

**Certificate of Merit**

This is to certify that the following team of students of B.Tech (CSE) studying in the School of Computer Science and Engineering, VIT, Chennai Campus have successfully completed the project work entitled

**……………………………………………………………..**

between 25th April 2023 and 5th July 2023.

The work was found commendable and good.

Team Members:

1. 21BCE1652 SINDHUJHA.S
2. 21BCE1504 SANJANA.E

Duly Examined and Certified by

**COLLEGE TRANSPORT SYSTEM WITH CHATBOT**

**1.INTRODUCTION:**

The introduction of the software requirement specification (SRS) provides an overview of the entire SRS which follows.

The aim of this document is to gather, analyze and give in-depth insight of transport system management software by defining the problem statement in detail. The detailed requirements of this software are provided in this document.

This is a web-based application which enables to facilitate student inquiries, provide bus schedules, and assist with route planning for improved transportation efficiency and user experience.

**2.SCOPE OF THE PROJECT:**

The scope of the project is to develop a chatbot for the transport management system. This includes development of a platform that allows students, faculty, and staff to interact with a chatbot to manage college transportation services. The chatbot would be designed to handle common queries related to college transportation, such as bus schedules, routes, and availability.

The primary objective is to streamline the transportation process within the college campus. The system aims to enhance the overall transportation experience by providing accurate and up-to-date information regarding bus routes, schedules, and other relevant details. It aims to improve convenience, punctuality, and reliability for students and staff members relying on the college transport services.

**3.SYSTEM REQUIREMENTS SPECIFICATION:**

The system requirements specification for the transport management system using a chatbot defines the functional and non-functional requirements. It outlines the features and capabilities that the system must have to meet the needs of its users. The following are some of the key requirements that should be included in the specification:

1. **Functional Requirements:**

* The chatbot need to be able to offer up-to-date details on the whereabouts of campus buses
* The chatbot should be able to provide details about bus availability, routes, and schedules
* Users should be able to request transportation details from the chatbot, such as ride schedule or reporting a problem, and the chatbot should be able to alert users to changes in bus schedules or routes

1. **Non-Functional Requirements:**

* The chatbot must to be simple to use and engaging.
* Users should have constant access to the chatbot
* The chatbot should to be able to process a lot of requests without stuttering or crashing
* The chatbot needs to be safe and secure, guarding against illegal access to user data
* The system needs to be scalable to support development and growth in the future

1. **Technical Requirements:**

* A secure and flexible platform, using web-based technologies will be used to build the system
* To increase the chatbot's precision and efficacy, machine learning (ML) and natural language processing (NLP) technologies should be used in its development

1. **Performance Requirements:**

* Requests from users should be answered by the chatbot in a matter of seconds.
* A big number of concurrent users should be supported by the system without it slowing down or crashing.
* The system should have a goal uptime of at least 99% and be highly available.

1. **Usability Requirements:**

* The chatbot should be simple to use and navigate, and responses to user queries should be easy to understand and succinct.
* The system should work with a variety of gadgets

1. **DESIGN SPECIFICATION:**

**MODULES:**

**User Interface(UI) Module:**

This module should define the graphical user interface (GUI) for the chatbot. The UI should be intuitive and user-friendly, allowing users to easily interact with the chatbot.

**Natural Language Processing (NLP) Module:**

This module should define the algorithms and techniques used to process natural language inputs from users. It should include methods for identifying intents, extracting entities, and generating responses.

**Database Module:**

This module should define the database schema and data access methods used to store and retrieve information related to bus schedules, routes, and other transportation-related data.

**Security Module:**

This module should define the security measures used to protect user data and prevent unauthorized access to the system. It should include methods for encrypting data, validating user credentials, and implementing access controls.

**Bus Schedule Module:**

Allow users to inquire about the schedule of buses to specific destinations.

Provide information on departure and arrival times, along with any stops or transfers.

