KOVAI SPOT BUS – BUS RECOMMENDATION SYSTEM

PROJECT REPORT

Submitted by

NIRANJAN E	**[711719104058]**
SASI KEERTHANA R	**[711719104083]**
SASI KUMAR	**[711719104084]**
THARINI M	**[711719104101]**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

KGiSL INSTITUTE OF TECHNOLOGY, COIMBATORE

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2023

ANNA UNIVERSITY: CHENNAI-600 025

BONAFIDE CERTIFICATE

Certified that this project report **"KOVAI SPOT BUS – BUS RECOMMENDATION SYSTEM"** is the bonafide work of **"NIRANJAN E, SASI KEERTHANA R, SASI KUMAR P, THARINI M"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. THENMOZHI T	**Ms. NAMBI RAJESWARI G**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
PROFESSOR	ASSISTANT PROFESSOR
Computer Science and Engineering	Computer Science and Engineering
KGiSL Institute of Technology	KGiSL Institute of Technology
Coimbatore-641035	Coimbatore-641035

Submitted for the Anna University Viva-Voce examination held on ______________

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

We express our deepest gratitude to our **Chairman and Managing Director Dr. Ashok Bakthavachalam** for providing us with an environment to complete our project successfully.

We are grateful to our CEO-Academic Initiatives **Mr. Aravind Kumar Rajendran**, our beloved Director-Academics **Dr. P. Shankar** and our honorable Secretary **Dr. N. Rajkumar**. We are thankful to our beloved Principal **Dr. M. Selvam** and Vice-Principal **Dr. S. Suresh Kumar** for their support and their valuable guidance and blessings.

We would like to thank **Dr. T. Thenmozhi, M.E., M.B.A., Ph.D.,** Head of the Department, Department of Computer Science and Engineering for his unwavering support during the entire course of this project work. We express our sincere thanks to **Ms. ARUNA T N M.E.**, our Project Coordinator, Professor, Department of Computer Science and Engineering who modelled us both technically and morally for achieving greater success in completing this project work.

We express our sincere thanks to our faculty guide **Ms. NAMBI RAJESWARI G M.E.,** Assistant Professor, Department of Computer Science and Engineering for her constant encouragement and support throughout our course, especially for the useful suggestions given during the course of the project period and being instrumental in the completion of our project with her complete guidance.

We also thank all the **Faculty Members** of our department and finally, we take this opportunity to extend our deep appreciation to our **Family** and **Friends**, for all they meant to us during the crucial times of the completion of our project.

ABSTRACT

Kovai spot bus is a real time project which is useful for the people who is the facing problems with the current manual work of bus searching. Kovai spot bus system contains all bus with their route details. The main purpose of this software is to reduce the manual errors involved in the travel process and make it convenient for the people to manage their bus details. This project has been made simpler and interactive. This is an android application used to find out the bus number from one place to another place. User need to give the details of source and destination. Accordingly, it will display the details of the bus number which is going in that route. It is a time saving application to user. User can easily get the information of the bus number of a particular route. It will also be very helpful for those people who are new to the city. In this application user can view the bus location will update the location of a bus on server frequently. The Coordinates of the location will be sent to the server, and the server will send the current location of the bus to people. This application informs bus schedules and their route details for public people. This application reduces the manual errors involved in travel process and makes it convenient to the people. User need to give the details of source and destination accordingly it will display the bus number and routes. Users are able to search the arrival of buses with accurate information. This application reduces the waiting time of people for the buses and improving overall public transportation efficiency.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

HTML	Hypertext Markup Language
CSS	Cascading style sheet
SRS	Software Requirements Specifications
SQL	Structured Query Language
GPS	Global Positioning System
API	Application Programming Interface

CHAPTER 1

INTRODUCTION

Kovai Spot Bus is a real time project which is useful for the people who is the facing problems with the current manual work of bus searching. Kovai Spot Bus contains all bus with their route details. The main purpose of this software is to reduce the manual errors involved in the travel process and make it convenient for the people to manage their bus details. This project has been made simpler and interactive. This is an android application used to find out the bus number from one place to another place. User need to give the details of source and destination. Accordingly, it will display the details of the bus number which is going in that route. It is a time saving application to user. User can easily get the information of the bus number of a particular route.

1.1 PROBLEM DEFINITION

- Kovai Spot Bus is an android application that informs bus schedules and their route details for public people.
- It will display the bus number and routes.
- This application reduces the waiting time of people for the buses and improving overall public transportation efficiency.

1.2 OBJECTIVE OF THE PROJECT

The objective of this project is to contribute to develop a android application that informs bus route and number for the public people. The main objectives of the android application can be defined as follows:

- The usage of this application greatly reduces the time required to search for a place.
- The application also provides searching facility for new peoples.
- Easy to search and less time consuming.
- This application reduces the waiting time of people for the buses and improving overall public transportation efficiency.

1.2.1 Scope

This research study covers the four basic operations of bus route, admin registration, bus number and timings and Location updates. This can involve collaborating with the local transportation authorities, bus operators, or utilizing existing data sources. The interface should be user-friendly and accessible across different platforms. Implement measures to handle any discrepancies, delays, or changes in schedules, and provide mechanisms for users to report issues or provide feedback.

1.2.2 Limitations

The bus service may have limited coverage and may not operate in all areas or neighborhoods of the city. This can be a limitation for passengers who reside or need to travel to areas not served by the Kovai Spot Bus. Bus services typically operate on fixed routes and schedules, which may not align with the specific needs or preferences of all passengers. This can be inconvenient for individuals with unique travel requirements or those looking for flexible transportation options.

1.3 ASSUMPTIONS AND HYPOTHESIS

The researchers assume the following assumptions:

- Internet connectivity is needed for the Kovai Spot Bus – Bus recommendation system. Internet speed may affect the perception of the systems users with regards to the system effectiveness and efficiency.
- Bus Location updating may increase the efficiency of the application and also increase the users.

The researchers identify the following hypotheses:

- There is a significant difference in the Manual monitoring and suggestions of bus timing and Kovai Spot Bus android application.
- It increases local transportation efficiency using Kovai spot bus android application while there is an increased risk when using manual-based one.

1.4 SIGNIFICANCE OF PROBLEM

The findings of this study will benefit passengers for making the transportation efficiency. problems faced by a bus service can have various levels of significance depending on their impact on passengers, operational efficiency, and overall service quality. Here are some potential areas where problems could have significance.

The main advantages of the system are:

- Delays and unpredictability can disrupt passengers' daily routines and erode trust in the service.
- Accessibility and Coverage: The significance of problems related to accessibility and coverage depends on the extent to which the Kovai Spot Bus service meets the transportation needs of the community.
- Any issues related to passenger safety, security, or comfort are of utmost significance.

CHAPTER 2

LITERATURE REVIEW

Kovai Spot Bus has its own database which makes independent of internet and its displays the bus number and bus timings. According to Amrita P Unnithan and Asiya Abu in their study entitled “Active Bus Tracking & Bus Recommendation System”, buses are an important part of the public transport system. Traveling by the bus is an option but in most cases the chance of a bus getting delayed is high, this can be of various reasons like heavy traffic, engine problems. The objective of this project is to provide the real-time locations, routes of travel, also recommending bus to a particular destination and best travel options, that ensures available bus routes, available buses to a destination, provide bus timing to reach the bus stop and required time to reach the destination. IoT is used to provide real-time bus location and machine learning to build recommendation engine. The recommendation based on most scored, reviewed, budget, travel comments, similar travels. Cloud-based database is used for storing and manipulating the data. Database stores the location along with the update time.

Updating time and location can be used to calculate the time taken to travel and average speed. If the bus is not running on that day, the proper indication is displayed along with the location and the timing of the next bus through the route will be given. Machine learning server uses data in the database to recommend the best route to travel and creates a model for future use. The user interface provided by using an android application. This proposed system saves time and increases the work efficiency of end-users because it reduces the user’s efforts to traveling for work and avoids the wastage of waiting time for the bus.

The public transportation system plays a major role in daily routine. Sometimes there are lots of problems arising because of the lack of information about the availability of buses in a route at a time. Lots of technologies can be applied to transportation systems, especially in buses, but they don’t run according to the predefined timetable as described due to traffic jams, breakdown, engine problems, etc. The bus corporations provide bus timetables on the websites, but such bus timetables are usually static and provide only very limited information to the user.

On the other hand, According to M.A Hannan, A.M. Mustapha, A.Hussain and H.Basri entitled that “Intelligent Bus Monitoring Management System” states Intelligent bus monitoring system based on current challenges and problems. In this system, radio frequency identification(RFID) and integrated sensing technologies such as global positioning system (GPS), general packet radio service (GPRS) and geographic information system (GIS) are used to monitor the movement of a bus. A new theoretical framework and ruled based decision algorithms are developed for the system. An experimental setup is developed for the prototype implementation. The results show that the choice of integrated technologies used in the system is suitable to monitor and manage a vehicle transportation system.

RFID is a wireless identification technology that has been used in many fields, including solid waste bin monitoring, animal, goods, and object tracking, and in street trees management. Past researchers have proven that the implementation of RFID in any identification and monitoring system can improve the overall performance of the system at affordable prices. For that reason, RFID is chosen as one of the technology implemented in the bus monitoring system. Along with RFID, other sensing technologies such as a GPS, GPRS, and GIS can be used in a monitoring system. GPS, GPRS, and GIS have been integrated together in various studies, and the good results demonstrate that the technologies are compatible. From the reviews, RFID, GPS, GPRS, and GIS are chosen to be integrated and tested in the realization of a bus monitoring and management system.

CHAPTER 3

SYSTEM ANALYSIS

System analysis is a problem-solving technique that decay a system into component pieces of purpose of studying how well those component parts work and interact to accomplish their purpose. The following chapter provides a detailed description of the existing system. It also provides an overview of the proposed system and feasibility of Kovai spot bus.

3.1 EXISTING SYSTEM

The Existing web-based application for the users who want real time information about bus routes in the city. There are some route finders and bus number finding systems in web-based application and that is money consuming. If the net is slow then the application will be slow. If the internet connectivity is weak then the process of finding the buses will also difficult. Sometimes the information provided by these systems may not be accurate due to delays or changes in the bus schedule. This can lead to confusion and inconvenience for users. Some systems may not provide real-time updates on bus timings, which can be a problem for users who need to plan their travel. To overcome these problems, we have developed Kovai spot bus that works with mobile.

