Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 11, Issue 3, pp: (1-28), Month: September - December 2024, Available at: www.noveltyjournals.com

Automating Transport Inquiries: A Voice-Activated Solution for Modern Travel Needs

Biswas Sougato[1], Hasan Md Rakibul[2], Musa Sheikh MD Abu Sk[3], Md Jobayer[4]

[1]College of Information Engineering and Artificial Intelligence, Yangzhou University, China

[2]College of Information Engineering and Artificial Intelligence, Yangzhou University, China

[3]College of mechanical engineering, Yangzhou University, China

[4]Department of Computer Science and Engineering, American International University Bangladesh, Bangladesh

DOI: https://doi.org/10.5281/zenodo.13882208

Published Date: 02-October-2024

Abstract: **The Automating Transport Inquiries aims to revolutionize how travelers access transport information at terminals like airports, train stations, and bus stops. This innovative system addresses significant challenges faced by users, such as long wait times for assistance and the unavailability of information services outside regular hours, particularly benefiting those with visual impairments or difficulties using text-based interfaces. By utilizing advanced technologies such as Python.NET and Microsoft Speech SDK, the system enables users to interact through voice commands, providing real-time transport details efficiently and accurately. The project not only emphasizes technological feasibility but also highlights economic benefits, including reduced operational costs and enhanced customer satisfaction through improved service accessibility. Furthermore, the report delves into the social implications of this technology, addressing privacy concerns and its potential impact on employment within the transport sector. The system is designed to be scalable, capable of handling multiple queries simultaneously while ensuring that users receive accurate information at all times. By automating transport inquiries, this initiative aligns with contemporary demands for efficient public service solutions, ultimately enhancing user experience and promoting inclusivity in accessing vital transport information. This comprehensive overview underscores a transformative approach to disseminating transport information through voice technology, setting a new standard for user-friendly interfaces in public transportation systems.**

Keywords: **Voice Recognition, Transport Enquiry, Automation, Accessibility, Database Integration, Real-time Information, Speech Technology.**

1. INTRODUCTION

The Automating Transport Inquiries System is feasible from technological, economic, operational, timetable, and legal standpoints. The advantages of increased efficiency, decreased expenses, and improved user experience beyond the necessary expenditure. Effective strategizing and meticulous implementation can result in the effective integration of this system.

Technical Feasibility:

- The system can be developed using existing technologies like Python.Net, SQL Server, and Microsoft Speech SDK.
- Hardware requirements are minimal - a standard PC with microphone and speakers is sufficient.
- Speech recognition and text-to-speech technologies are mature enough to support this application.
- Integration with existing transport databases is feasible.

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 11, Issue 3, pp: (1-28), Month: September - December 2024, Available at: www.noveltyjournals.com

Economic Feasibility:

- Development costs are reasonable as it uses standard technologies and hardware.
- Can reduce operational costs by automating enquiry services and reducing staff requirements.
- Potential for increased customer satisfaction and efficiency can provide good return on investment.
- Maintenance costs are low as it requires minimal human intervention once deployed.

Operational Feasibility:

- Addresses a real need for faster access to transport information at terminals.
- Voice-based interface makes it easy to use for a wide range of users.
- Can operate 24/7 without human staffing.
- Easily scalable to handle multiple simultaneous queries.
- Administrators can easily update information through the system.

Schedule Feasibility:

- Can be developed in a reasonable timeframe using standard software development methodologies.
- Phased implementation possible - core functionality can be deployed first and additional features added later.

Legal Feasibility:

- No major legal hurdles as it automates an existing service.
- Privacy and data protection laws need to be considered for handling user data.

1.1 Problem definition

The Automating Transport Inquiries System is designed to meet the demands of travelers seeking quick and easy access to transportation-related information at locations such as airports, train stations, and bus stops. It aims to address a number of important issues, including lengthy wait times for passengers to speak with human staff members, a lack of 24/7 availability for information services, inefficiencies and potential errors in manual information dissemination, challenges faced by those with vision impairments or those who have trouble interacting with text-based interfaces when accessing transportation information, the need for a more interactive and user-friendly method of retrieving transportation details in comparison to traditional methods, the need for an automated system that can handle multiple simultaneous queries without the need for human intervention, the need for a solution that is easily updated with new routes, timings, and transportation information, and the requirement for a system that can operate in multiple languages to accommodate a variety of user groups. By using voice commands for input, making it accessible to a wide range of users, offering voice output for information, improving usability, automating the information retrieval process to cut down on wait times and human error, providing 24/7 availability of transport information, facilitating simple updates to transport data through an administration interface, and possibly supporting multiple languages for increased accessibility, the Voice-Based Automatic Transport Enquiry System aims to address these issues. By addressing these issues, the system aims to significantly improve the efficiency and user experience of obtaining transport information at terminals.

1.2 Purpose of writing

Now this is the age of speed. Everything happens in the speed of supersonic. The data can be transferred at the speed of light in the digital medium, can travel in the supersonic speed, hence three is a need of information inflow in the same speed. Here is one such need of information fast enough. We have experienced in waiting to a transport terminal for transport controllers to get the information about the transport facility. We encounter so many times there will be no person for providing this information which significantly wastes the time just to know whether there is any facility or not. Here is one solution for such a problem which lessens the human intervention in providing such information in the transport terminals. Voice Based Automated Transport Enquiry System is the enquiry system which operates based on the voice input given by the user. There is no communication which is understood more appropriately than voice. This system too uses the voice commands and gives the required information in the form of voice. This system is can be installed in any transport terminal like Bus stands, Railway terminals or airports.

1.3 Requirements and Objectives

User Requirements

1. Voice command recognition and processing: The system accept, recognize, and process voice commands from users. Commands are given in the form of voice and processed using voice processing and speech recognition.

2. Database integration and information retrieval: The system search and retrieve appropriate information from the database based on user queries. It is able to provide details such as bus timings, available seats, and platform information.

3. Multi-modal output: The system provides retrieved information in the form of voice output using speech control, and also display relevant information on a screen for verification.

System Requirements

A set of system services and constraints in detail, The System requirements are the more detailed specification of the User Requirements it sometimes serves as a contract between the user and the developer.

Software requirements

1. PyCharme
2. Python.NET
3. MS Speech SDK
4. MS SQL

Hardware requirements (minimum)

1. Processor : Pentium IV
2. Monitor : SVGA
3. RAM : 128MB
4. Speed : 1.5GHz
5. Secondary Device : 20GB
6. Speaker
7. Microphone

Objectives

- **Improve Efficiency:**
- Reduce wait times for passengers by providing quick access to transport information.
- Reduce the amount of human mistake and intervention by automating the information retrieval process.
- **Enhance User Experience:**
- Provide an interactive and user-friendly interface through voice commands.
- Makes sure that the system can be used by those who have trouble interacting with text-based interfaces or who are visually impaired.
- **24/7 Availability:**
- Ensure that the system is always open and that users may always access transportation-related data.

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 11, Issue 3, pp: (1-28), Month: September - December 2024, Available at: www.noveltyjournals.com

- **Ease of Maintenance:**
- Provide an administrative interface for simple updates to transport data, routes, and timings.
- **Support Multiple Languages:**
- To accommodate a range of user demographics, it may be possible to support different languages
- **Scalability and Cost Efficiency:**
- By automating enquiry services, create a system that can scale effectively and save operating expenses.

1.4 Feasibility of technical conditions

Technical Feasibility of the Voice-Based Automatic Transport Enquiry System in Python .NET

Software Requirements

1. **Python .NET Integration**:

- Python .NET (Python for .NET) allows Python code to interact with .NET Common Language Runtime (CLR) libraries. This integration is crucial for leveraging .NET libraries while writing the main application logic in Python.
- pythonnet facilitate this integration, enabling the use of .NET functionalities within Python scripts.

2. **Speech Recognition and Text-to-Speech**:

- **Speech Recognition**: We have used Microsoft Speech SDK for capturing and processing voice commands. This supports various speech recognition engines and APIs, including Google Web Speech API.
- **Text-to-Speech**: Used gTTS (Google Text-to-Speech) to convert text responses into speech output.

3. **Database Management**:

- **SQL Server**: Using MySQL for SQL Server databases. These librarie provide robust support for executing SQL queries and managing database connections.
- **SQLite**: For simpler implementations or prototypes, SQLite can be used as a lightweight database solution. The sqlite3 module in Python provides a straightforward interface for SQLite databases.

4. **Development Environment**:

- **IDE**: PyCharm is used for developing the application, offering extensive support for Python development and debugging.
- **Frameworks**: Django is used for developing the web interface. This framework provides robust support for building web applications and APIs.