**Testing and Quality Assurance Module:**

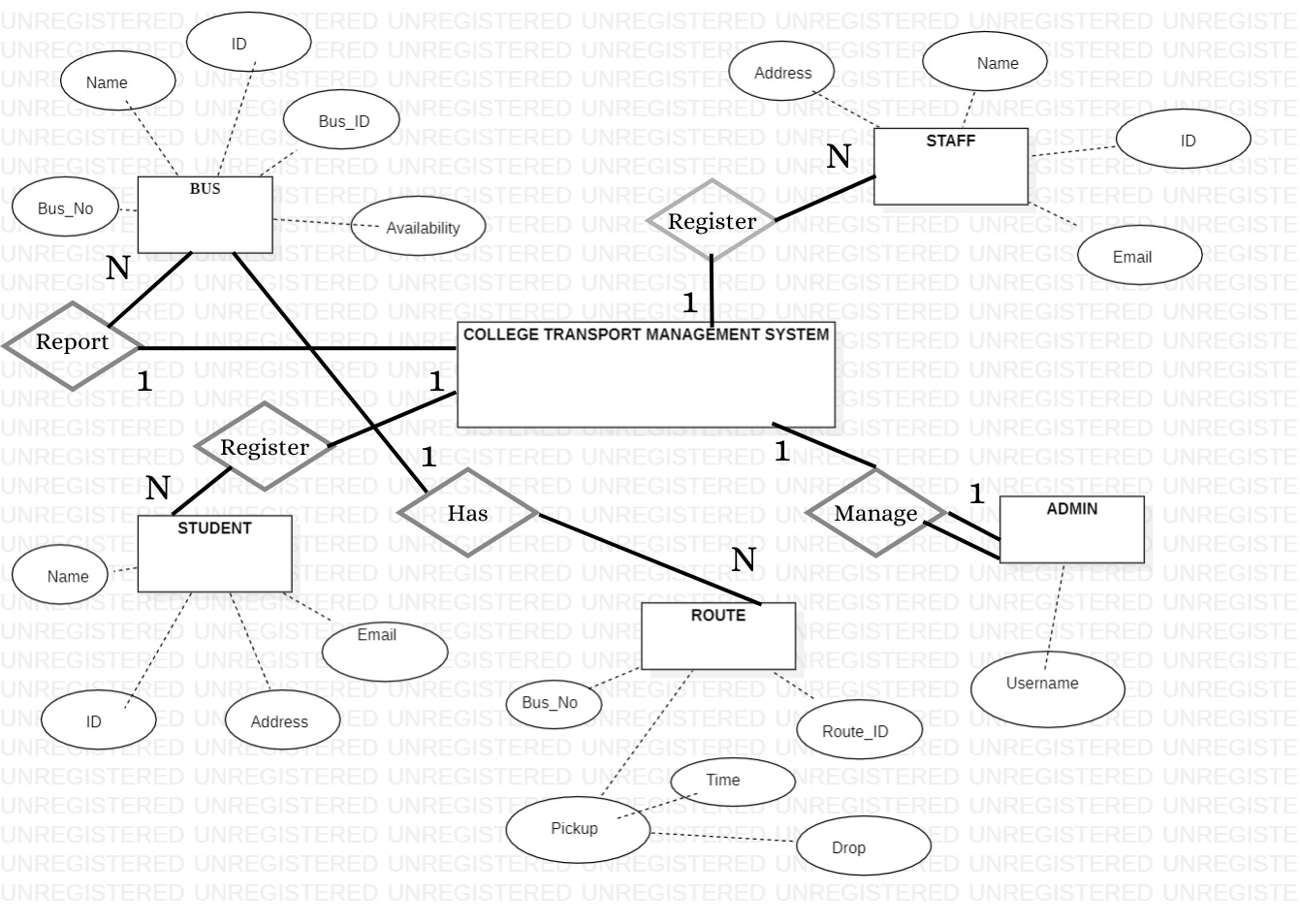
This module should define the methods and procedures used to test and validate the system. It should include methods for identifying and fixing bugs, as well as ensuring that the system meets performance, usability, and other quality standards.

**Bus Route Module:**

Enable users to find the routes of different bus lines within the college campus. Provide detailed descriptions of each route, including major stops and landmarks. Allow users to search for the nearest bus stop based on their current location.