3.1.1 DISADVANTAGES

- Route details are advertised through bus stand notice board.
- Online systems may be vulnerable to security breaches, such as hacking or phishing attacks.
- Time consuming is high to, storing and updating the data.
- Some existing systems may charge users for access to certain features or services.
- Details are enquired through phone

3.2 PROPOSED SYSTEM

Kovai spot bus application is an android application people can use on their mobile. User need to give source and destination and it displays the bus timing and bus number accordingly. This application has its own database which makes independent of internet and its displays the bus number and bus timings. Kovai spot bus makes finding of buses easy, fast and efficient. This system presents an adaptive location information sharing that allows for a user to know the exact location of a bus. The people no need to wait for a long to catch the bus. The bus location updated on the server. The web service receives the location information of a bus continuously.

In this kovai spot bus application, users can easily access updated information of bus timing and bus number. If there are any changes in schedules, admin can update the schedules. Users can easily search for bus routes, stops and view information about the next bus arrival time and delays or changes to the schedule. This application improves overall public transportation efficiency

3.2.1 ADVANTAGES

- Kovai spot bus provides real-time information on the location and estimated time of arrival of the buses. This information can be accessed by the users through the mobile application and information display boards, which help them plan their journey.
- The users can get the information on their smart phones without having to visit the bus stops or wait for the buses to arrive.
- The real-time information on bus timings and locations can help commuters reduce their waiting time at bus stops.
- It is very flexible and user-friendly and Helpful.
- This application also provides searching facilities for new people and it is easy to search.

3.3 FEASIBILITY STUDY

The feasibility study is performed to determine whether the proposed system is viable considering the Technical, Operational and Economical factors. After going through feasibility study, we can have a clear-cut view of the system’s benefits and drawbacks.

3.3.1 Tests of Feasibility

Feasibility study is conducted once the problem is clearly understood. Feasibility study is necessary to determine that the proposed system in Kovai spot bus is feasible by considering the technical, operational, and economical factors. By having a detailed feasibility study the management in the will have a clear-cut view of the proposed system of the cruor collection system. Feasibility study encompasses the following things:

- Technical Feasibility
- Economical Feasibility
- Operational feasibility

3.3.1.1 Technical Feasibility

The Kovai Spot Bus system requires a GPS tracking system to be installed on all the buses in Coimbatore, and a centralized server to process the data and provide real-time information to users. The technology for GPS tracking and mobile applications is widely available and proven, making the technical feasibility of the system high.

3.3.1.2 Operational Feasibility

Operational feasibility is dependent on human resources available for the project involves projecting whether the system will be used if it is developed and implemented. The Kovai Spot Bus system has the potential to improve the overall quality of the bus service in Coimbatore and make it more convenient for commuters.

3.3.1.3 Economical Feasibility

The cost of implementing the system will depend on factors such as the number of buses, the cost of installing GPS tracking systems, and the development cost of the mobile application. The system's economic feasibility will depend on the benefits it provides to commuters, such as reduced waiting times and increased convenience.

CHAPTER 4
SYSTEM SPECIFICATION

4.1 FUNCTIONAL REQUIREMENTS

4.1.1 User registration and Login

- Users will need to register with the Kovai Spot Bus application by providing basic information such as their name, email address, and phone number.
- Once registered, users can log in to the application using their email address and password. They can also choose to use social media accounts such as Facebook or Google to log in.

4.1.2 Search Route

- The search route functionality should allow users to input their origin and destination points in a user-friendly manner.
- The system should use a search algorithm that considers the user input and matches it with the available bus routes and timings.
- The search route functionality should use real-time data from the GPS tracking system to provide accurate information on bus timings and locations.

4.1.3 Login

- The bus information functionality displays the real-time location, estimated time of arrival, and bus number of all buses on the mobile application and information display boards at bus stops.
- The bus information display should have a user-friendly interface that is easy to understand and navigate. The display should use clear and concise.

4.1.4 Administrative system

- Real-time Monitoring: The administrative system should provide real-time monitoring of bus locations, arrival times, and route adherence.

- User Management: The administrative system should provide user management features that allow administrators to manage user accounts, including registration, login, and access control.

4.2 NON-FUNCTIONAL REQUIREMENTS

4.2.1 User Interface

- The user interface should be designed with intuitive navigation, allowing users to easily find the information they need and access the features they want to use.
- The information should be updated in real-time and should be accurate.
- The system shall maintain an easy to use interface across all functionality and for all users.

4.2.2 Scalability

Kovai Spot Bus can be made more scalable by integrating with other systems and services using APIs (Application Programming Interfaces). This allows for greater flexibility and interoperability, as well as the ability to handle more traffic and data.

4.2.3 Security

- The system must have a secure authentication and authorization mechanism to ensure that only authorized users can access the system.
- Sensitive data should be encrypted both in transit and at rest to prevent unauthorized access.
- The system should be subject to regular security audits to identify vulnerabilities and ensure that the system is up-to-date with the latest protocols.

4.2.4 Portability

- The system should be written in a platform-independent language and designed to run on any operating system or platform.
- The system should be designed with APIs (Application Programming Interface.

4.2.5 Maintainability

- This system is easy to update, modify, and fix, making it more reliable and cost effective.
- There should be a clear separation between the data access object that map the database and the business logic code.

4.2.6 Exception Handling

Exceptions should be reported effectively to the user if they occur.

4.2.7 Ethics

The system shall not store or process any information about its users.

4.3 HARDWARE REQUIREMENTS

Processor	:	i3
Processor Speed	:	3.00 GHZ
RAM	:	4 GB
Hard Disk	:	500 GB
Monitor	:	16’’ Color Monitor
Keyboard	:	Standard 110 keys
Pointing Device	:	Mouse
Smart Phone	:	Any type

4.4 SOFTWARE REQUIREMENTS

Operating System	:	Windows/Linux/Visual studio
Front End	:	Android Eclipse
Back End	:	SQL server, java
Web Service	:	Visual Studio

CHAPTER 5
SOFTWARE DESCRIPTION

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide. Software requirements specification establishes the basis for an agreement of the bus information and on route of the local bus service what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules.

5.1 FRONT END

The front end is designed using framework which includes a collaborative platform for admin and the passengers. It is the collaborative end to end platform made by developers. Here all the web tools are integrated and it allows interaction between the client and the user interface. It uses java and sql for backend functions. Thia application become more easy because of android platform.

5.1.1 Android Eclipse

Python Eclipse is an integrated development environment (IDE) used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment. It is the second-most-popular IDE for Java development. Eclipse is written mostly in Java and its primary use is for developing Java applications

5.1.1.1 Features

- API.
- Code Development.
- Code Editing.

- Collaboration Tools.
- Data Modeling.
- Debugging.
- Deployment Management.
- For Developers.

5.1.1.2 Advantanges

- Open source and community development
- Portable and Interactive
- Ideal for prototypes – provide more functionality with less coding
- Highly Efficient
- allows setting breakpoints
- automatically validates syntax
- offers a robust debugger
- provides you readymade code template
- robust Java editor
- supports code refactoring
- supports syntax coloring.

5.2 BACK END

The back end is designed using MySQL, whose primary function is to store data securely and retrieve it later, as requested by other software applications.

5.2.1 MySQL

MySQL is an open-source relational database management system (RDBMS). The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality.

MySQL is a central component of the LAMP open-source web application software stack (and other "AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python". Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, Simple Machines Forum, phpBB, MyBB, and Drupal.

5.2.2 Features of MySQL

- A broad subset of ANSI SQL 99, as well as extensions.
- Cross-platform support.
- Stored procedures, using a procedural language that closely adheres to SQL/PSM.
- Triggers and Cursors.
- Updatable views.
- Online DDL when using the InnoDB Storage Engine.
- Information schema.

5.2.3 Advantages of MySQL

- Data Security
- On-Demand Scalability
- High Performance
- Round-the-Clock Uptime
- Comprehensive Transactional Support

CHAPTER 6

PROJECT DESCRIPTION

Kovai spot bus is a mobile application that helps users find the best routes and schedules for buses in their area. The app uses GPS technology and real-time data to provide accurate information on bus locations, arrival times, and delays.

In search for routes, Users can search for routes to their desired destination by entering their current location and destination address. In real-time bus tracking, the app provides real-time information on the location of buses, so users can see when the next bus is expected to arrive and how long it will take to reach their destination. In dule information, Users can view the schedules for buses, including departure and arrival times, and plan their trips accordingly.

6.1 OVERVIEW OF THE PROJECT

Kovai Spot Bus is a complete set of methods in order to display Bus number and timing. It uses GPS Technology to update the location.

6.2 MODULE DESCRIPTION

6.2.1 Admin Authentication

This module has been used for admin. Admin can enter the route details. Admin need to login with valid login credentials. Admin can add bus with route details.

6.2.2 Add Route Information

Route information module specifies the bus number, route name, starting place, time, ending place of all the buses added by admin.

6.2.3 Search Route

Users can select sources and destinations using the mobile app. This information getting from the database. it's used for user searching routes. This module contains results getting from the user selecting source and destination. its shows the result bus details like bus no, route and time getting information from database system.