System Components

1. **Speech Recognition Component**:

- Captures and processes voice commands using the SpeechRecognition library.
- Converts audio input into text that the system can understand and act upon.

2. **Database Interaction Component**:

- Manages connections to the SQL Server.
- Executes queries to retrieve transport information based on user commands.

3. **Text-to-Speech Component**:

- Converts text responses into speech using gTTS.

- Ensures that the user receives information audibly.

4. **User Interface Component**:

- Provides a visual interface for displaying transport information.
- Developed using Django for the interfaces.

5. **Administration Component**:

- Allows maintenance personnel to update transport information and manage user commands.
- Provides an interface for adding new routes, updating timings, and modifying existing data.

1.5 Feasibility of soclal factors

The Automating Transport Inquiries system appears socially feasible, offering substantial advantages in terms of accessibility and user satisfaction. Nevertheless, it is important to thoroughly contemplate privacy issues, cultural sensitivity, and potential repercussions on work. To ensure the system's success and acceptance in society, it is essential to address these social concerns through careful design and execution.

- **Accessibility and Inclusivity:** The approach possesses significant promise for enhancing accessibility, particularly for individuals with visual impairments or those who have difficulties with text-based interfaces. Utilizing voice commands expands the availability of transportation information to a broader spectrum of individuals, including those with little reading or technological proficiency. This is in line with society's objectives of establishing more comprehensive public services.

- **User Experience and Adoption:** Popular voice assistants such as Siri and Alexa are making voice-based technologies more and more common among individuals. The increasing familiarity with speech technologies may result in a faster implementation and acceptance of the transit enquiry system. The natural language interface is characterized by its intuitive and user-friendly nature, distinguishing it from traditional text-based systems. This quality has the potential to attract a wide spectrum of users.

- **Privacy and Trust:** Like any voice-based technologies, there may be societal issues around privacy and the security of data. Ensuring transparent information regarding the handling of user data will be essential for establishing confidence. Privacy issues and social standards may cause certain users to be reluctant to utilize voice commands in public areas.

- **Impact on Employment:** The automation of transport enquiries could raise concerns about job displacement for human operators. This social impact needs to be carefully considered and addressed, perhaps by retraining staff for other roles or emphasizing how the system can complement rather than replace human customer service.

- **Digital Divide:** Although the method enhances accessibility for a wide range of individuals, it may also create a disparity for people who are not at ease with or unable of utilizing speech technology. It is crucial to have alternate access ways to prevent the exclusion of any user groups.

- **Public Benefit:** The potential to reduce wait times, improve efficiency, and provide 24/7 access to transport information aligns well with societal demands for better public services. This could lead to improved overall satisfaction with public transport systems.

2. DESIGN

2. Demand analysis

The demand analysis for the Automating Transport Inquiries System entails comprehending the necessity of implementing such a system, identifying the intended users, and assessing the possible advantages and obstacles. This study facilitates the assessment of the viability and prospective triumph of the system in the market.

Identified Needs and Problems

1. **Rapid Information Access**:

- Users at transport terminals often face long wait times to get information from human staff.
- There is a need for quick and efficient access to transport information, especially during peak hours.

2. **24/7 Availability**:

- Traditional information desks are not available round-the-clock, leading to inconvenience for travelers.
- An automated system can provide information at any time, enhancing user convenience.

3. **Accessibility**:

- Visually impaired individuals and those who struggle with text-based interfaces need an alternative way to access transport information.
- A voice-based system can cater to these users, making transport information more accessible.

4. **Efficiency and Accuracy**:

- Manual information dissemination can be prone to errors and inefficiencies.
- An automated system can reduce human errors and provide accurate, real-time information.

5. **User-Friendly Interaction**:

- Users prefer interactive and intuitive systems.
- Voice commands offer a natural and user-friendly way to interact with the system.

Target Users

1. **General Public**:

- Daily commuters, occasional travelers, and tourists who need quick access to transport information.

2. **Visually Impaired Individuals**:

- Users who rely on auditory information due to visual impairments.

3. **Elderly and Technologically Challenged Users**:

- Individuals who may find it difficult to use text-based or complex digital interfaces.

4. **Transport Authorities**:

- Organizations managing transport terminals looking to improve service efficiency and reduce operational costs.

Potential Benefits

1. **Improved User Experience**:

- Faster access to information and reduced wait times enhance overall user satisfaction.
- Voice interaction makes the system more accessible and user-friendly.

2. **Operational Efficiency**:

- Reduces the need for human staff at information desks, leading to cost savings.
- Automates routine queries, allowing staff to focus on more complex tasks.

3. **Inclusivity**:

- Provides an accessible solution for visually impaired users and those with other disabilities.

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 11, Issue 3, pp: (1-28), Month: September - December 2024, Available at: www.noveltyjournals.com

- Supports multiple languages, catering to a diverse user base.

4. **24/7 Availability**:

- Ensures that users can access transport information at any time, improving convenience.

5. **Scalability**:

- The system can handle multiple simultaneous queries, making it suitable for busy transport terminals.

Challenges

1. **Speech Recognition Accuracy**:

- Ensuring high accuracy in recognizing voice commands, especially in noisy environments, is a significant challenge.
- Advanced noise-cancellation techniques and robust speech recognition algorithms are required.

2. **Integration with Existing Systems**:

- Seamless integration with existing transport management systems and databases is crucial for real-time updates and data exchange.
- Compatibility issues need to be addressed to ensure smooth operation.

3. **User Privacy and Trust**:

- Addressing privacy concerns related to voice data collection and storage is essential to build user trust.
- Clear communication about data handling practices is necessary.

4. **Cultural and Linguistic Diversity**:

- The system must support multiple languages and dialects to cater to a diverse user base.
- Cultural sensitivities need to be considered in the system design.

5. **Maintenance and Updates**:

- Regular updates to transport information and system maintenance are required to ensure accuracy and reliability.
- An easy-to-use administration interface is necessary for updating data.

2.1 Data flow diagram of the target system

A Data Flow Diagram (DFD) is a graphical representation of the flow of data through a system. It shows how data is processed by a system in terms of inputs and outputs. Below is a detailed description of the DFD for the Automating Transport Inquiries System based on the provided document.

Context Level DFD (Level 0)

The context level DFD provides a high-level overview of the system, showing the system as a single process with its interactions with external entities. **Entities:**

1. **User**: Interacts with the system by providing voice commands and receiving information.
2. **Administrator**: Manages the system by updating data and handling user accounts.
3. **Accountant**: Manages financial and operational data related to the transport system.

Process:

- **Voice-Based Automated Transport Enquiry System (VBATES)**: The main system that processes user commands and provides transport information.

Data Stores:

- **Bus Information Database**: Stores details about buses, routes, timings, and availability.
- **User Commands Database**: Stores recognized voice commands and their corresponding actions.

Data Flows:

1. **User to VBATES**: Voice commands for transport enquiries.
2. **VBATES to User**: Transport information (e.g., bus timings, routes).
3. **Administrator to VBATES**: Updates to bus information and user accounts.
4. **VBATES to Administrator**: Confirmation of updates.
5. **Accountant to VBATES**: Financial and operational data.
6. **VBATES to Accountant**: Reports and data summaries.

Detailed DFD (Level 1)

The detailed DFD breaks down the main process into sub-processes, showing more specific data flows and interactions.**Processes:**

1. **Capture Voice Command**: Captures and processes the user's voice input.
2. **Recognize Command**: Uses speech recognition to convert voice input into text.
3. **Query Database**: Searches the database for relevant transport information based on the recognized command.
4. **Provide Information**: Converts the retrieved data into voice output and displays it on the screen.
5. **Update Database**: Allows the administrator to update bus information and user accounts.
6. **Generate Reports**: Provides financial and operational reports to the accountant.

Data Stores:

- **Bus Information Database**: Stores bus details, routes, timings, and availability.
- **User Commands Database**: Stores recognized commands and their corresponding actions.

Data Flows:

1. **User to Capture Voice Command**: Voice input from the user.
2. **Capture Voice Command to Recognize Command**: Audio data for speech recognition.
3. **Recognize Command to Query Database**: Recognized text command.
4. **Query Database to Bus Information Database**: Query for transport information.
5. **Bus Information Database to Query Database**: Retrieved transport information.
6. **Query Database to Provide Information**: Transport information to be converted to voice and displayed.
7. **Provide Information to User**: Voice output and visual display of transport information.
8. **Administrator to Update Database**: Updates to bus information and user accounts.
9. **Update Database to Bus Information Database**: Updated bus information.
10. **Accountant to Generate Reports**: Request for financial and operational data.
11. **Generate Reports to Accountant**: Financial and operational reports.

