Various commercial enterprise software program modules: metropolis records view, marketplace view, warning view, enterprise view, resort view, resort view, motel view, staff, process emptiness view.

Student Module

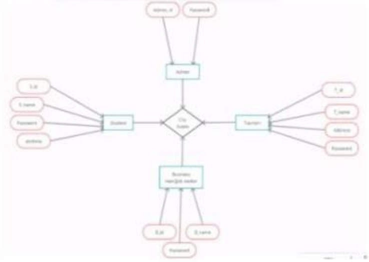
The pupil module incorporates various facts approximately numerous institutions inside the city. Various programs: library view, book view, institute view, engineering university view, take a look at center view, magazine view.

V. COLLECTION FRAMEWORK

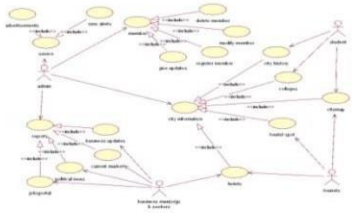
The series device turned into no longer included within the original launch of Java. Collections were brought in J2SE 1.2. Prior to Java 2, Java furnished special classes such as Dictionary, Vector, Stack, and Properties to keep and prepare businesses of items. The collection framework affords many big classes and mechanisms for aggregating and organizing agencies of comparable items.



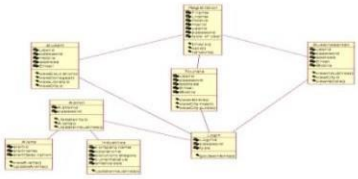
Data flow diagram



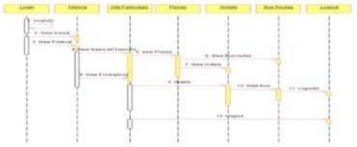
USECASE DIAGRAM



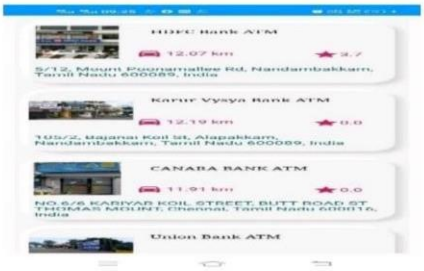
Class diagram



Sequence diagram



VI. SCREEN SHOTS



HDFC Bank ATM

12.07 km ★ 2.7

5/12, Mount Poonamallee Rd, Nandambakkam, Tamil Nadu 600089, India

Karur Vysya Bank ATM

12.19 km ★ 0.0

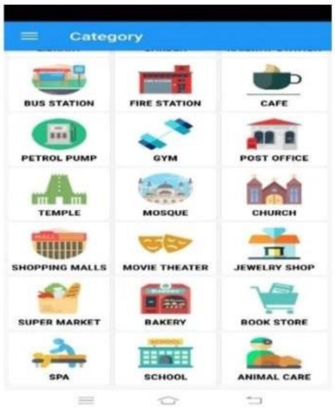
105/2, Bajanai Koil St, Alapakkam, Nandambakkam, Tamil Nadu 600089, India

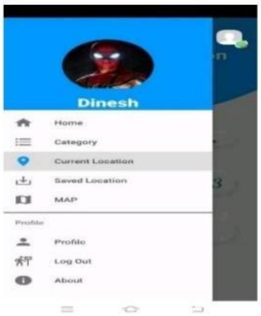
CANARA BANK ATM

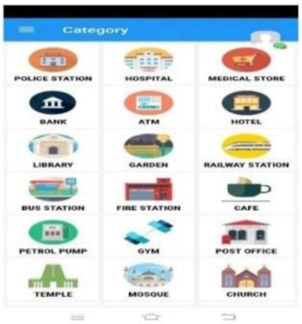
11.91 km ★ 0.0

NO.6/6 KARIYAR KOIL STREET, BUTT ROAD ST THOMAS MOUNT, Chennai, Tamil Nadu 600016, India

Union Bank ATM







VII. CONCLUSION

This Java venture offers facts approximately diverse elements of the metropolis, along with tourism, institutions, industry, maps, place gadgets, and so on. Implementing this method solves most of the problems that a brand new traveller will face whilst coming to a brand new metropolis, as an example: finding a direction, seeking out a lodge, booking tickets and extra.

REFERENCES

- Li et al. (2019) Privacy and Security of Mobile Devices and Services
- Abosaq et al (2019) Smart City Infrastructure
- Ainane et al. (2018); Smart City Infrastructure
- Chatterjee et al. 2017 Smart City Infrastructure
- Ferraz et al 2014 Smart City Infrastructure
- Alamaniotis et al. 2017 Smart Power Systempresented
- Sanduleac et al 2016 Smart Power System
- Alromaihi et al. 2018 Smart Power System
- Beltran et al. 2017 Protocols to Improve Security and Privacy
- Stromire et al (2018) Protocols to Improve Security and Privacy
- Dewi Rosadi et al. 2018 Operational Vulnerabilities for Smart Cities
- Belanche-Gracia et al. 2015; Use and Adoption of Smart Services by Citizens