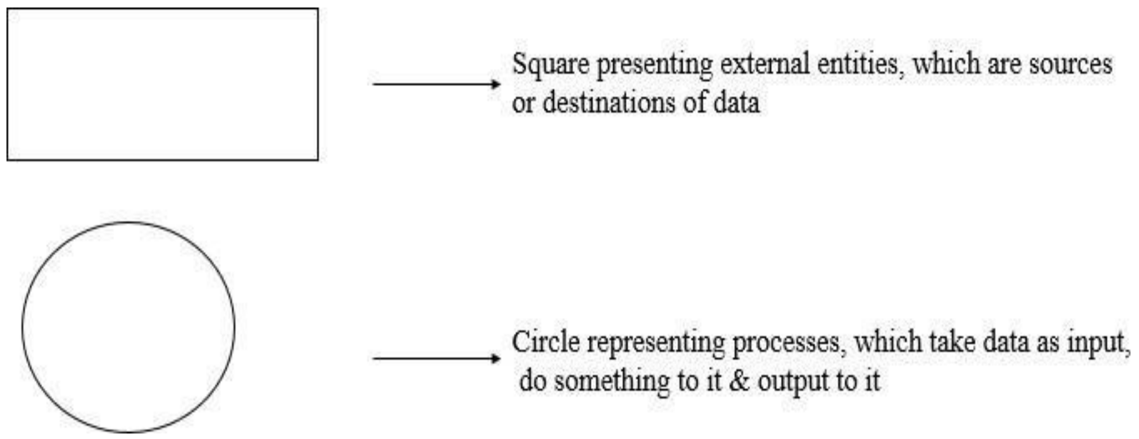
6.2.4 Vehicle Location Update

While the people need the current location of a specific bus, the user can view the vehicle information using their mobile device. The vehicle number as the request which is passed to the web service and the web service fetches the current location of a requested vehicle and sends it to the user mobile device.

6.3 DATA FLOW DIAGRAM

Data flow diagram is used to describe how the information is processed and stored and identifies how the information flows through the processes. Data flow diagram illustrates how the data is processed by a system in terms of inputs and outputs. The data flow diagram also depicts the flow of the process and it has various levels. The initial level is context level which describes the entire system functionality and the next level describes each and every sub module in the main system as a separate process or describes all the process involved in the system separately.

Data flow diagram are made up of number of symbols,



Square presenting external entities, which are sources or destinations of data

Circle representing processes, which take data as input, do something to it & output to it

Arrows representing the data flows, which can either, be electronic data or physical items.

Parallel lines representing data stores, including electronic stores such as databases or XML files and physical stores

6.3.1 DFD LEVEL 0

DFD Level 0 is a basic overview of whole system or process being analyzed or modeled. Its designed to be an at-a-glance view.

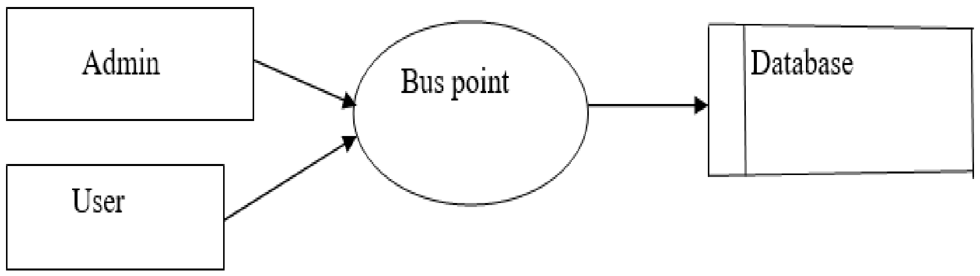


Fig 6.3.1 DFD Level 0

6.3.2 DFD LEVEL 1

A level 1 DFD notates each of the main sub-processes that together form the complete System. We can think of a level 1 DFD as an “exploded view” of the context diagram.

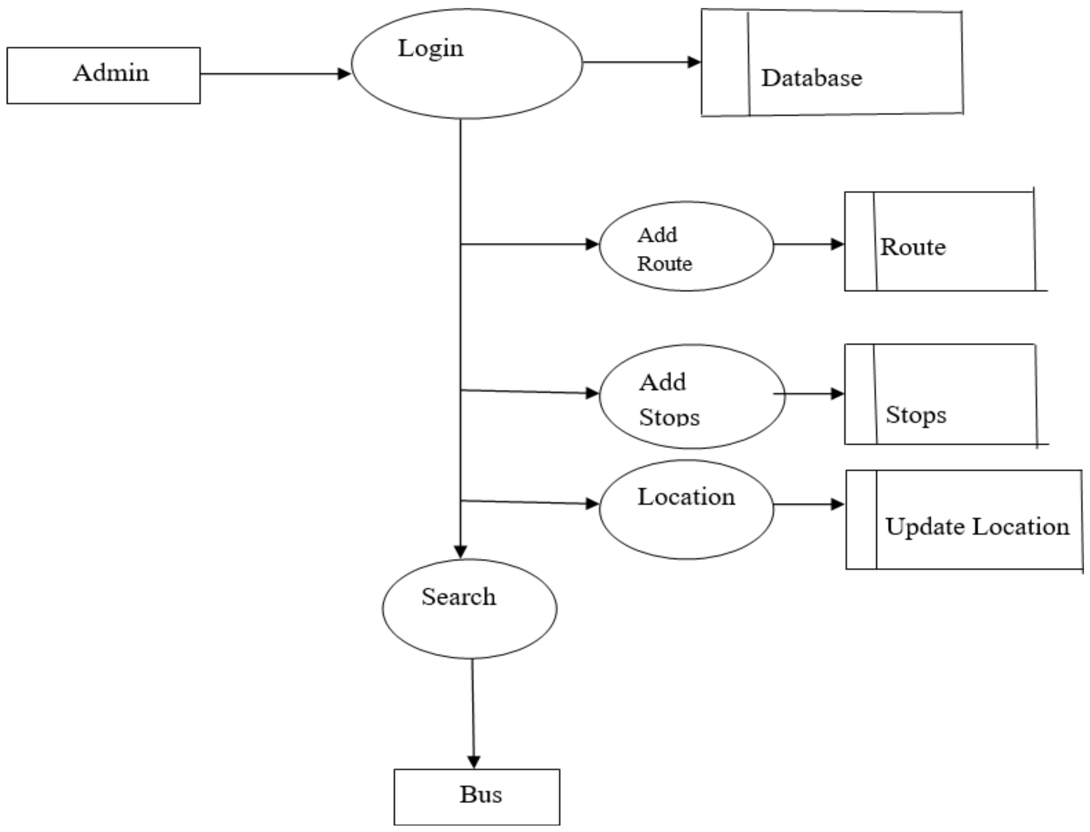


Fig 6.3.2 DFD Level 1

6.3.3 DFD LEVEL 2

There can be a level 2 DFD for each process that appears in the level 1 DFD.

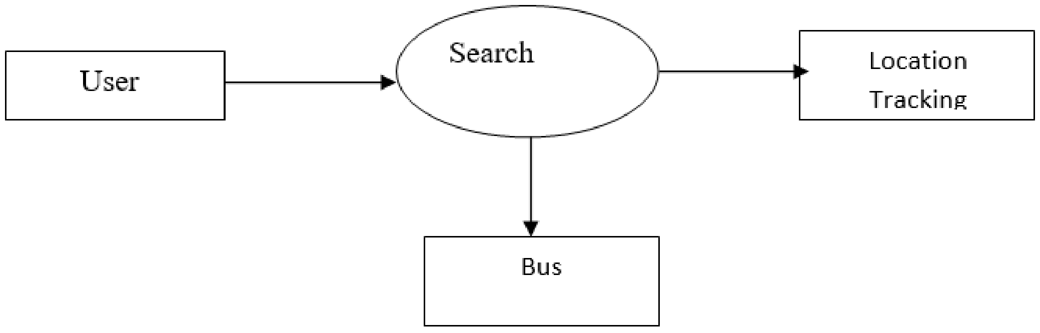


Fig 6.3.3 DFD Level 2

6.4 DATABASE DESIGN

Database design is the process of producing a detailed data model of database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then used to create a database.

1. User Table:
   - User ID (primary key)
   - Name
   - Email
   - Password
2. Bus Table:
   - Bus ID (primary key)
   - Bus Number
   - Source & Destination
   - Departure Time
   - Arrival Time

In this design, the User table stores information about registered users, including their unique User ID, name, email, and password. The Bus table contains information about all available buses, such as the Bus ID, bus number, source, destination, departure time,arrival time, total seats, available seats, and price.

The Bus Booking table links users and buses, storing information about each booking, including the Booking ID, User ID, Bus ID, booking date, and seat number. This database design allows the app to efficiently store and retrieve information about users,buses, and bookings. With this design, the app can quickly find all available buses, check the availability of seats, and create bookings for users. It also provides a way to link users and bookings to the relevant bus information, making it easier to manage and track bookings.

6.5 INPUT DESIGN

The input design is the link between the information system and the user. 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.

Here are some input design considerations for a bus searching app:

1. User Input:
   - User registration: Collect user information such as first name, last name,

email and password.

- Search parameters: Allow users to search for buses based on parameters such as source,destination, departure time and price range.

2. Data Validation:
   - Validate user input fields such as email and phone number to ensure that they are the correct format.
   - Check that the user's preferred travel dates and times are valid, and that theselected bus has available seats.
3. User Experience:
   - Use clear and concise language in input fields to make it easy for users to understand what information is required.
   - Provide clear feedback messages if input fields are left empty or if the entered information is not valid.

By implementing these input design considerations, a bus searching app can provide a user-friendly interface that helps users find and book their preferred bus quickly and easily while maintaining the security and privacy of user data.

6.6 OUTPUT DESIGN

Output design generally refers to the results and information that are generated by the system for many end-users; output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application. The output is designed in such a way that it is attractive, convenient and informative. Forms are designed in C#.NET with various features, which make the console output more pleasing.

As the outputs are the most important sources of information to the users, better design should improve the system's relationships with us and also will help in decision-making. Form design elaborates the way output is presented and the layout available for capturing information.

- Search
- Location Tracking

CHAPTER 7
SYSTEM TESTING

One of the most important phases in the development of any software is testing. Testing is a dynamic technique of verification and validation, which ensures that the software meets the expectations of the user. It involves executing an implementation of the software with test data. Test data is the set of data that the system will process as a normal input. A test case is a set of sequential steps to execute a test, operating on a set of predefined inputs to produce certain expected outputs. Testing demonstrates the software functions work according to specifications. In addition data collected from testing provides a good indication of software reliability and quality. Testing results in the deduction of number of errors. In this project critical modules are tested. The fields are tested for various validation methods using the different testing techniques.