Novelty Journals
ISSN 2394-7314
International Journal of Novel Research in Computer Science and Software Engineering
Vol. 11, Issue 3, pp: (1-28), Month: September - December 2024, Available at: www.noveltyjournals.com

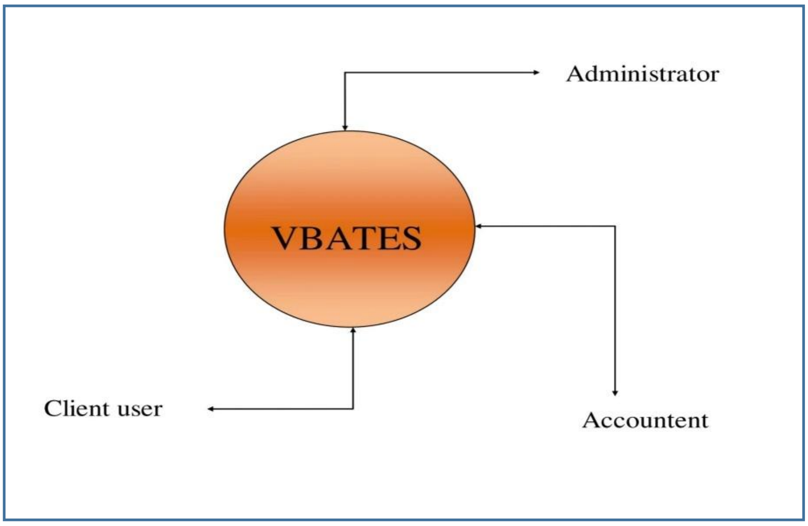


Fig 1: Data Flow Diagram

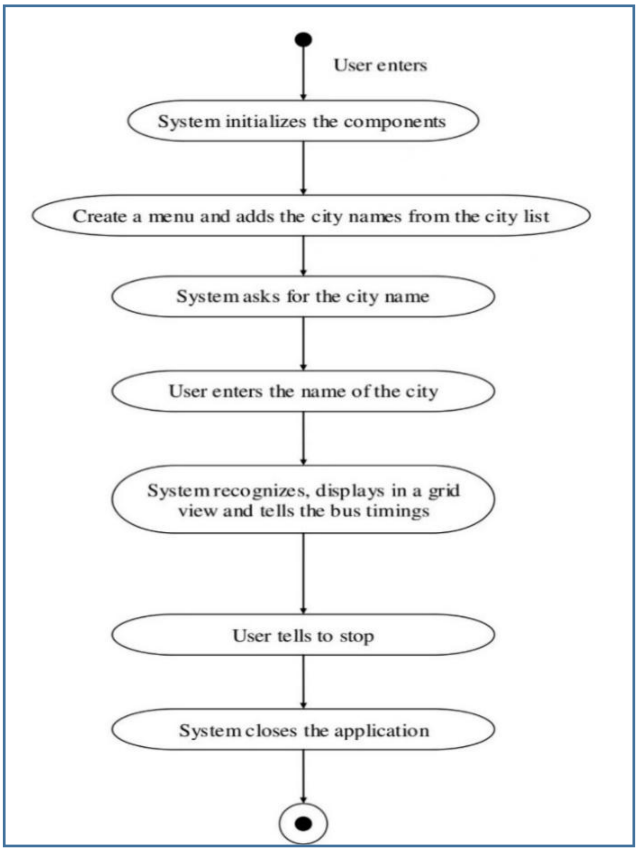


Fig 1.2: User Flow Diagram

2.2 Data dictionary of the target system

This data dictionary covers the main entities and attributes of the Automating Transport Inquiries System based on the information provided in the search results. It includes tables for managing users, places, bus types, buses, platforms, routes, timings, voice commands, and enquiry logs. The relationships between these entities allow the system to process voice commands and provide accurate transport information to users.

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 11, Issue 3, pp: (1-28), Month: September - December 2024, Available at: www.noveltyjournals.com

Table 2.1 BUS

	Id	Name	Bus Type	Total Seats	Available Seats
1	10	babu	7	54	23
2	11	rakib	6	56	34

Table 2.2 Bus Type

	Id	Name	Picture
1	6	Normal	BusTypy\Blackdog.jpg
2	7	Delux	BusType\WhiteSnow.jpg

Table 2.3 CREATE NEW USER

	Id	Name	Password	repassword	U_Id
1	11	Bbu	Xrz	Xrz	1

Table 2.4 PLACES

	Id	PLace	Pictures
1	10	Yangzhou	Places\Waterlilies.jpg
2	11	Nanjing	Places\Winter.jpg

Table 2.5 Timing

	Id	FromPLace	ToPLa..	Time	Day	BusID	Platform	Via	RoadMap
1	11	10	11	5	SunDay	10	13	as, er	RootsMaps\Waterlilies.jpg
2	12	11	10	6	SunDay	11	14	Sd,dsf	RootMaps\Winter.jpg

1. **User Table**

- U_Id (Primary Key): Unique identifier for each user
- U_Name: User's name
- Password: User's password
- Role: User's role (e.g., Administrator, Accountant, Client)

2. **Place Table**

- Place_Id (Primary Key): Unique identifier for each place
- Place_Name: Name of the place
- Picture: Image file name for the place

3. **Bus Type Table**

- BusType_Id (Primary Key): Unique identifier for each bus type
- Name: Name of the bus type (e.g., Express, Local)
- Picture: Image file name for the bus type

4. **Bus Table**

- Bus_No (Primary Key): Unique identifier for each bus
- BusType_Id (Foreign Key): References Bus Type table
- Bus_Name: Name of the bus
- Total_Seats: Total number of seats in the bus
- Picture: Image file name for the bus

5. **Platform Table**

- Platform_Id (Primary Key): Unique identifier for each platform
- Platform_No: Platform number
- Platform_Name: Name of the platform
- Picture: Image file name for the platform

6. **Route Table**

- Route_Id (Primary Key): Unique identifier for each route
- From_Place: Starting place of the route
- To_Place: Destination place of the route
- Via: Intermediate stops on the route

7. **Timing Table**

- Timing_Id (Primary Key): Unique identifier for each timing entry
- Bus_No (Foreign Key): References Bus table
- Route_Id (Foreign Key): References Route table
- Day: Day of the week
- Time: Departure time
- Available_Seats: Number of available seats

8. **Command Table**

- Command_Id (Primary Key): Unique identifier for each voice command
- Command_Text: The actual voice command text
- Command_Type: Type of command (e.g., Place, BusType, BusList)
- Description: Description of what the command does

9. Enquiry Log Table

- Log_Id (Primary Key): Unique identifier for each enquiry log
- U_Id (Foreign Key): References User table
- Enquiry_Time: Timestamp of the enquiry
- Command_Used: Voice command used
- Result: Result of the enquiry

Novelty Journals
ISSN 2394-7314
International Journal of Novel Research in Computer Science and Software Engineering
Vol. 11, Issue 3, pp: (1-28), Month: September - December 2024, Available at: www.noveltyjournals.com