**DATA DESIGN :**

**ER DIAGRAM:**

****

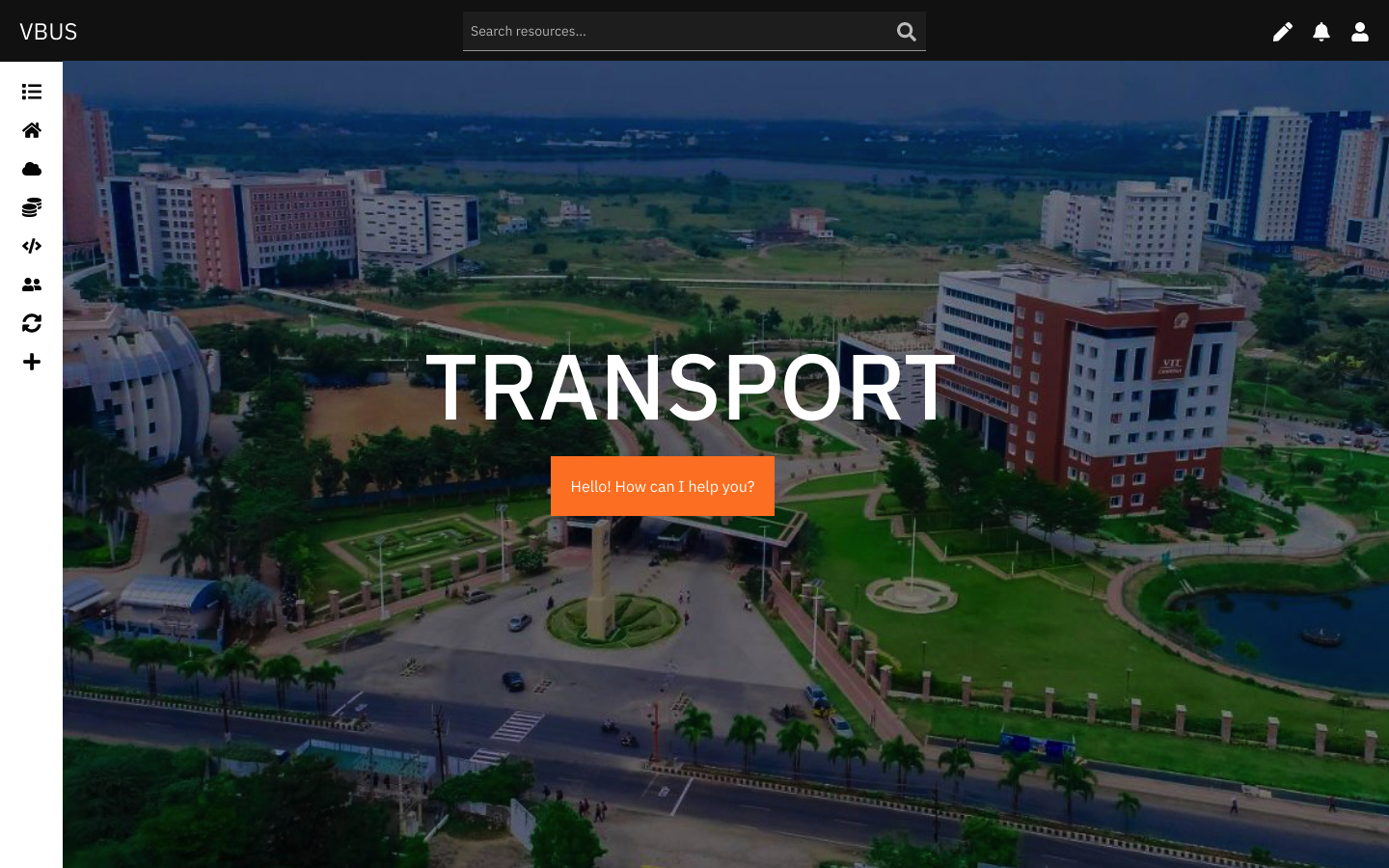
**DATA DICTIONARY:**

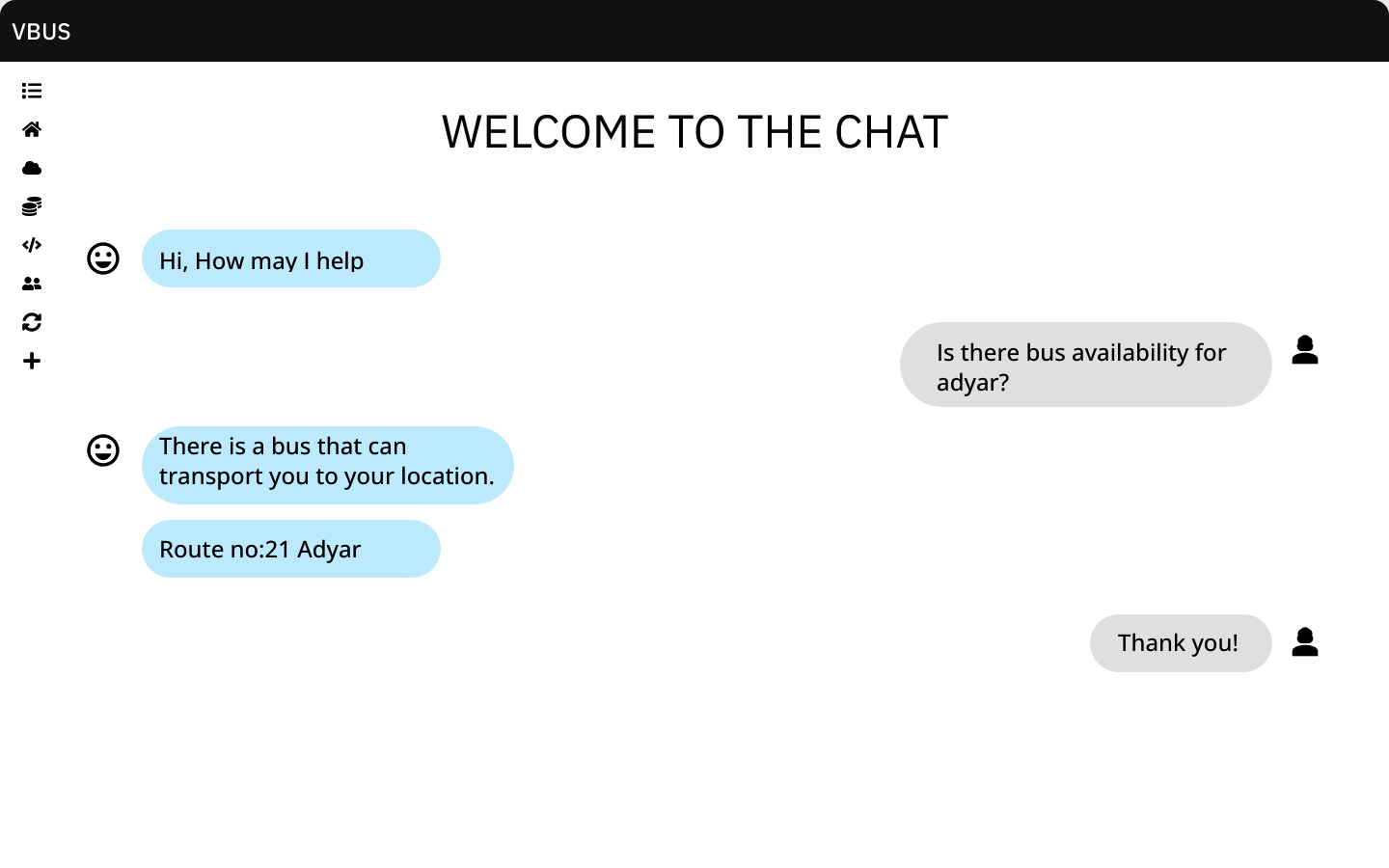
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DATA ITEM** | **DATA TYPE** | **DATA FORMAT** | **DESCRIPTION** | **EXAMPLE** |
| USER\_ID | STRING | XNNNN | Unique identifier for user | M1234 |
| NAME | STRING |  | Name of user | Scott |
| STUDENT\_ID | STRING | XNNNN | Unique identifier for students | M1234 |
| STAFF\_ID | STRING | YYMM | Unique identifier for staffs | SA23 |
| EMAIL | STRING |  | Email of the user | Scott@gmail.com |
| DEPARTMENT | STRING |  | Department to which the user belongs | Computer science |
| PHONE | STRING | NNNNNNNNNN | Unique identifier for user | 1234567890 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DATA ITEM** | **DATA TYPE** | **DATA FORMAT** | **DESCRIPTION** | **EXAMPLE** |
| BUS\_ID | STRING |  | Unique id for all buses | S1234 |
| BUS\_NO | INT |  | Bus number of respective bus | 22 |
| DRIVER\_NAME | STRING |  | Name of the driver | Robert |
| ROUTE\_ID | STRING |  | Unique id for all routes | 600DDE |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **DATA ITEM** | | **DATA TYPE** | | **DATA FORMAT** | | **DESCRIPTION** | | **EXAMPLE** |
| ROUTE\_ID | | STRING | |  | | Unique id for all routes | | 600DDE |
| ROUTE\_NAME | | STRING | |  | | Name of the route | | MCC |
| START\_LOCATION | | STRING | |  | | Start point of the bus | | MCC COLLEGE |
| END\_LOCATION | | STRING | |  | | End point of the bus | | COLLEGE |
| **DATA ITEM** | **DATA TYPE** | | **DATA FORMAT** | | **DESCRIPTION** | | **EXAMPLE** | |
| SCHEDULE\_ID | STRING | |  | | Unique identifier for all schedules | | 12678 | |
| BUS\_ID | STRING | |  | | Unique identifier for all buses | | S1234 | |
| ROUTE\_ID | STRING | |  | | Unique identifier for all routes | | 600DDE | |
| DEPARTURE\_TIME | STRING | |  | |  | | 16:00 PM | |