7.1 TESTING METHODS

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.
- Features to be tested
- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

7.2 TYPES OF TESTING

7.2.1 Unit Testing

This is the first level of testing. The different modules are tested against the specifications produced during the integration. This is done to test the internal logic of each module. Those resulting from the interaction between modules are initially avoided. The input received and output generated is also tested to see whether it falls in the expected range of values. Unit testing is performed from the bottom up, starting with the smallest and lowest modules and proceeding one at a time. The units in a system are the modules and routines that are assembled and integrated to perform a specific function. The programs are tested for correctness of logic applied and detection of errors in coding. Each of the modules was tested and errors are rectified. They were then found to function properly.

7.2.2 Integration Testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the tested modules are combined into sub- systems, which are then tested. The goal of integration testing to check whether the modules can be integrated properly emphasizing on the interfaces between modules. The different modules were linked together and integration testing done on them.

7.2.3 Validation Testing

The objective of the validation test is to tell the user about the validity and reliability of the system. It verifies whether the system operates as specified and the integrity of important data is maintained. User motivation is very important for the successful performance of the system. All the modules were tested individually using both test data and live data. After each module was ascertained that it was working correctly and it had been "integrated" with the system. Again the system was tested as a whole. We hold the system tested with different types of users. The System Design, Data Flow Diagrams, procedures etc. were well documented so that the system can be easily maintained and upgraded by any computer professional at a later.

7.2.4 System Testing

The integration of each module in the system is checked during this level of testing. The objective of system testing is to check if the software meets its requirements. System testing is done to uncover errors that were not found in earlier tests. This includes forced system failures and validation of total system as the user in the operational environment implements it. Under this testing, low volumes of transactions are generally based on live data. This volume is increased until the maximum level for each transactions type is reached. The total system is also tested for recovery after various major failures to ensure that no data are lost during the breakdown.

7.2.5 User Acceptance Testing

The objective of the acceptance test is to tell the user about the validity and reliability of the system. It verifies whether the system operates as specified and the integrity of important data is maintained. User motivation is very important for the successful performance of the system. All the modules were tested individually using both test data and live data. After each module was ascertained that it was working correctly and it had been "integrated" with the system. Again the system was tested as a whole. We hold the system tested with different types of users. The System Design, Data Flow Diagrams, procedures etc. were well documented so that the system can be easily maintained and upgraded by any computer professional at a later

7.2.6 White Box Testing

White Box Testing is the testing of a software solution's internal coding and infrastructure. It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability. White box testing is also known as Clear Box testing, Open Box testing, Structural testing, Transparent Box testing, Code-Based testing, and Glass Box are testing. It is one of two parts of the "box testing" approach of software testing.

Its counter-part, black box testing, involves testing is based on the inner workings of an application and revolves around internal testing. The term "white box" was used because of the see-through box concept. The clear box or white box name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings. Likewise, the "black box" in "black box testing" symbolizes not being able to see the inner workings of the software so that only the end-user experience can be tested.

7.2.7 Black Box Testing

Black box testing is a software testing technique in which functionality of the software under test (SUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software type of testing is based entirely on the software requirements and specifications.

In this Kovai spot bus, the implementation part is been checked for its correctness.

7.2.7.1 Methods of Black Box Testing

There are many types of Black Box Testing but following are the prominent ones

- Functional testing - This black box testing type is related to functional requirements of a system. It is done by software testers.
- Non-functional testing - This type of black box testing is not related to testing of a specific functionality, but non-functional requirements such as performance, scalability, usability.

7.3 TESTING STRATEGY

Implementation is the most crucial stage in achieving a successful system and giving the user's confidence that the new system is effective and workable. Implementation of this project refers to the installation of the package in its real environment to the full satisfaction of the users and operations of the system. Testing is done individually at the time of development using the data and verification is done the way specified in the program specification. In short, implementation constitutes all activities that are required to put an already

Test Strategy is also as test approach defines how testing would be carried out. Test approach has two techniques:

- Proactive - An approach in which the test design process is initiated as early as possible in order to find and fix the defects before the build is created.
- Reactive - An approach in which the testing is not started until after design and coding are completed.

CHAPTER 8
SYSTEM IMPLEMENTATION

This section discusses the implementation details declaring each stage to fulfill the main scenario listed above in system description. The purpose of the implementation of the process is to design and create (or fabricate) a system element conforming to that this is an element's design properties and/or requirements. The element is constructed employing appropriate technologies and industry practices. This process bridges the system definition processes and the integration process. Next sub-section clarifies algorithms implemented in the project based on use-cases declared earlier. The implementation of a bus searching app involves several key steps.

8.1 DEFINE REQUIREMENTS

The first step in implementing the app is to define the requirements based on the needs of the users and the business. This involves identifying the features and functionality that the app must have to meet the needs of the users, as well as any technical requirements or constraints.

8.2 DEVELOP A DESIGN

The next step is to develop a design for the app that incorporates the identified requirements. This includes creating wireframes, mockups, and prototypes to visualize the user interface and user experience.

8.3 CHOOSE A DEVELOPMENT PLATFORM

Once the design is finalized, the development platform must be chosen. This could involve choosing between native development for iOS and Android platforms or using cross-platform tools such as React Native or Xamarin

8.4 BUILD AND TEST THE APPLICATION

The development team can build the application using the chosen platform and programming language, while testing the application to ensure it meets the requirements and functions as expected.

8.5 DEPLOY THE APPLICATION

Once the app has been built and tested, it can be deployed to the app stores for iOS and Android, or distributed through an enterprise app store.

8.6 MONITOR AND MAINTAIN

After deployment, the app must be monitored and maintained to ensure it continues to function as expected. This involves ongoing testing, bug fixing and updating the app to meet changing user needs or technical requirements.

8.7 COLLECT AND ANALYZE DATA

Finally, data should be collected and analyzed to understand how the app is being used, identify areas for improvement, and inform future enhancements.

Overall, the implementation of a bus searching app involves a systematic approach that requires careful planning, design, development, and maintenance to create a successful product that meets the needs of its users.

CHAPTER 9
CONCLUSION & FUTURE ENHANCEMENTS

9.1 CONCLUSION

In conclusion, a Kovai spot bus app can be a useful tool for individuals who rely on public transportation to navigate their daily lives. The application can help users find the most efficient routes and schedules for their desired destination, as well as provide real-time updates on bus locations and delays. Additionally, the application may offer features such as fare information, user reviews, and the ability to save favourite routes for future use. Overall, a well-designed bus searching app has the potential to greatly enhance the user experience and improve the efficiency of public transportation.

This system is user friendly so everyone can use easily. Proper documentation is provided. The end user can easily understand how the whole system is implemented by going through the documentation. The system is tested, implemented and the performance is found to be satisfactory. All necessary output is generated. Thus, the project is completed successfully.

Further enhancements can be made to the application, so that the application functions very attractive and useful manner than the present one. The speed of the transactions become more enough now.

9.2 FUTURE ENHANCEMENT

There are several potential future enhancements that could be implemented for a Kovai spot bus app to improve its functionality and user experience. Some possible examples include:

- Integration with other modes of transportation: To provide a more comprehensive solution for users, the app could be expanded to include other modes of transportation such as trains, subways, and taxis. This would allow users to plan their entire journey from start to finish using a single application.

- Personalized recommendations: By collecting data on users' preferences and past travel patterns, the app could provide personalized recommendations for routes and schedules that are most likely to meet their needs.
- Augmented Reality (AR) features: AR technology could be used to provide users with real-time information about their surroundings, such as bus stop locations and nearby landmarks. This would help users navigate more easily and efficiently.
- Voice-activated assistance: Voice-activated assistance could be integrated into the app, allowing users to access information and perform actions hands-free. This would be particularly useful for users who are visually impaired or have difficulty using a touchscreen.
- Gamification: Gamification features could be added to the app to incentivize users to take public transportation more frequently. For example, users could earn points or rewards for using the app to plan their trips or for taking certain routes during off-peak hours.

Overall, these enhancements could make the bus searching app more useful, engaging and accessible for a wider range of users.