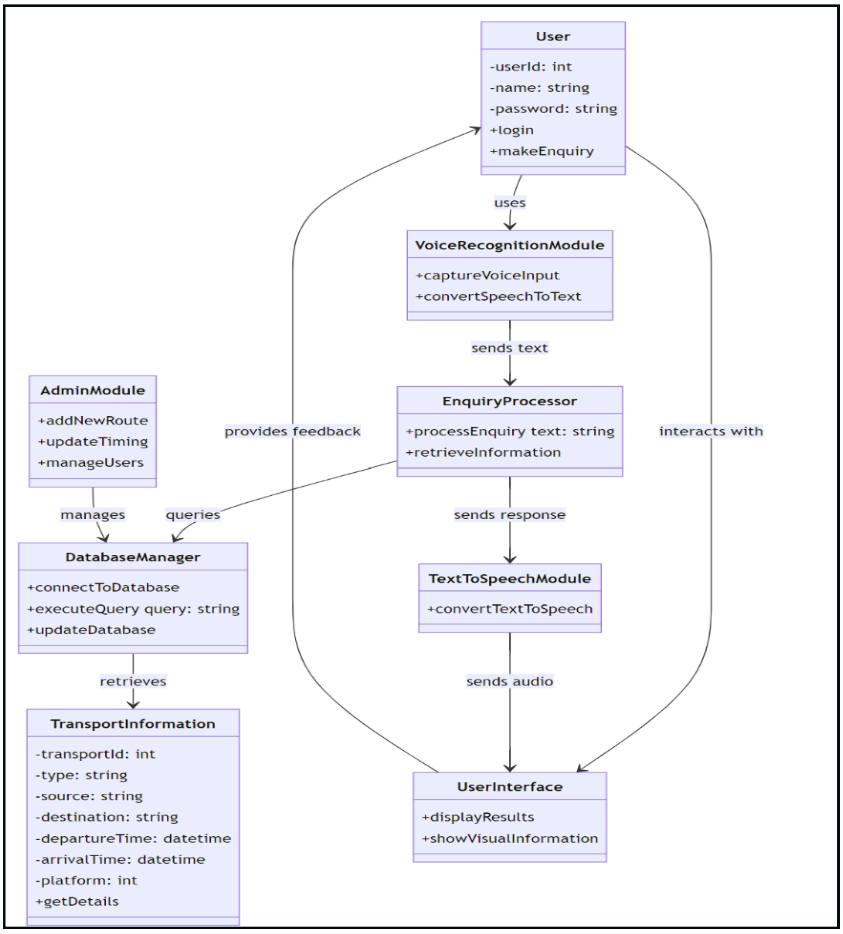


Fig 2.2.1: Class Diagram

2.3 E-R diagram of the target system

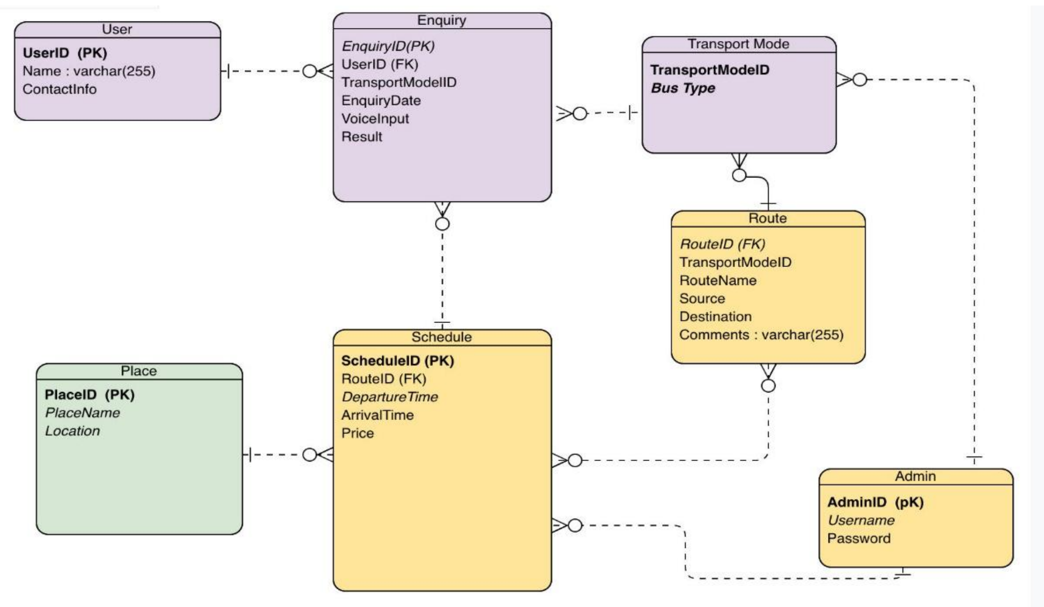


Fig 2.2.2: ER Diagram

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 11, Issue 3, pp: (1-28), Month: September - December 2024, Available at: www.noveltyjournals.com

Here is a description of the entities and relationships, followed by an analysis of its normalization to 3NF:Entities and Relationships:

1. VBATES (Voice Based Automated Transport Enquiry System): The central entity

- Has relationships with Administrator, Accountant, and Client

2. Administrator:

- Can create new users
- Can view all users
- Can view timings
- Can manage places

3. Accountant:

- Can manage bus types
- Can manage buses
- Can add timings

4. Client:

- Can select place
- Can select bus type
- Can select bus number

5. Platform:

- Associated with VBATES

6. Bus Type:

- Has attributes: Name, Picture, Total seats

7. Buses:

- Has attributes: B_No, B_Name, B_Type, Picture, Available seats, From, To, Via, Timings

8. Place:

- Has attributes: Name, Picture

9. Timings:

- Has attributes: Day, Time

Analysis for 3NF:

1. First Normal Form (1NF):

- All attributes contain atomic values
- There are no repeating groups

2. Second Normal Form (2NF):

- The schema is in 1NF
- All non-key attributes are fully functionally dependent on their primary key

3. Third Normal Form (3NF):

- The schema is in 2NF
- There are no transitive dependencies of non-prime attributes on the primary key

Proof of 3NF:

1. Each entity has a clear primary key (e.g., B_No for Buses, Name for Bus Type and Place)
2. All attributes in each entity depend only on their respective primary keys
3. There are no apparent transitive dependencies within the entities
4. Relationships between entities are represented using foreign keys (implied by the connections in the ER diagram)
5. Multivalued attributes (like Timings for Buses) are separated into their own entity with a many-to-one relationship

3. SYSTEM DESIGN

The Voice-Based Automatic Transport Enquiry System can effectively capture and process user voice commands, retrieve relevant transport information, and provide it in both voice and visual formats. The design ensures that the system is user-friendly, accessible, and capable of handling multiple simultaneous queries.

1. Architectural Design

The architectural design of the Voice-Based Automatic Transport Enquiry System involves defining the structure of the system, its components, and their interactions. The system is divided into several key components, each responsible for specific functions:

1. **Speech Recognition Component**:

- Captures and processes the user's voice commands.
- Converts audio input into text using speech recognition algorithms.

2. **Search Component**:

- Takes the processed text command and queries the database to retrieve relevant transport information.
- Passes the results to the display and speech components.

3. **Display Component**:

- Displays the retrieved information on a screen, providing a visual representation of the data.

4. **Speech Component**:

- Converts the retrieved information into voice output using text-to-speech technology.
- Ensures that the user receives the information audibly.

5. **Administration Component**:

- Allows maintenance personnel to update the system's database with new information, commands, and route timings.
- Ensures that the system remains up-to-date and accurate.

2. User Interface Design

Good user interface design is critical to the success of the system. The interface should be intuitive and accessible, especially for individuals with disabilities. Key aspects of the user interface design include:

1. **Voice Command Interface**:

- Users interact with the system using voice commands.
- The system captures and processes these commands to retrieve the required information.

2. **Visual Display**:

- The system displays the retrieved information on a screen.
- This dual-mode of information delivery ensures that users can verify the information visually.

3. **Navigation**:

- Users can navigate through the retrieved results using voice commands.
- The system allows users to move to the previous or next result, change the query, or stop the current query.

4. **Command Management**:

- Users can add new commands, edit existing commands, and update route timings through the system's administration interface.

3. Sequence Diagram

The sequence diagram illustrates the interactions between different components of the system over time. It shows how the system processes a user's voice command and provides the required information.

1. **User Interaction**:

- The user provides a voice command to the system.
- The system captures the voice input and processes it using speech recognition.

2. **Command Processing**:

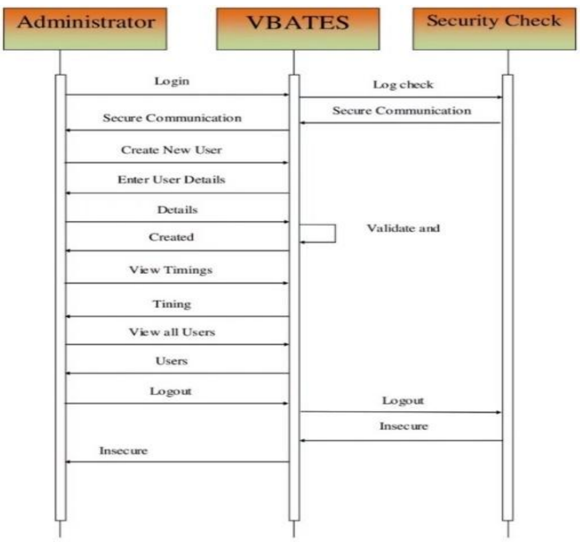
- The recognized command is sent to the search component.
- The search component queries the database for relevant transport information.

3. **Information Delivery**:

- The retrieved information is converted to voice output and displayed on the screen.
- The user can navigate through the results using additional voice commands.

4. **Administration**:

- Maintenance personnel can update the system's database through the administration interface.
- This ensures that the information remains current and accurate.