**UI DESIGN:**

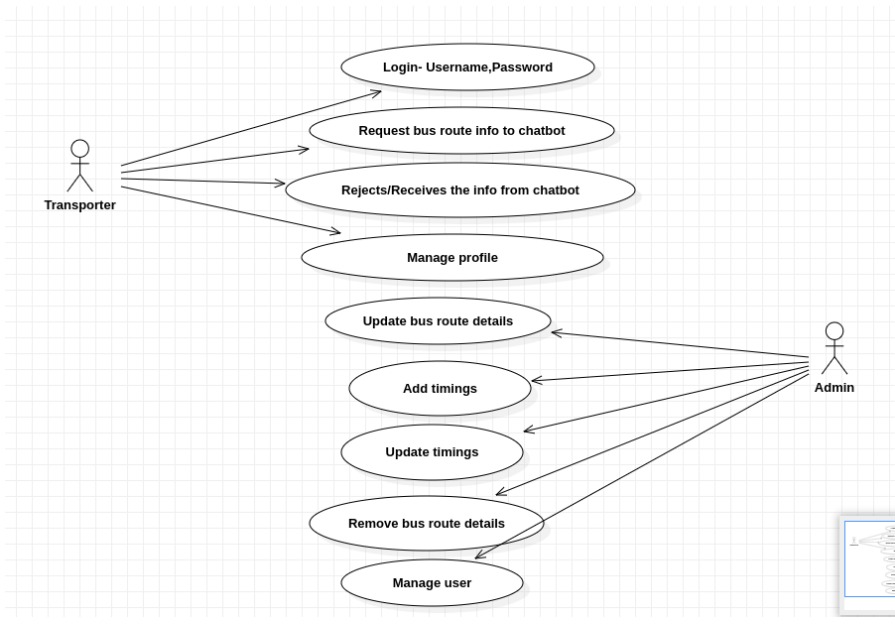




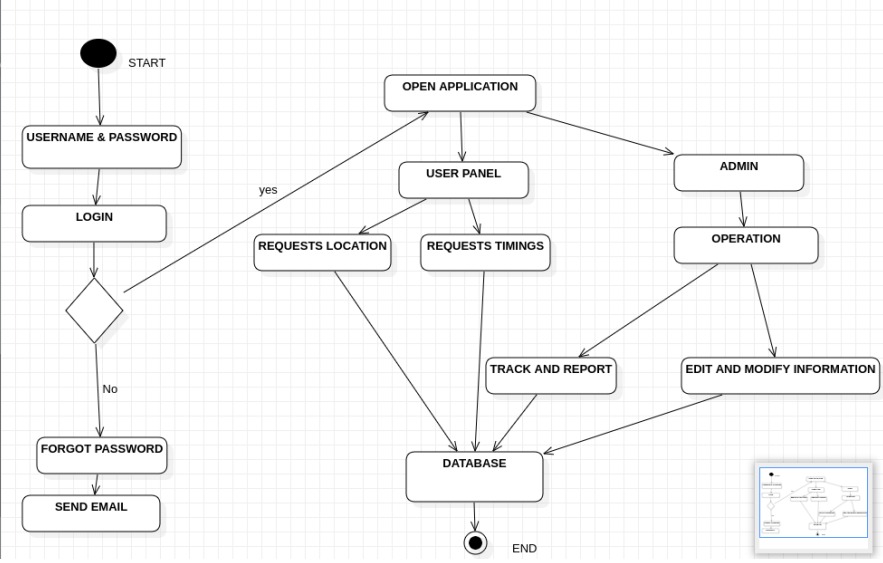
**OVERALL ARCHITECTURE:**

1. **RELEVANT UML DIAGRAMS:**

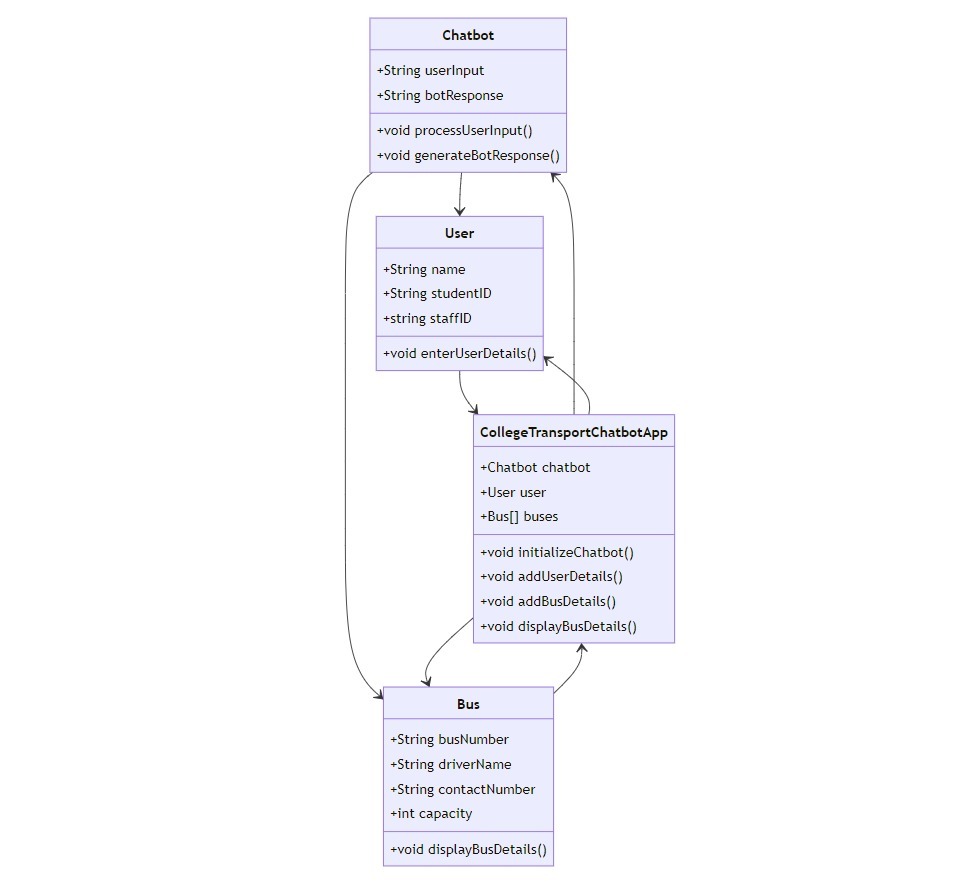