CHAPTER 10

APPENDIX

10.1. SOURCE CODE

```
Package com.example.citybussearching;
import android.os.Bundle;
Import android.os;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;importandroid.view.View;
import android.widget.ArrayAdapter;
importandroid.widget.Spinner;
publicclassSearchextendsActivity{Spinners1,s2;
@Override
protectedvoidonCreate(BundlesavedInstanceState){super.onCreate(savedInstanceState);
setContentView(R.layout.activity_search);
s1=(Spinner)findViewById(R.id.spinner2);s2=(Spinner)findViewById(R.id.spinner3);
Stringgetsub=WebService.viewrouteonly("getsub","routeview");
String[]str1=null;
if (getsub.toString()!="") {str1=getsub.split("#");
ArrayAdapter<String>dataAdapter
 =newArrayAdapter<String>(getApplicationContext(),android.R.layout.simple_spinner_ite
m,str1);
dataAdapter.setDropDownViewResource(R.layout.spinner_item);
s1.setAdapter(dataAdapter);s2.setAdapter(dataAdapter);
}
} @Override
publicbooleanonCreateOptionsMenu(Menumenu)
{
// Inflate the menu; this adds items to the action bar if it is
present.getMenuInflater().inflate(R.menu.search,menu);
 return true;
}
publicvoidgotoview(Viewarg)
{
Intent i = new Intent(this, Viewresult.class);
i.putExtra("source",s1.getSelectedItem().toString());
i.putExtra("destination",s2.getSelectedItem().toString());
startActivity(i);
}
publicvoidgototrack(Viewarg)
{
 Intenti=newIntent(this,VehicleTracking.class);
 startActivity(i);
}
}
packagecom.example.citybussearching;
import android.location.Criteria;
importandroid.location.Location;
import android.location.LocationManager;
importandroid.os.Bundle;
import android.os.Handler;
importandroid.os.StrictMode;
import android.annotation.SuppressLint;
import android.app.Activity;
importandroid.app.Notification;
import android.app.NotificationManager;
importandroid.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
importandroid.widget.TextView;
importandroid.widget.Toast;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
 import java.util.Locale;
 import java.util.TimerTask;
import android.location.Address;
import android.location.Criteria;
import android.location.Geocoder;
import android.location.Location;
import  android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.Handler;
import android.os.StrictMode;
import android.annotation.SuppressLint;
import android.app.Activity;
importandroid.app.Notification;
import android.app.NotificationManager;
importandroid.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.util.Log;
import android.view.Menu;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import  android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
importandroid.location.Location;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.provider.Settings;
import android.view.View;
import android.widget.Button;
importandroid.widget.TextView;
publicclassVehicleTrackingextendsActivity
{
public Buttonb2,b3;
publicStringlogin;
public Spinner s1;
publicEditTexte1;
//publicTextViewtv1;
privateHandlerhandler=newHandler();
```

```
public Strings,result,result1;
StringBuilder strAddress = new StringBuilder();
LocationManagerlocationManager;
Stringprovider;
Stringlati,log;
//AppLocationServiceappLocationService;Handler mHandler=null;
//ArrayListdataModels;
//privateCustomAdapteradapter;Stringdrivname;
@SuppressLint("NewApi")
@Override
protectedvoidonCreate(BundlesavedInstanceState)
{
super.onCreate(savedInstanceState);
setContentView(R.layout.vehicle_tracking);
b2=(Button)findViewById(R.id.exit);
b3=(Butt on)findViewById(R.id.loginbutton);
s1=(Spinner)findViewById(R.id.spinner1);
e1=(EditText)findViewById(R.id.editText1);
publicclassVehicleTrackingextendsActivity
{
public Buttonb2,b3;
publicStringlogin;
public Spinner s1;
publicEditTexte1;
//publicTextViewtv1;
privateHandlerhandler=newHandler();
ArrayAdapter<String>dataAdapter=newArrayAdapter<String>(getApplicationContext(),an
droid.R.layout.simple_spinner_item,str1);
}
```

```
//dataModels=newArrayList();
if(android.os.Build.VERSION.SDK_INT>=9)
{
 StrictMode.ThreadPolicy    policy =
newStrictMode.ThreadPolicy.Builder().permitAll().build();
StrictMode.setThreadPolicy(policy);
}
b2.setOnClickListener(newView.OnClickListener()
{
 @Override
publicvoidonClick(Viewarg)
{
finish();
System.exit(0);
}
});
ArrayAdapter<String>dataAdapter=newArrayAdapter<String>(getApplicationContext(),an
droid.R.layout.simple_spinner_item,str1);
dataAdapter.setDropDownViewResource(R.layout.spinner_item);
s1.setAdapter(dataAdapte  r);
}
private class GeocoderHandler extends Handler
{
@Override publicvoidhandleMessage(Messagemessage)
{
StringlocationAddress;
switch (message.what)
{
case1:
```

```
Bundle bundle = message.getData();
locationAddress = bundle.getString("address");
break;
locationAddress=null;
}
e1.setText(locationAddress);
}
}
publicStringgetMyLocationAddress(doublell,doublelon)
{
String address="";
try
{
Geocodergeocoder;
List<Address>addresses;
geocoder=newGeocoder(this,Locale.getDefault());
addresses=geocoder.getFromLocation(lon,ll,1);
address=addresses.get(0).getAddressLine(0);

@SuppressLint("NewApi")
publicvoidgetlocation(Viewv)
{
if(android.os.Build.VERSION.SDK_INT>=9)
{
StrictMode.ThreadPolicy     policy =         new
StrictMode.ThreadPolicy.Builder().permitAll().build();
StrictMode.setThreadPolicy(policy);
}
```

```
String city = addresses.get(0).getLocality();
Stringstate=addresses.get(0).getAdminArea();
String country = addresses.get(0).getCountryName();
String postalCode = addresses.get(0).getPostalCode();
StringknownName=addresses.get(0).getFeatureName();
//DBWorker(strAddress.toString());
} catch(IOExceptione)
{
// TODO Auto-generated catch blocke.printStackTrace();
Toast.makeText(getApplicationContext(),"Could          not     get     address..!",
Toast.LENGTH_LONG).show();
}
return address;
}
*/
@SuppressLint("NewApi")
publicvoidgetlocation(Viewv)
{
 if(android.os.Build.VERSION.SDK_INT>=9)
{
StrictMode.ThreadPolicy     policy =         new
StrictMode.ThreadPolicy.Builder().permitAll().build();
StrictMode.setThreadPolicy(policy);
}
login=WebService.getlocation(s1.getSelectedItem().toString(),"getlocation");
if(!login.equals("not OK")& !login.equals("")){}
else
{
 e1.setText(login);
Toast.makeText(getApplication(),"No      Data
Found",Toast.LENGTH_SHORT).show();
//e1.setText("NoDataFound");
}
} privateStringgetAddressString(doublelatitude,doublelongitude){StringstrAddress="";
Geocodergeocoder=newGeocoder(this,Locale.getDefault());
try
{
List<Address>addresses=geocoder.getFromLocation(latitude,longitude,1);
if(addresses !=null)
{
//Toast.makeText(getApplication(),addresses.toString(),Toast.LENGTH_SHORT).show();
e1.setText("Latitude:"+latitude+"Longitude:"+longitude+"\n"+address+""+state);
strAddress=strReturnAddress.toString();Log.w("address",
""+strReturnAddress.toString());
}
Else
{
Log.w("address","NoAddress found!");
}
}
catch (Exception e)
{
e.printStackTrace();
Log.w("address","Can'tgetAddress!");
}
 returnstrAddress;
}
 privatefinalRunnablem_Runnable=newRunnable()
```

```
{
 publicvoidrun()
{
//getusers();
}
};
}
// e1.setText(addresses.toString());AddressreturnAddress=addresses.get(0);
StringBuilderstrReturnAddress=newStringBuilder("");
String
address=addresses.get(0).getAddressLine(0);//Ifanyadditionaladdresslinepresentthanonly,ch
eckwithmaxavailableaddresslinesbygetMaxAddressLineIndex()
String city = addresses.get(0).getLocality();Stringstate=addresses.get(0).getAdminArea();
String country = addresses.get(0).getCountryName();String postalCode =
addresses.get(0).getPostalCode();StringknownName=addresses.get(0).getFeatureName();
//OnlyifavailableelsereturnNULL
//for(inti=0;i<returnAddress.getMaxAddressLineIndex(); i++)
{
//strReturnAddress
//.append(returnAddress.getAddressLine(i)).append(
//"\n");
//}
e1.setText("Latitude:"+latitude+"Longitude:"+longitude+"\n"+address+""+state);
strAddress=strReturnAddress.toString();Log.w("address",
""+strReturnAddress.toString());
}
Else
{
Log.w("address","NoAddress found!");
}
```