International Journal of Novel Research in Computer Science and Software Engineering
Vol. 11, Issue 3, pp: (1-28), Month: September - December 2024, Available at: www.noveltyjournals.com

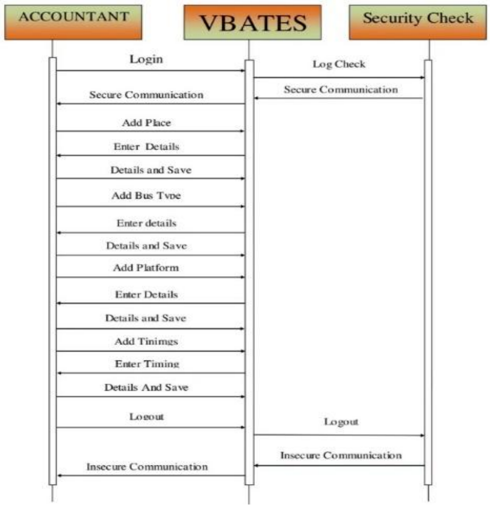


Fig 3.1: SEQUENCE DIAGRAM of Admistrator

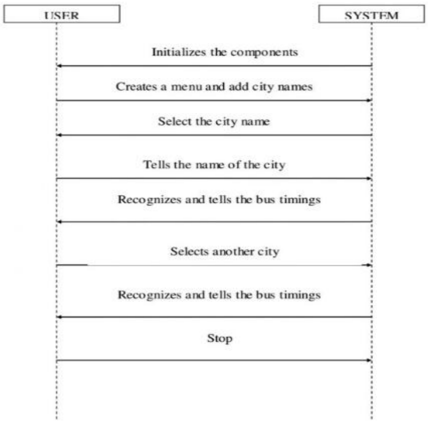


Fig 3.2: SEQUENCE DIAGRAM Of User

4. Database Design

The database design involves defining the structure of the database tables and their relationships. Key tables in the database include:

1. **User Table**:

- Stores user information, including user ID, name, password, and role.

2. **Place Table**:

- Stores information about various places, including place ID, name, and image.

3. **Bus Type Table**:

- Stores information about different types of buses, including bus type ID, name, and image.

4. **Bus Table**:

- Stores detailed information about each bus, including bus number, type, name, total seats, and image.

5. **Platform Table**:

- Stores information about platforms, including platform ID, number, name, and image.

6. **Route Table**:

- Stores information about routes, including route ID, starting place, destination, and intermediate stops.

7. **Timing Table**:

- Stores bus timings, including timing ID, bus number, route ID, day, time, and available seats.

8. **Command Table**:

- Stores voice commands that the system can recognize and respond to, along with their descriptions.

9. **Enquiry Log Table**:

- Stores logs of user enquiries, including log ID, user ID, enquiry time, command used, and result.

3.1 System function analysis

User Information Function Analysis

The user information function of the Voice-Based Automated Transport Enquiry System is designed to provide users with accurate and timely information about transportation options. This function includes several key features:

1. **Voice Commands**:

- Users can interact with the system using voice commands. The system recognizes these commands and processes them to retrieve the required information from the database.

2. **Information Retrieval**:

- The system retrieves information such as bus routes, timings, available seats, and platform details from the database based on the user's query.

3. **Voice Output**:

- The retrieved information is converted to voice output using text-to-speech technology. This allows users to receive the information audibly.

4. **Visual Display**:

- In addition to voice output, the system also displays the retrieved information on a screen. This dual-mode of information delivery ensures that users can verify the information visually.

5. **Navigation**:

- Users can navigate through the retrieved results using voice commands. The system allows users to move to the previous or next result, change the query, or stop the current query.

6. **Command Management**:

- Users can add new commands, edit existing commands, and update route timings through the system's administration interface.

3.2 System Operting Fnvironment

The detailed system operating environment the Automating Transport Inquiries System guarantees optimal system performance and user satisfaction. By complying with the necessary hardware and software prerequisites, the system may efficiently collect and process voice commands, acquire pertinent transport information, and provide it in both auditory and visual modes. This environment facilitates the system's objective of enhancing efficiency and user experience at transportation terminals, rendering it a vital instrument for users in search of prompt and precise transport information.

ISSN 2394-7314
International Journal of Novel Research in Computer Science and Software Engineering
Vol. 11, Issue 3, pp: (1-28), Month: September - December 2024, Available at: www.noveltyjournals.com

Hardware Requirements

1. **Processor**:

- **Minimum**: Pentium IV
- **Recommended**: Intel Core i3/i5/i7 or higher for better performance and faster processing of voice commands and database queries.

2. **Memory (RAM)**:

- **Minimum**: 128 MB
- **Recommended**: 4 GB or higher to ensure smooth operation, especially when handling multiple simultaneous queries.

3. **Storage**:

- **Minimum**: 20 GB of secondary storage
- **Recommended**: 500 GB or higher to accommodate extensive transport data, logs, and future scalability.

4. **Display**:

- Monitor: SVGA or higher resolution for clear visual output.

5. **Audio Devices**:

- High-quality microphone for capturing voice commands accurately.
- Speakers for delivering voice output clearly.

6. **Additional Peripherals**:

- Keyboard and mouse for administrative tasks and system maintenance.

Software Requirements

1. **Operating System**:

- Windows 7 or higher (Windows 10 recommended for better compatibility, security, and performance).

2. **Development Environment**:

- Visual Studio 2010 or higher for development and debugging.

3. **Frameworks and Libraries**:

- .NET Framework 2.0 or higher for building and running the application.
- Python .NET (pythonnet) for integrating Python with .NET libraries, allowing the use of Python for scripting and .NET for core functionalities.

4. **Speech Recognition and Text-to-Speech**:

- Microsoft Speech SDK for speech recognition and text-to-speech functionalities.
- Python libraries like SpeechRecognition for capturing and processing voice commands.
- pyttsx3 or gTTS for converting text responses into speech.

5. **Database Management**:

- SQL Server for managing transport information, ensuring efficient data retrieval and updates.
- Python libraries like pyodbc or SQLAlchemy for database interactions.

6. **Web Server**:

- Internet Information Services (IIS) 5.1 or above for hosting web components and ensuring reliable web service delivery.

7. **Web Browsers**:

- Internet Explorer, Google Chrome, or any modern web browser for accessing web-based interfaces.

Functional Requirements

1. **Voice Command Processing**:

- The system should accept and process voice commands accurately.
- Convert voice input to text using speech recognition technology.

2. **Database Interaction**:

- Search and retrieve appropriate transport information based on voice commands.
- Store and manage transport-related data (routes, timings, etc.).

3. **Information Output**:

- Display retrieved information on a screen for visual verification.
- Convert text information to voice output for auditory delivery.

4. **User Navigation**:

- Allow users to navigate through results using voice commands.
- Enable users to change or stop queries as needed.

5. **Administration**:

- Provide options for administrators to add new commands and edit route timings.
- Allow maintenance of transport information to ensure data accuracy and currency.

Non-Functional Requirements

1. **Performance**:

- Quick processing of voice commands and information retrieval.
- Ability to handle multiple simultaneous queries efficiently.

2. **Usability**:

- User-friendly interface accessible to a wide range of users, including those with disabilities.
- Clear voice recognition with minimal background noise interference.

3. **Reliability**:

- Consistent and accurate information provision.
- Robust error handling and recovery mechanisms.

4. **Scalability**:

- Ability to handle an increasing number of users and queries.
- Scalable architecture to support future growth and additional features.

5. **Security**:

- Secure storage and handling of transport data.
- Protection against unauthorized access and data breaches.

6. **Maintainability**:

- Easy updates to transport data, routes, and timings.
- Simple system maintenance and troubleshooting procedures.

3.3 System strueture

This structure capitalizes on the advantages of both environments - Python's exceptional libraries for voice and language processing, and .NET's strong application foundation and user interface capabilities. The Python.NET bridge facilitates seamless interoperability between the Python and .NET environments, resulting in a robust and adaptable solution.

1. Core Python Components:

- Speech Recognition Module: Uses Python libraries like SpeechRecognition to capture and interpret voice commands from users.
- Natural Language Processing (NLP) Module: Processes the recognized speech to understand user intent and extract key information.
- Database Interface: Handles querying and retrieving transport information from a SQL database using a library like SQLAlchemy.
- Text-to-Speech Module: Converts text responses back into speech using a library like pyttsx3.

2. .NET Integration Layer:

- Python.NET Bridge: Allows seamless integration between the Python code and .NET framework.
- C# Wrapper Classes: Provide an interface for .NET components to interact with the Python modules.

3. .NET Components:

- User Interface: Built using Windows Forms or WPF to display visual information and provide a graphical interface for users.
- Application Logic: Handles the flow of the application, coordinating between user inputs, Python processing, and output display.
- Data Visualization: Uses .NET charting libraries to display routes, schedules, or other relevant information graphically.

4. Shared Resources:

- Configuration Files: Store settings for both Python and .NET components.
- Logging System: Captures errors and important events from both environments.

5. External Integrations:

- GPS Module: Potentially integrates with external GPS services for real-time tracking.
- Payment Gateway: If ticket booking is a feature, this would handle secure transactions.