**USE CASE DIAGRAM:**

****

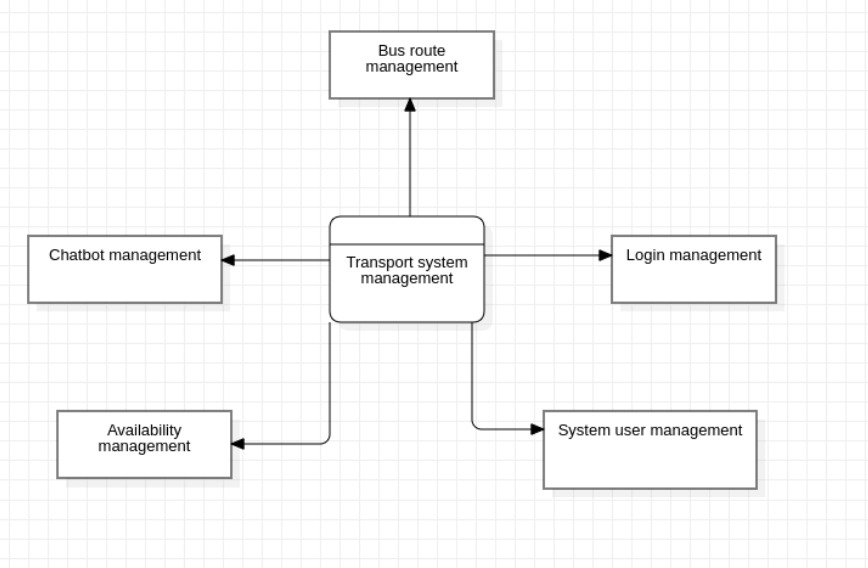
**ACTIVITY DIAGRAM:**

****

**CLASS DIAGRAM:**

****

**DATAFLOW DIAGRAM:**

****

1. **ALGORITHM:**

**Initialize the Chatbot:**

Load necessary data and resources.