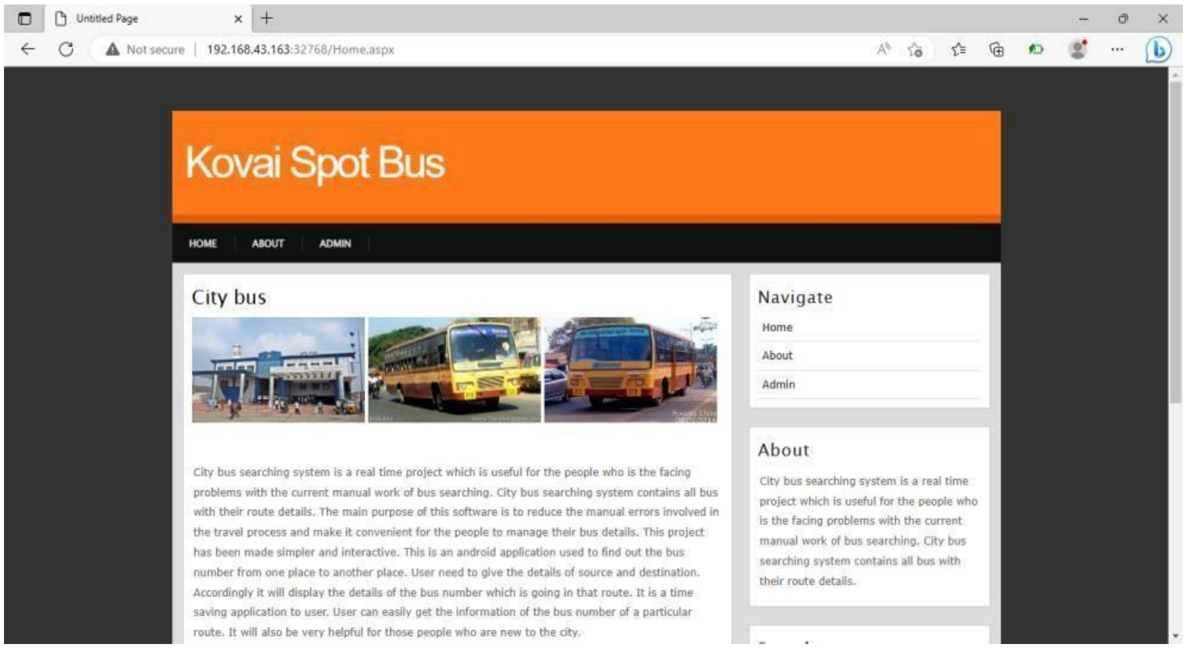
```
}
catch (Exception e)
{
e.printStackTrace();
Log.w("address","Can'tgetAddress!");
}
returnstrAddress;
}
privatefinalRunnablem_Runnable=newRunnable()
{
publicvoidrun()
{
//getusers();
}
};
}
packagecom.example.citybussearching;
importjava.io.IOException;
importjava.net.MalformedURLException;import java.net.URL;
//strReturnAddress
//.append(returnAddress.getAddressLine(i)).append(
//"\n");
//}
```

```
importandroid.os.Bundle;
import android.app.Activity;import android.content.Intent;
import android.view.Menu;
importandroid.widget.TextView;
import android.os.Bundle;
importandroid.os.StrictMode;
importandroid.annotation.SuppressLint;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
importandroid.graphics.Bitmap;
importandroid.graphics.BitmapFactory;
import android.view.LayoutInflater;
importandroid.view.Menu;
import android.view.View;
importandroid.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import android.widget.ListView;
importandroid.widget.TextView;
importandroid.widget.AdapterView.OnItemClickListener;
publicclassViewresultextendsActivity{
TextViewtv1,tv2;
Stringsource,destination;
public String[] Busno= new String[41];publicString[]Source=new
String[41];publicString[]Destination=newString[41];publicString[] Bustime=new
String[41];
//	publicString[]Latitude=newString[41];
//	publicString[]Longitude=newString[41];
```

```
publicListViewlv;
public String login;publicintrowcount=0;
@SuppressLint("NewApi")@Override
protectedvoidonCreate(BundlesavedInstanceState){super.onCreate(savedInstanceState);set
ContentView(R.layout.activity_viewresult);
Intentii=getIntent();
tv1=(TextView)findViewById(R.id.textView1);tv2=(TextView)findViewById(R.id.textVie
w2);source =ii.getStringExtra("source");
tv1.setText(source);
destination = ii.getStringExtra("destination");tv2.setText(destination);
lv=(ListView)findViewById(R.id.listView1);
//
if (android.os.Build.VERSION.SDK_INT>=9){StrictMode.ThreadPolicy     policy =
 new
StrictMode.ThreadPolicy.Builder().permitAll().build();
StrictMode.setThreadPolicy(policy);}
login=WebService.Busroutedetailsview(tv1.getText().toString(),tv2.getText().toString(),"B
usroutedetails");
//Toast.makeText(getApplication(),login,Toast.LENGTH_LONG).show();
String[] listss= login.split("#");int xx=listss.length;
int iRows = listss.length;rowcount=iRows;
Busno = new String[iRows];Source = new
String[iRows];Destination=newString[iRows];Bustime=new String[iRows];
//Latitude=newString[iRows];
//Longitude=newString[iRows];
inti=0;
// looping through all rows and adding to listfor(i=0;i<listss.length;i++){
String[]ListItems=listss[i].toString().split(",");
Busno[i]=ListItems[0].toString();Source[i]=ListItems[1].toString();Destination[i]=ListItem
s[2].toString();Bustime[i]=ListItems[3].toString();
//Latitude[i]=ListItems[4].toString();
//Longitude[i]=ListItems[5].toString();
//          i++;
}
lv.setAdapter(newImageAdapter(getApplicationContext()));
lv.setOnItemClickListener(new OnItemClickListener() {@Override
publicvoidonItemClick(AdapterView<?>parent,Viewview,intposition,longid)
{
TextViewbugid=(TextView)view.findViewById(R.id.p1);
//TextViewbagname=(TextView)view.findViewById(R.id.bugname);
//TextViewprice=(TextView)view.findViewById(R.id.tvStatus);
//Stringurl=WebService.URLimg+imgpath[position];
//Toast.makeText(UserHome.this,tv.getText(),Toast.LENGTH_LONG).show();
//Intentij=newIntent(Viewcomplaint.this,Updatestatus.class);
//ij.putExtra("bugid",bugid.getText());
//ij.putExtra("userid",tv.getText().toString());
//ij.putExtra("bagname",bagname.getText());
//ij.putExtra("price",price.getText());
//ij.putExtra("url",url);
//startActivityForResult(ij,500);
}
});
//
}
privateclassImageAdapterextendsBaseAdapter{
private Context ctx;privateTextViewtv1;
private TextView tv2;private TextView tv3;private TextView tv4;private TextView
tv5;private TextView tv6;privateImageViewimage;private TextView
tv7;privateTextViewtv8;
privateLayoutInflaterlayoutInflater;
```

```
publicImageAdapter(Contextcontext)
{
this.ctx=context;
layoutInflater=(LayoutInflater)ctx.getSystemService(LAYOUT_INFLATER_SERVICE);}
@Override
publicintgetCount(){
// TODO Auto-generated method stubreturnrowcount;}
@Override
publicObjectgetItem(intposition){
// TODO Auto-generated method stubreturnposition;}
@Override
publiclonggetItemId(intposition){
// TODO Auto-generated method stubreturnposition;}
@Override
publicViewgetView(intposition,ViewconvertView,ViewGroupparent){
ViewGroup onerow=(ViewGroup)
layoutInflater.inflate(R.layout.customcomplaint,null);
tv1=(TextView)onerow.findViewById(R.id.p1);tv2=(TextView)onerow.findViewById(R.i
d.p2);tv3=(TextView)onerow.findViewById(R.id.p3);tv4=(TextView)onerow.findViewByI
d(R.id.p4);
//tv5=(TextView)onerow.findViewById(R.id.p5);
//tv6=(TextView)onerow.findViewById(R.id.p6);
//image=(ImageView)onerow.findViewById(R.id.ivIcon);
URLurl;
//try {
//url=newURL(WebService.URLimg+imgpath[position]);
//Bitmap bmp=BitmapFactory.decodeStream(url.openConnection().getInputStream());
//image.setImageBitmap(bmp);
//}catch(MalformedURLExceptione) {
//TODOAuto-generatedcatchblock
//e.printStackTrace();
//}catch(IOExceptione){
//TODOAuto-generatedcatchblock
//e.printStackTrace();}
tv1.setText(Busno[position]);tv2.setText(Source[position]);tv3.setText(Destination[positio
n]);tv4.setText(Bustime[position]);
//tv5.setText(Latitude[position]);
//tv6.setText(Longitude[position]);
//position++;returnonerow;}}
@Override
publicbooleanonCreateOptionsMenu(Menumenu){
// Inflate the menu; this adds items to the action bar if it is
present.getMenuInflater().inflate(R.menu.viewresult,menu);
return true;
}
}
```

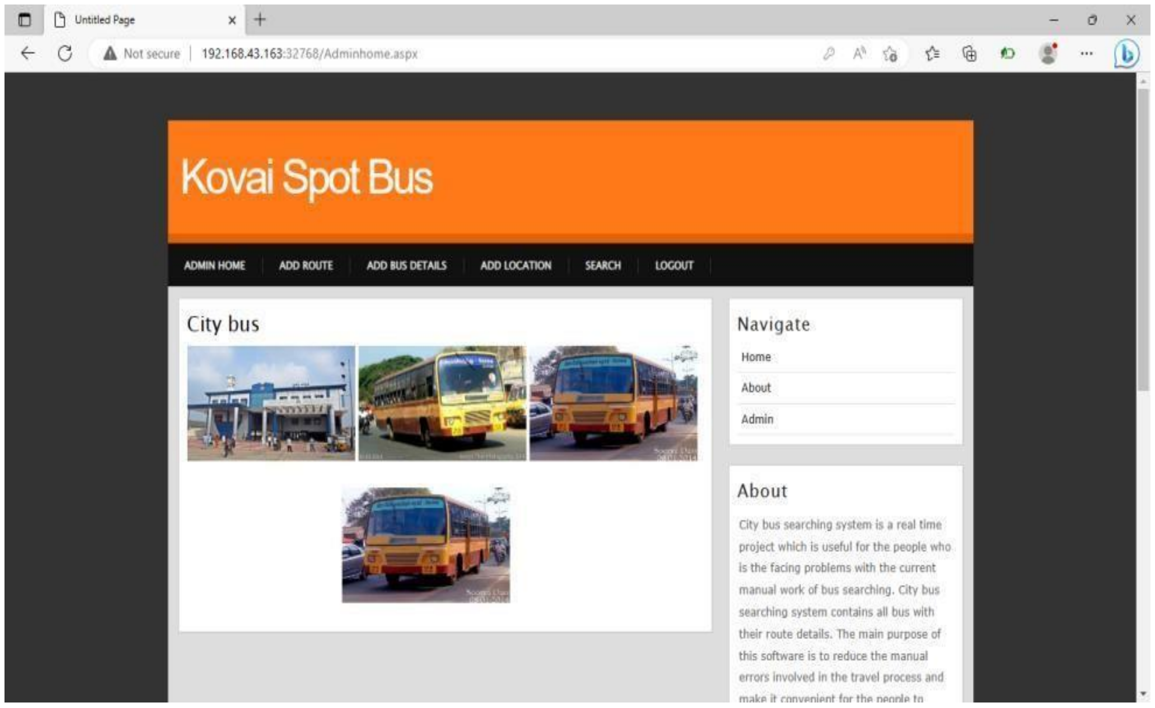
10.2 SCREENSHOTS



City bus searching system is a real time project which is useful for the people who is the facing problems with the current manual work of bus searching. City bus searching system contains all bus with their route details. The main purpose of this software is to reduce the manual errors involved in the travel process and make it convenient for the people to manage their bus details. This project has been made simpler and interactive. This is an android application used to find out the bus number from one place to another place. User need to give the details of source and destination. Accordingly it will display the details of the bus number which is going in that route. It is a time saving application to user. User can easily get the information of the bus number of a particular route. It will also be very helpful for those people who are new to the city.

City bus searching system is a real time project which is useful for the people who is the facing problems with the current manual work of bus searching. City bus searching system contains all bus with their route details.

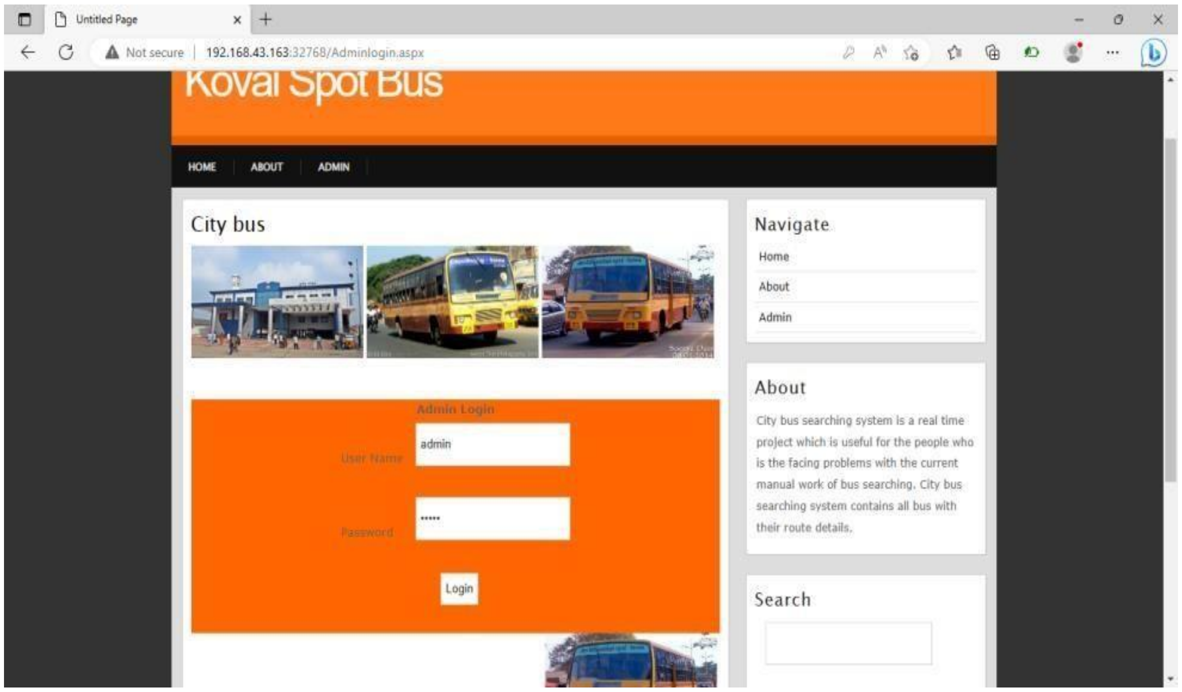
Fig 10.2.1 Home Screen



City bus searching system is a real time project which is useful for the people who is the facing problems with the current manual work of bus searching. City bus searching system contains all bus with their route details. The main purpose of this software is to reduce the manual errors involved in the travel process and make it convenient for the people to