The system would operate by:

1. Capturing voice input through the .NET UI
2. Passing the audio data to the Python speech recognition module

Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 11, Issue 3, pp: (1-28), Month: September - December 2024, Available at: www.noveltyjournals.com

3. Processing the recognized text using NLP
4. Querying the database based on the interpreted command
5. Generating a response in Python
6. Converting the response to speech
7. Sending the audio and any visual data back to the .NET UI for presentation to the user

4. DETAILED DESIGN

4.1 Login Module:

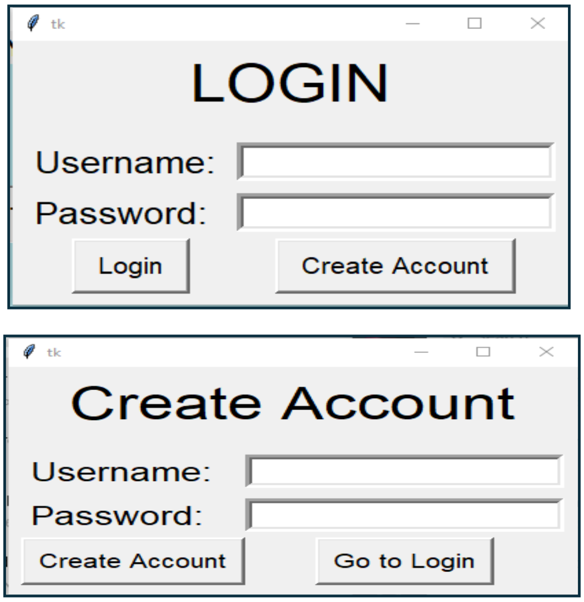
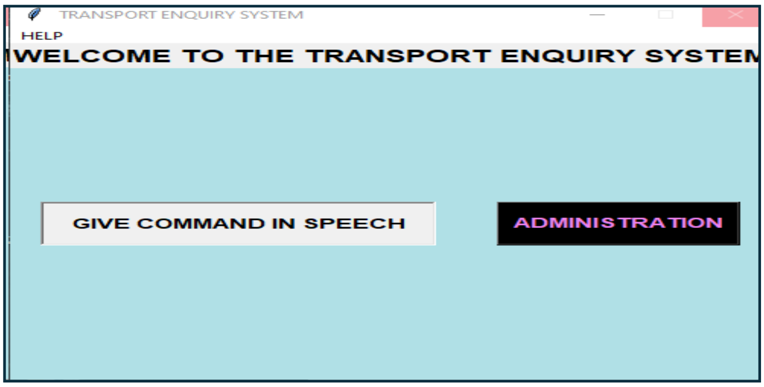
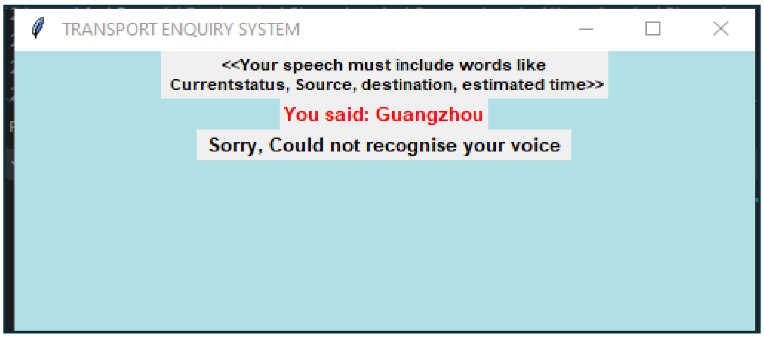


Fig 4.1: Log in Module

4.2 Querying a Module:





Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 11, Issue 3, pp: (1-28), Month: September - December 2024, Available at: www.noveltyjournals.com

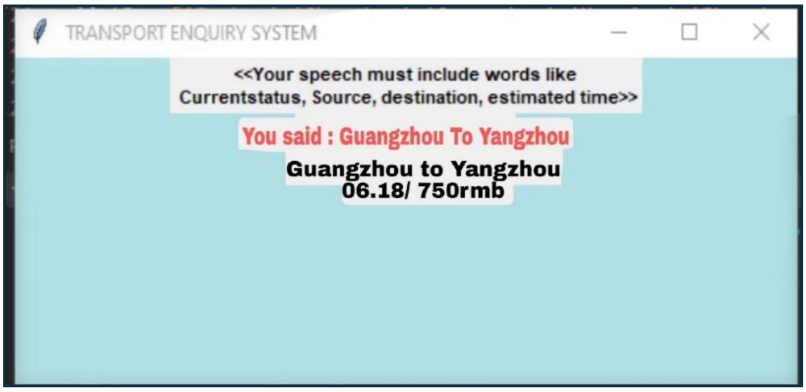


Fig 4.2: Querying a Module

4.3 Administrator Module.

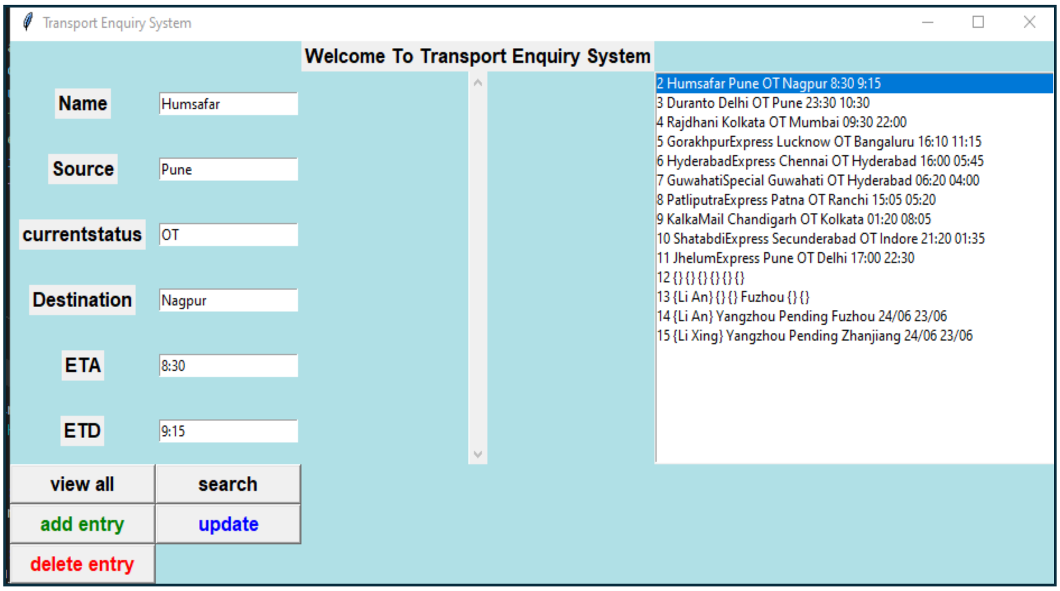


Fig 4.3: Administrator Module

5. TEST

The Automating Transport Inquiries System is thoroughly validated to ensure it meets all functional and non-functional requirements by following this structured test plan, method, and detailed test cases

5.1 Test Plan

The test plan outlines the strategy, objectives, resources, and schedule for the testing activities. It ensures that all aspects of the Automating Transport Inquiries System are thoroughly tested to meet the requirements and function correctly.

Objectives:

- Firstly we need to verify that the system accurately recognizes and processes voice commands.
- Ensuring the system retrieves and displays correct transport information.
- Validation the integration between Python.NET components and .NET framework.
- Testing the user interface for usability and accessibility.

Resources:

- Test environment with necessary hardware (microphone, speakers etc.).
- Test data including various transport schedules and routes.
- Tools for automated and manual testing.

5.2 Test Method

The test method describes the approach and techniques used to test the system.

Approach:

- **Unit Testing:** Each speech recognition, NLP, database interface is tested individually to ensure they function correctly.
- **Integration Testing:** We have tested the interaction between Python components and .NET components to ensure they work together seamlessly.
- **System Testing:** Conducted end-to-end testing of the entire system to verify it meets the specified requirements.
- **User Acceptance Testing (UAT):** Involving end-users to validate the system's functionality and usability in a real-world scenario.

Techniques:

- **Black Box Testing:** Focus on input and output without considering internal code structure.
- **White Box Testing:** Examine the internal workings of the system, especially for unit testing.
- **Automated Testing:** Use tools to automate repetitive tests, especially for regression testing.
- **Manual Testing:** Perform exploratory testing to identify issues that automated tests might miss.

5.3 Test Items and Test Cases

Test Items

1. **Voice Recognition Accuracy**

  - Ensure the system accurately recognizes and processes voice commands.

  - Test for different accents, speech speeds, and background noise levels.