Set up the chatbot interface.

Receive User Input:

Listen for user queries or commands through the chat interface.

Process and sanitize the input to remove any irrelevant characters or symbols.

**Determine User Intent:**

Apply natural language processing techniques to understand the user's intent.

Use techniques such as intent recognition, keyword matching, or machine learning models to identify the user's desired action or information.

Route the User Request:

Based on the user intent, direct the request to the appropriate module for further processing.

Match the user intent with predefined patterns or use machine learning models to determine the relevant module.

Process the User Request:

Within the specific module, handle the user's request according to the identified intent.

Access relevant data sources or APIs to retrieve the required information.

**Generate Response:**

Based on the user request and processed information, generate a suitable response.

Format the response in a clear and understandable manner.

Display Response:

Send the generated response back to the chat interface to be displayed to the user.

Update the chat interface with the response text.

**Repeat the Process:**

Continue listening for user input and repeat the steps from 2 to 7 until the user ends the conversation.

1. **TEST PLAN / METHODS:**

**1. Objective:**

The objective of the testing phase is to ensure that the College Transport Management System, integrated with a chatbot on a website, meets the specified requirements and functions as intended. The testing process aims to identify any defects, errors, or areas for improvement.

**2. Test Environment:**

The testing environment should replicate the production environment as closely as possible. It should include:

- Web server and database server

- Latest version of the website with chatbot integration

- Test data (users, buses, routes, schedules etc.)

**3. Test Scenarios:**

Define a set of test scenarios covering various aspects of the College Transport Management System. These scenarios should encompass both positive and negative test cases. Some examples include:

a. User Registration:

- Test user registration functionality and ensure new users can successfully register through the chatbot interface.

b. Bus Schedule:

- Verify that the chatbot displays accurate and up-to-date bus schedules based on the selected date and location.

c. Error Handling:

- Validate that the chatbot provides appropriate error messages and handles invalid user inputs gracefully.

d. Integration Testing:

- Ensure smooth integration between the website, chatbot, and database systems.

**4. Test Data:**

Create a complete set of test data to account for various circumstances. Sample user profiles of students, employees, and administrators may be included.

- Different bus configurations (capacity, route, and driver variations)

- Variable schedules

**5. Test Execution:**

Execute the specified test scenarios using the test data that has been produced. Record the outcomes, noting observable behaviours, system reactions, and any problems or flaws found. As necessary, combine manual testing with automated testing technologies.

**6. Defect Reporting:**

Keep a record of all flaws found, including their severity, how to recreate them, and any pertinent screenshots or logs. Utilizing an established defect tracking system, notify the development team of the issues.

**7. Regression Testing:**

After fixing any reported bugs, run regression testing to make sure no new problems were caused by the fixes. Rerun the previously completed test scenarios and compare the outcomes to what was anticipated.

**8. Performance Testing:**

Examine the chatbot's and the website's performance under various load scenarios. To make that the system can efficiently handle several concurrent user interactions, test response times, system stability, and scalability.

**9. User Acceptance Testing (UAT):**

Involve end users in the User Acceptance Testing (UAT) phase, such as college employees and students. Obtain feedback and confirm that the system satisfies users' needs, is simple to use, and performs as expected.

**10. Documentation:**

Document the test plan, test scenarios, test data, test results, and any additional observations or recommendations. This documentation will serve as a reference for future maintenance, upgrades, or enhancements.

1. **SCREEN SHOTS:**
2. **CONCLUSION AND FUTURE WORK:**

The College Transport System using a chatbot has been designed and implemented successfully, as outlined in the Software Requirements Specification (SRS) report. The integration of a chatbot on the website has provided an efficient and user-friendly interface for managing the college transport system. The SRS report identified the key requirements and functionalities of the system, and the development team has worked diligently to meet those requirements.

In conclusion, the campus Transport System, which manages transportation within the campus by using a chatbot on a website, has been a great success. It has made communication easier to understand and given an intuitive user interface. The system has huge potential for improvement, making it a crucial tool for effective college transportation administration.