Fig 10.2.2 Admin Login

45



Koval Spot Bus

City bus

Admin Login

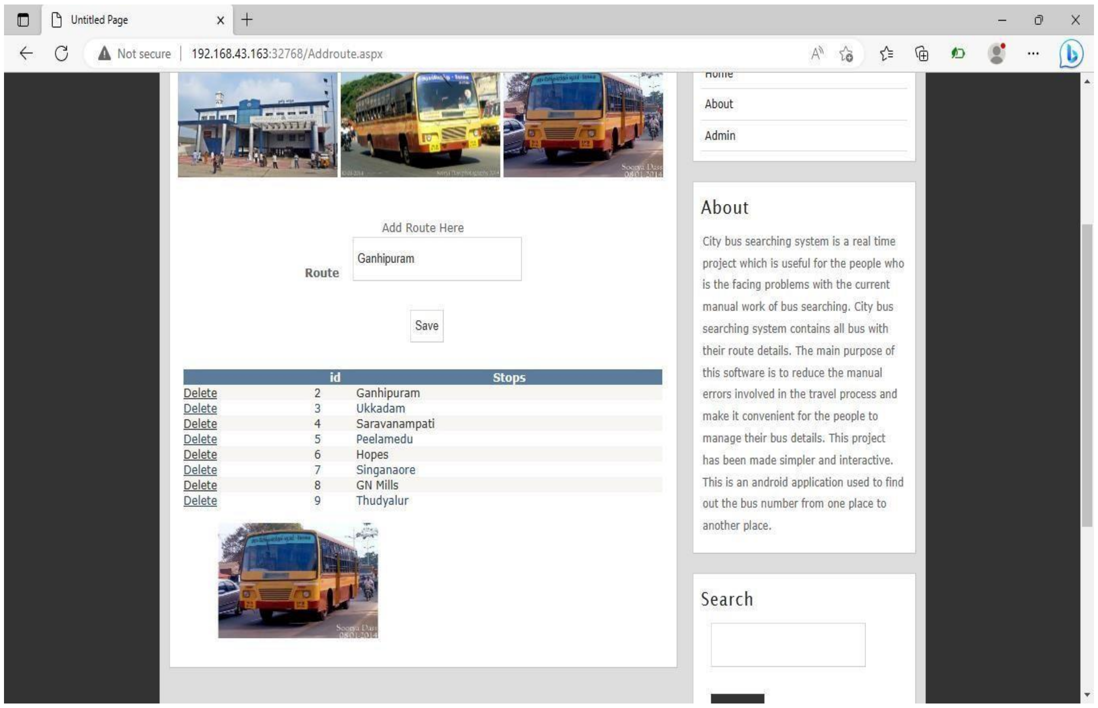
User Name: admin

Password: •••••

About

City bus searching system is a real time project which is useful for the people who is the facing problems with the current manual work of bus searching. City bus searching system contains all bus with their route details.

\



Add Route Here

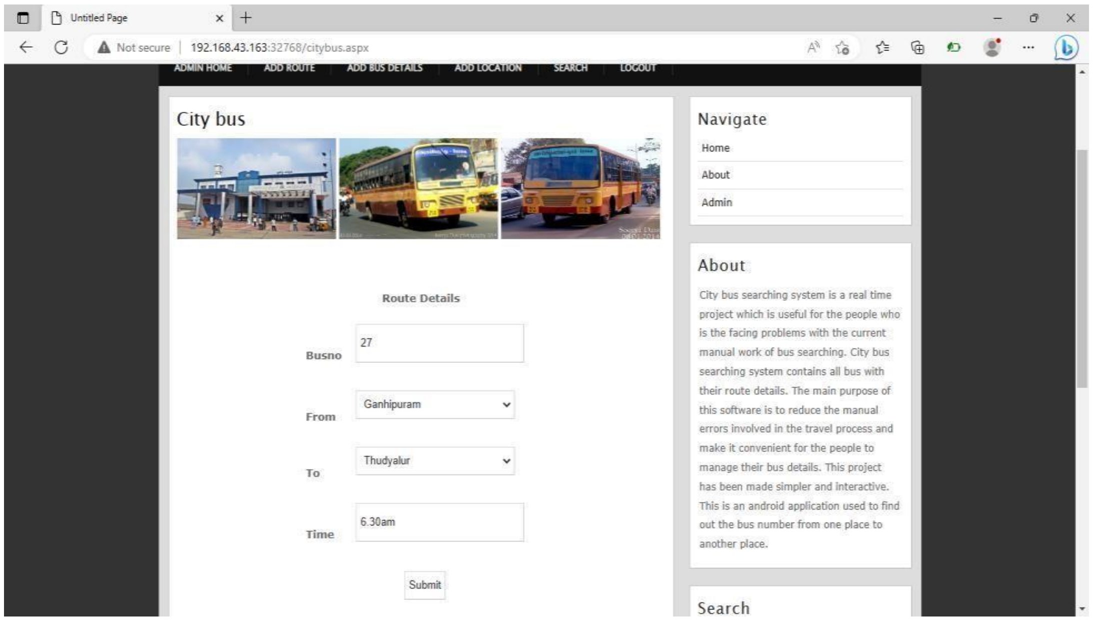
Route: Ganhipuram

	id	Stops
Delete	2	Ganhipuram
Delete	3	Ukkadam
Delete	4	Saravanampati
Delete	5	Peelamedu
Delete	6	Hopes
Delete	7	Singanaore
Delete	8	GN Mills
Delete	9	Thudyalur

About

City bus searching system is a real time project which is useful for the people who is the facing problems with the current manual work of bus searching. City bus searching system contains all bus with their route details. The main purpose of this software is to reduce the manual errors involved in the travel process and make it convenient for the people to manage their bus details. This project has been made simpler and interactive. This is an android application used to find out the bus number from one place to another place.

Fig 10.2.3 Add Bus Route



City bus

Route Details

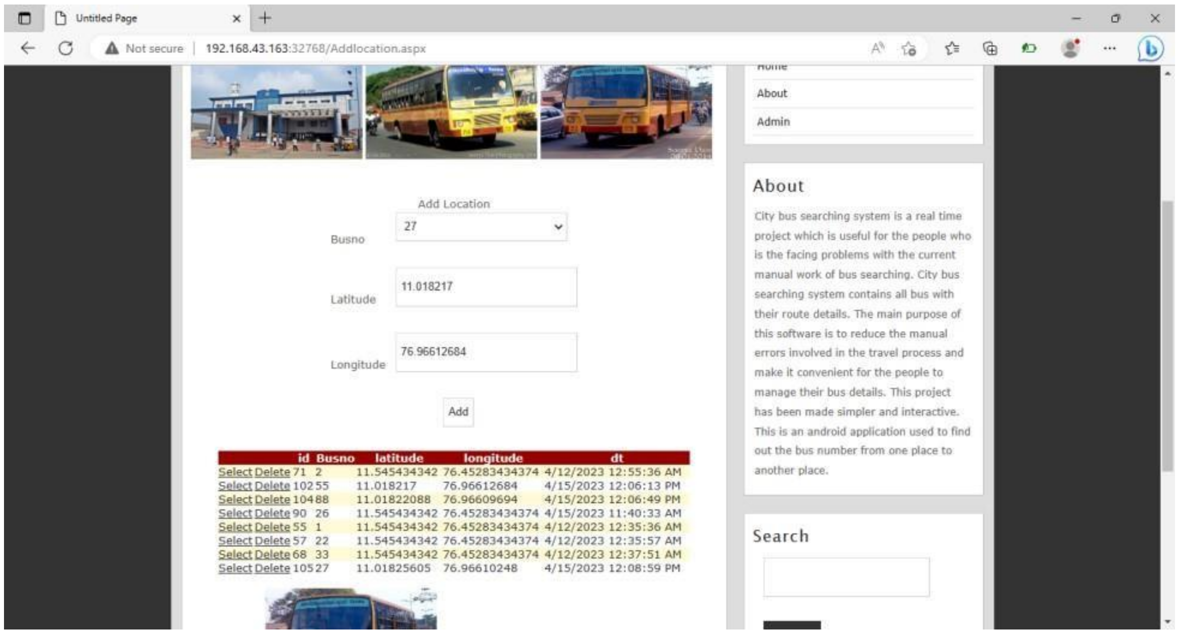
Busno	27
From	Ganhipuram
To	Thudyalur
Time	6.30am

Submit

About

City bus searching system is a real time project which is useful for the people who is the facing problems with the current manual work of bus searching. City bus searching system contains all bus with their route details. The main purpose of this software is to reduce the manual errors involved in the travel process and make it convenient for the people to manage their bus details. This project has been made simpler and interactive. This is an android application used to find out the bus number from one place to another place.