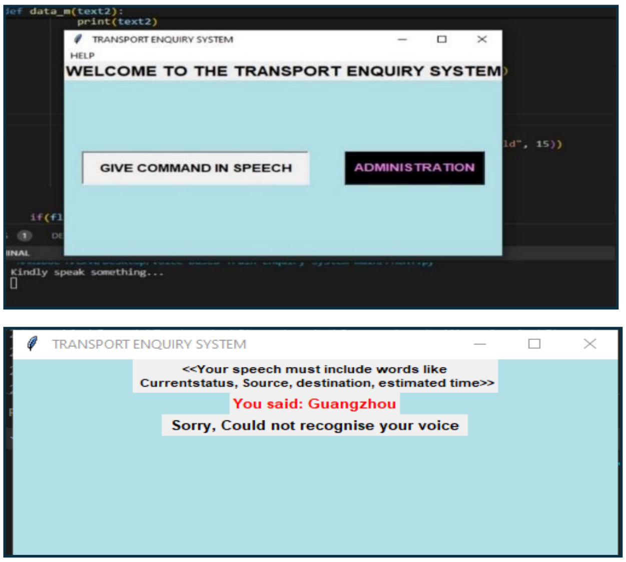


Fig 5.3.1: Voice Recognition Accuracy

2. **Query Processing**

- Validate the system's ability to process and respond to various transport-related queries.
- Test for different types of queries such as route information, schedule timings, and fare details.

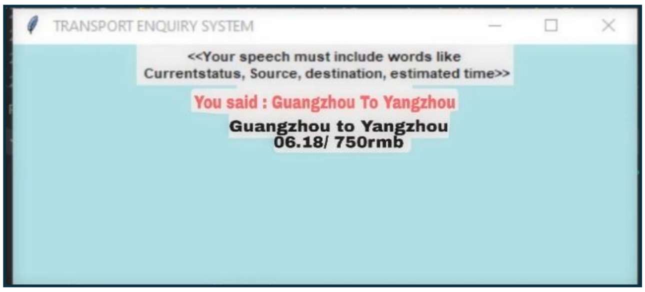


Fig 5.3.2: Query Processing

3. **Database Interaction**

- Ensure the system correctly retrieves and updates information in the database.
- Test the addition, modification, and deletion of transport data by the admin.

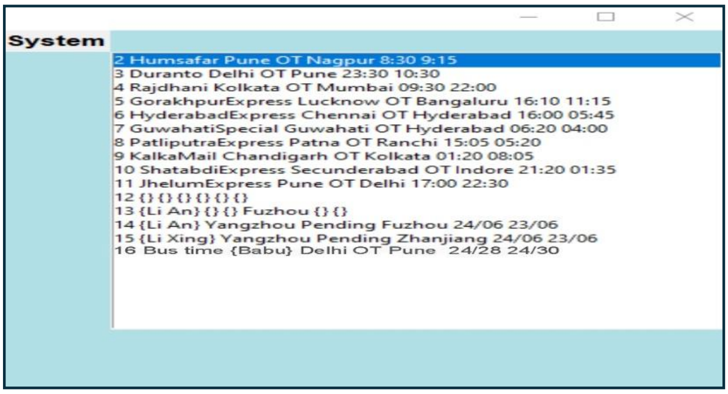


Fig 5.3.3: Database Interaction

4. **Response Time**

- Measure the time taken by the system to respond to user queries.
- Ensure the response time is within acceptable limits for a smooth user experience.

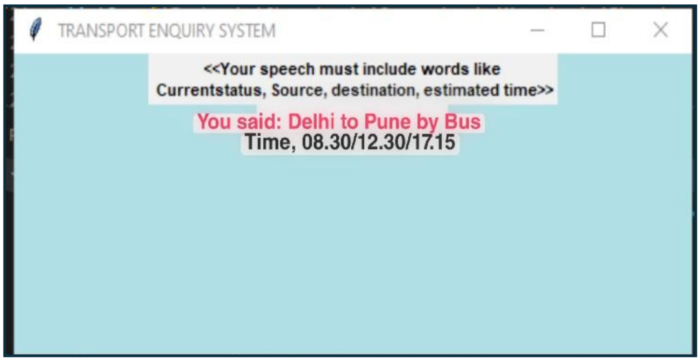


Fig 5.3.4: Response Time

Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 11, Issue 3, pp: (1-28), Month: September - December 2024, Available at: www.noveltyjournals.com

The response time depends on the availability of the transport system. If there is many options it takes around 12/13 seconds. For less options like this one it takes around 5~6 sec.

5. **User Interface**

- Verify the display of information on the screen.

- Test the navigation through results using voice commands.

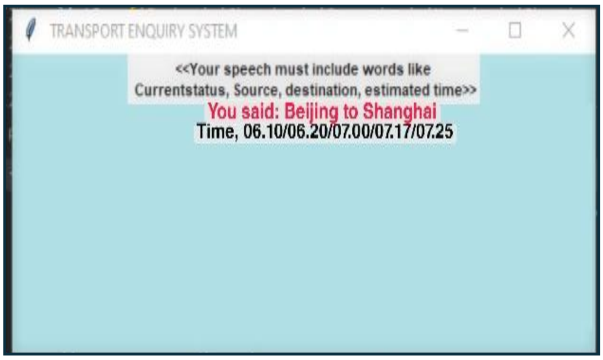


Fig 5.3.5: User Interface

6. **Error Handling**

- Ensure the system handles unrecognized commands gracefully.

- Test the system's response to invalid inputs and network failures.

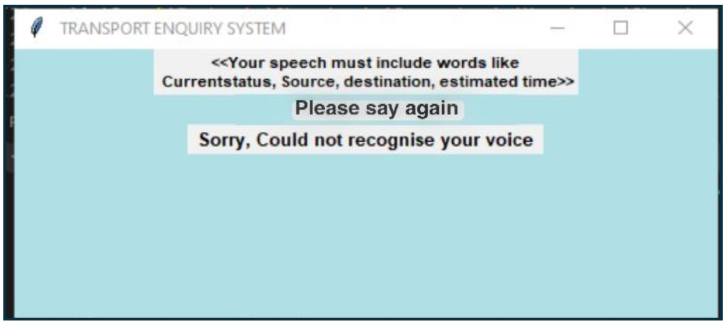


Fig 5.3.6: Error Handling**

6. TEST CASES

1. **Voice Recognition Accuracy**

- **Test Case 1.1**: Speak a query in a clear, standard accent.

  - **Expected Result**: The system accurately recognizes and processes the query.

- **Test Case 1.2**: Speak the same query with a different accent.

  - **Expected Result**: The system accurately recognizes and processes the query.

- **Test Case 1.3**: Speak the query in a noisy environment.

  - **Expected Result**: The system accurately recognizes and processes the query or prompts for a clearer input.

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 11, Issue 3, pp: (1-28), Month: September - December 2024, Available at: www.noveltyjournals.com

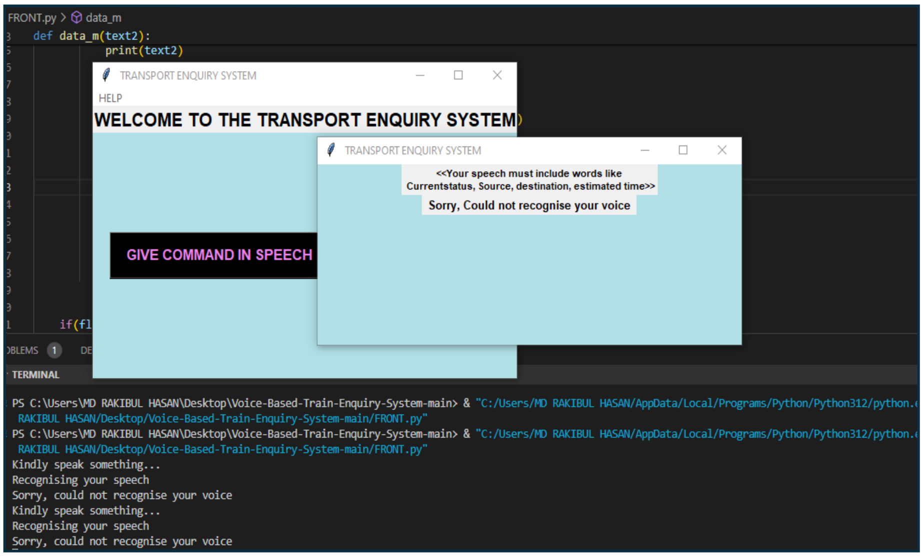


Fig 6.1: Voice Recognition Accuracy

This is a real life based system for helping the people in need. Currently it has some noise issue. User must use noise cancelation device or microphone to input the voice.

2. **User Authentication**

- **Test Case 2.1**: Attempt to log in with correct credentials.

  - **Expected Result**: Successful login.

- **Test Case 2.2**: Attempt to log in with incorrect credentials.

  - **Expected Result**: Login failure with an appropriate error message.

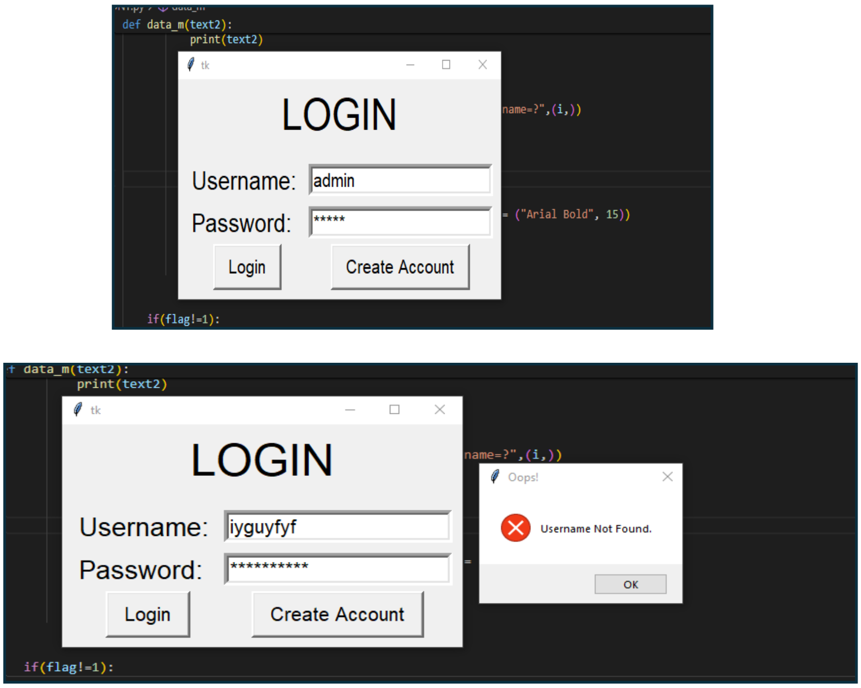


Fig 6.2: User Authentication

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 11, Issue 3, pp: (1-28), Month: September - December 2024, Available at: www.noveltyjournals.com

3. **Query Processing**

- **Test Case 3.1**: Query for bus route information.
  - **Expected Result**: The system provides accurate route details.
- **Test Case 3.2**: Query for schedule timings.
  - **Expected Result**: The system provides accurate schedule timings.
- **Test Case 3.3**: Query for fare details.
  - **Expected Result**: The system provides accurate fare details.

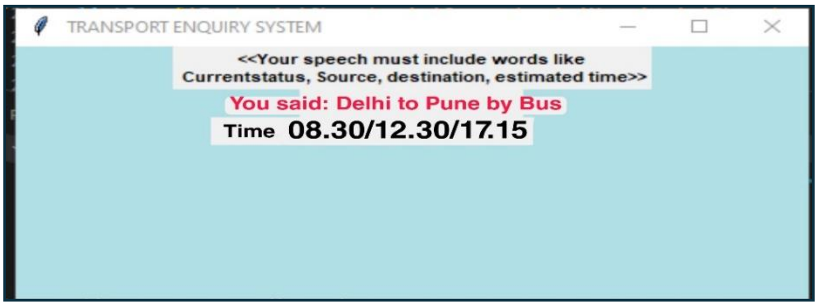
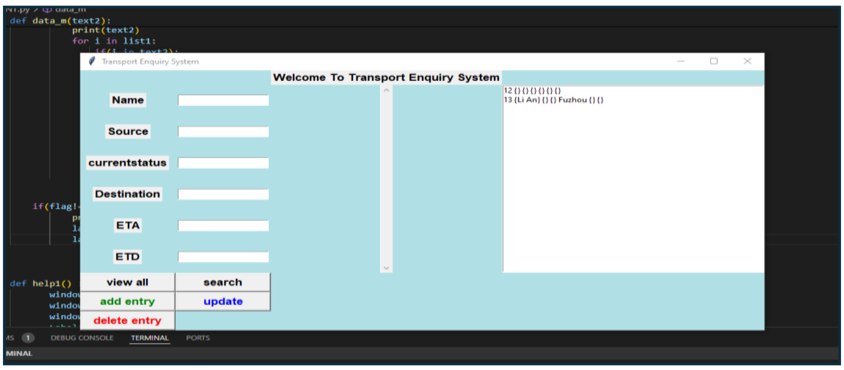


Fig 6.3: Query Processing

4. **Database Interaction**

- **Test Case 4.1**: Admin adds a new bus route.
  - **Expected Result**: The new route is successfully added and retrievable.
- **Test Case 4.2**: Admin updates an existing schedule.
  - **Expected Result**: The schedule is updated correctly in the database.
- **Test Case 4.3**: Admin deletes a bus route.
  - **Expected Result**: The route is successfully deleted from the database.



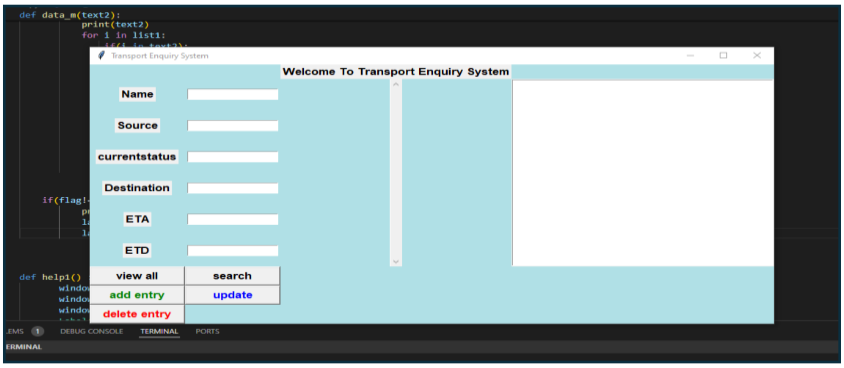


Fig 6.4: Database Interaction

7. CONCLUSION

The Automating Transport Inquiries offers a transformative solution to the ongoing challenges faced by travelers at transport terminals, utilizing advanced voice recognition technology to streamline access to essential transport information and enhance user experience. The feasibility study confirms that this innovative system is not only technically viable but also economically and operationally sound, allowing for effective implementation with existing technologies such as Python.NET and Microsoft Speech SDK. By prioritizing accessibility, the system is particularly advantageous for individuals with visual impairments or those who struggle with traditional text-based interfaces, providing a 24/7 service that reduces wait times and minimizes human error. Its scalability ensures that it can efficiently handle multiple queries at busy transport hubs. However, it is vital to address potential social implications, including privacy concerns and the impact on employment within the transport sector, by ensuring transparency in data handling and retraining staff for new roles. Additionally, accommodating diverse user demographics through multi-language support will foster inclusivity. Ultimately, this approach not only meets current demands but also sets a new standard for public service technology, significantly improving how users interact with transport systems. By embracing automation and voice technology, we can make information retrieval faster, easier, and more accessible for everyone, paving the way for a more efficient and user-friendly transportation experience that benefits all members of society.

REFERENCES

[1] Celikbilek, Y., & Erol, F. (2021). A comprehensive review of speech recognition for transportation systems. IEEE Access, 9, 99-120.

[2] Teixeira, A., Ferreira, F., Almeida, N., Rosa, A., Casimiro, J., Silva, S., ... & Freitas, D. (2019). Speech-centric multimodal interaction for easy-to-access online services–A personal life assistant for the elderly. Procedia Computer Science, 154, 263-270.

[3] Lau, J., Zimmerman, B., & Schaub, F. (2018). Alexa, are you listening? Privacy perceptions, concerns and privacy-seeking behaviors with smart speakers. Proceedings of the ACM on Human-Computer Interaction, 2(CSCW), 1-31.

[4] Dierbach, C. (2020). Introduction to computer science using Python: A computational problem-solving focus. John Wiley & Sons.

[5] https://www.intechopen.com/books/speech-recognition-systems/speech-recognition-techniques-and-applications

[6] https://towardsdatascience.com/a-beginners-guide-to-speech-recognition-with-python-1f2c4b4e5b1d

[7] https://www.analyticsvidhya.com/blog/2020/11/a-beginners-guide-to-speech-recognition-in-python/

[8] https://towardsdatascience.com/introduction-to-speech-recognition-in-python-dccf8e7a6e1d

[9] https://medium.com/@davidmichaels/the-future-of-transportation-is-voice-technology-cb8f8c2a7c8b