Fig 10.2.4 Add Bus Information



Add Location

Busno	27
Latitude	11.018217
Longitude	76.96612684

Add

	id	Busno	latitude	longitude	dt
Select Delete	71	2	11.545434342	76.45283434374	4/12/2023 12:55:36 AM
Select Delete	102	55	11.018217	76.96612684	4/15/2023 12:06:13 PM
Select Delete	104	88	11.01822088	76.96609694	4/15/2023 12:06:49 PM
Select Delete	90	26	11.545434342	76.45283434374	4/15/2023 11:40:33 AM
Select Delete	55	1	11.545434342	76.45283434374	4/12/2023 12:35:36 AM
Select Delete	57	22	11.545434342	76.45283434374	4/12/2023 12:35:57 AM
Select Delete	68	33	11.545434342	76.45283434374	4/12/2023 12:37:51 AM
Select Delete	105	27	11.01825605	76.96610248	4/15/2023 12:08:59 PM

About

City bus searching system is a real time project which is useful for the people who is the facing problems with the current manual work of bus searching. City bus searching system contains all bus with their route details. The main purpose of this software is to reduce the manual errors involved in the travel process and make it convenient for the people to manage their bus details. This project has been made simpler and interactive. This is an android application used to find out the bus number from one place to another place.

Fig 10.2.5 Add and Update Location

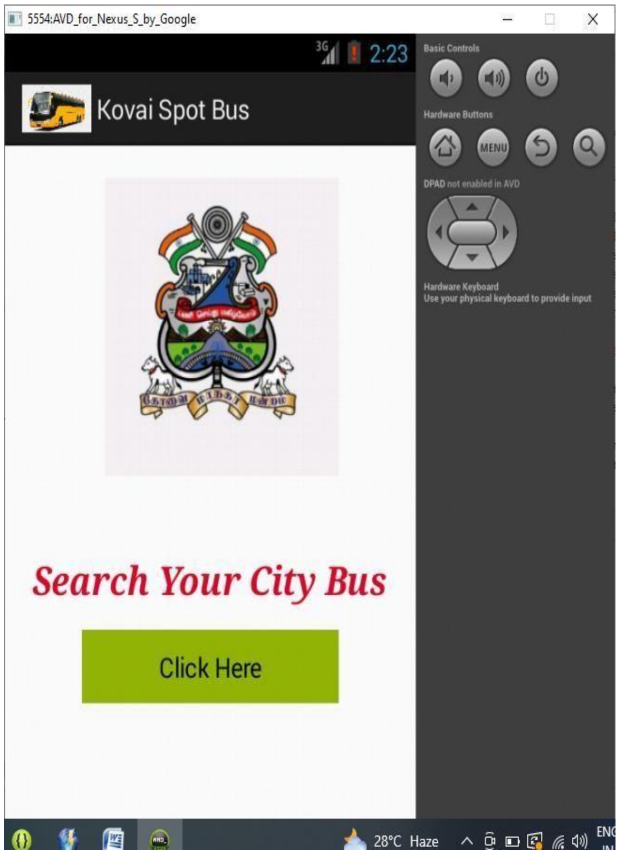


Fig 10.2.6 Home Page – Android Application

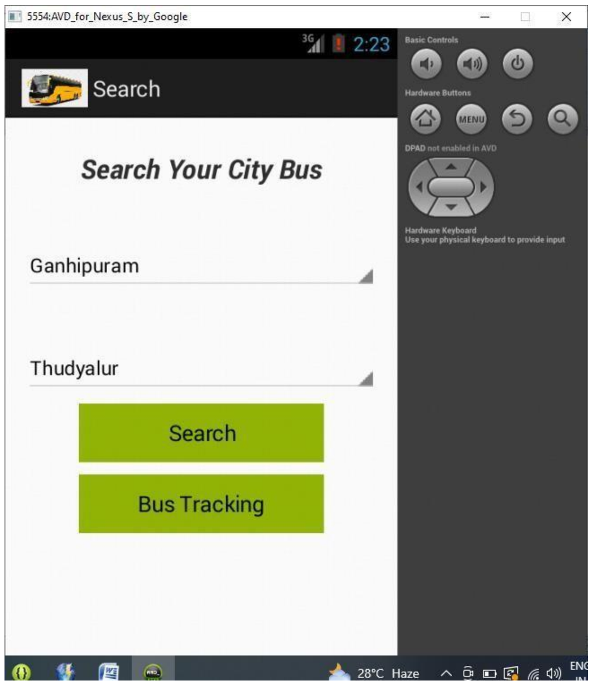


Fig 10.2.7 Search Bus Route

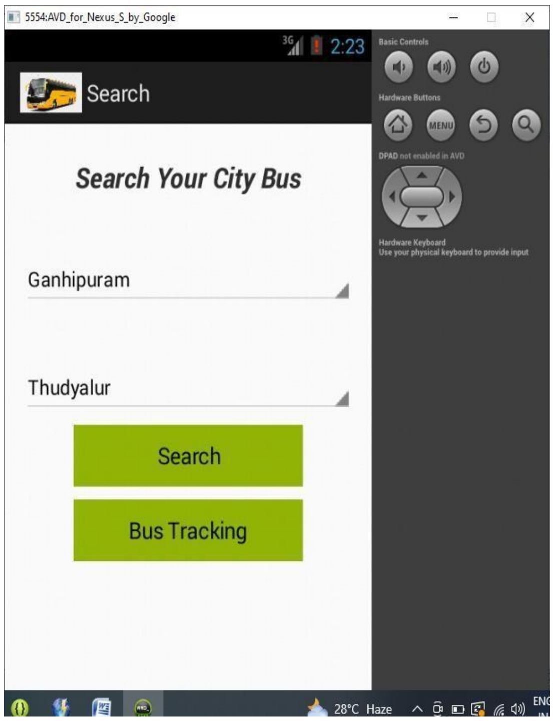


Fig 10.2.7 Show Bus Details

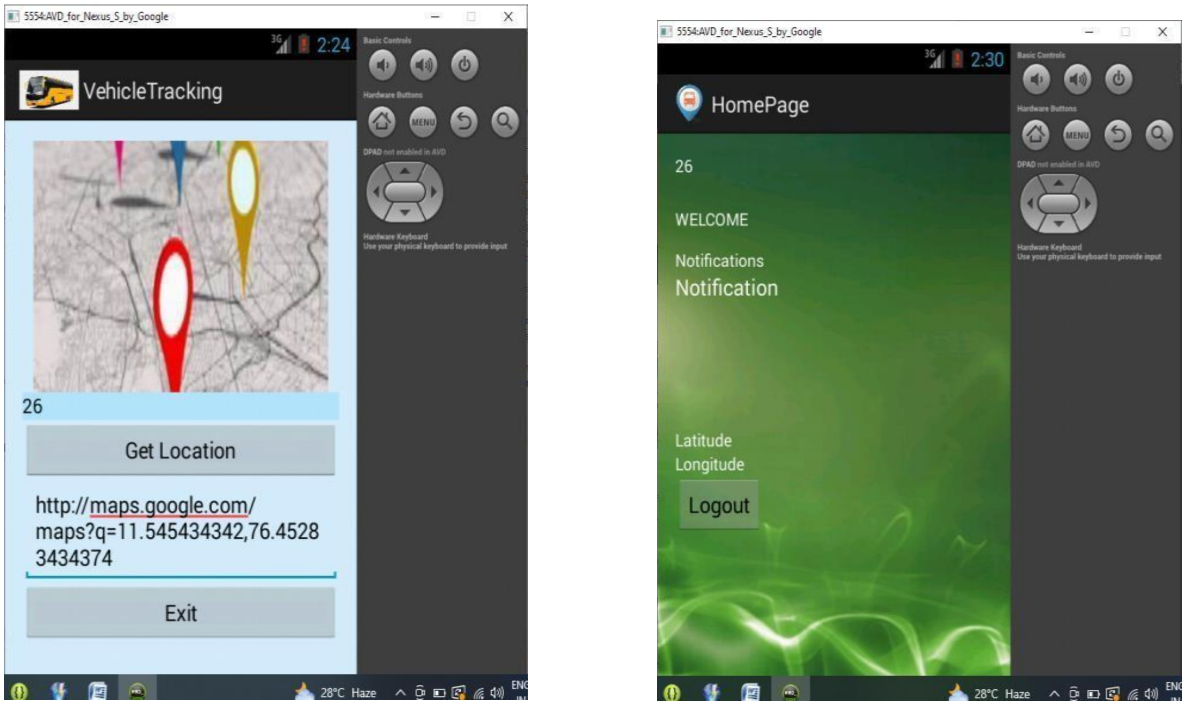

5554:AVD_for_Nexus_S_by_Google
2:24
VehicleTracking
26
Get Location
http://maps.google.com/maps?q=11.545434342,76.45283434374
Exit
5554:AVD_for_Nexus_S_by_Google
2:30
HomePage
26
WELCOME
Notifications
Notification
Latitude
Longitude
Logout

Fig 10.2.8 Location Update

CHAPTER 11

REFERENCES

[1] Ramesh Chandra Gadri "Land Vehicle Tracking Application on Android Platform" – (2019).

[2] Swati Chandurkar, Sneha Mugade - Implementation of Real Time Bus Monitoring and Passenger Information System-(2019).

[3] Sun C W. Zhou and Y. Wang, Scheduling Combination and Headway Optimization of Bus Rapid Transit. Journal of Transportation Systems Engineering and Information Technology-(2019).

[4] Grava S, Urban Transportation System McGraw-Hill Professional-(2020)

[5] Van Oudheusden D.L and Zhu-Trip frequency scheduling for bus route management in Bangkok. European Journal of Operational Research-(2020).

[6] Yan S and H.L Chen - A scheduling model and a solution algorithm for inter-city bus carriers. Transportation Research Part A: Policy and Practice- (2020).

[7] R.K Cheung and Y. Wan - Merging bus routes in Hong Kong's central business Analysis and models transportation Research-(2020).

[8] Sarah Aimi Saad, Aisha Badrul Hisham, Mohamad Hafis - Real-time on-Campus Public Transportation Monitoring System-(2020).

[9] Jerrin George James and Sreekumar Nair - Efficient Real time Tracking of Public Transport Using LoRaWAN and RF Transceivers-(2021).

[10] Darshan Ingle and A. B. Bagwan – Real Time Analysis and Simulation of Efficient Bus Monitoring System-(2022)