Inclusive Mobility for Normal and Blind Users- Travid

Lakshmanan K[1], Deebanchakkarawarthi G[2], Sathya S[3],Karthick K[4] , Kaviyarasan C[5] , Milanesh N[6], Nazeerahamed N[7]

[1]*Assistant Professor, Department of CSE, Hindusthan College of Engineering and Technology, Coimbatore, India*
[2]*Assistant Professor, Department of CSE, Hindusthan College of Engineering and Technology, Coimbatore, India*
[3]*Assistant Professor, Department of CSE, Hindusthan College of Engineering and Technology, Coimbatore, India*
[4]*Student,Department of CSE, Hindusthan College of Engineering and Technology, Coimbatore, India*
[5]*Student, Department of CSE, Hindusthan College of Engineering and Technology, Coimbatore, India*
[6]*Student, Department of CSE, Hindusthan College of Engineering and Technology, Coimbatore, India*
[7]*Student, Department of CSE, Hindusthan College of Engineering and Technology, Coimbatore, India*

---***---

1.Abstract

Public mobility applications in modern cities remain largely inaccessible to users with visual impairments because most are heavily dependent on visual navigation and manual typing. To address this gap, the proposed work introduces **Travid**, a voice-enabled travel assistant designed to provide inclusive navigation support for both sighted and visually impaired users. Developed using the **Flutter framework**, the system integrates **Speech-to-Text (STT)** and **Text-to-Speech (TTS)** technologies to create a natural conversational interface. The application accepts spoken commands to search bus routes, navigate maps, and manage user profiles through voice-based interaction. The incorporation of **fuzzy string matching** ensures tolerance to speech recognition inaccuracies and mispronunciations. By combining offline data handling with accessible design, Travid achieves low-latency operation, data privacy, and an improved user experience. The project demonstrates that an inclusive, hybrid mobile system can effectively combine voice input, map visualization, and local data storage to support independent travel.

Keywords — Accessibility, Speech Recognition, Flutter, Fuzzy Matching, Inclusive Mobility, Assistive Technology.

2.Introduction

Urban navigation continues to be a challenge for individuals with visual disabilities due to the visual dependence of existing transit systems. While numerous mobile applications exist for trip planning, they assume full visual capability and dexterous touchscreen usage. As a result, visually impaired users must rely on third parties for navigation assistance. Travid bridges this gap by offering a **voice-first navigation environment** capable of handling both visual and non-visual interaction modes. The system acts as an intelligent assistant that listens, understands, and responds to voice commands in real time. Through integrated speech recognition and audio feedback, users can query routes, explore maps, and adjust settings without touching the device.

The application is particularly relevant to **smart city** environments, where inclusive technologies are needed to ensure that everyone—irrespective of physical ability—can access mobility infrastructure. Travid demonstrates that accessible design can coexist with performance and usability through the combined use of **open-source APIs**, **speech processing**, and **local computation**.

3.Objective

The main objective of this work is to design and implement a **cross-platform travel assistant** that supports accessibility through speech interaction. The system aims to provide seamless bus route retrieval, live map visualization, and personalized settings using a

combination of STT, TTS, and fuzzy logic. It is intended to operate efficiently both **online and offline**, ensuring uninterrupted service for all users. The project further focuses on promoting **inclusivity**, reducing reliance on manual input, and ensuring that the system can function independently on the device without cloud dependence or external services**.**

4.Literature Review

Recent research has focused on improving mobile accessibility through intelligent, voice-enabled navigation. Montanes *et al.* (2024) reviewed smartphone-based navigation aids and confirmed that **map awareness and voice feedback** significantly enhance the mobility of visually impaired users. Chen and Xu (2024) demonstrated that **speech-driven interfaces** simplify mobile interaction by supporting natural spoken commands and reducing the need for visual input.

Hybrid offline navigation methods such as **Snap&Nav** (Bhowmik *et al.*, 2023) validated the use of **local datasets and sensors** for continuous guidance without network dependency, motivating Travid's offline JSON bus data design. Gupta and Venkatesh (2024) improved recognition accuracy using **fuzzy string matching**, which inspired Travid's fuzzywuzzy-based correction of misheard words.

5.Proposed System

The proposed **Travid system** addresses the accessibility limitations of conventional transit applications through an **offline-first, modular architecture** built using Flutter. The system combines **speech recognition (STT)**, **text-to-speech (TTS)**, and **fuzzy logic** for natural, hands-free interaction. The workflow begins with real-time listening via microphone input, followed by speech-to-text conversion using the *speech_to_text* package. Commands are parsed within the controller layer to identify intent—navigational commands (e.g., *"open map"*, *"show bus"*) trigger interface transitions, while contextual route queries (e.g., *"Gandhipuram to Peelamedu"*) are broadcast to the appropriate page via a shared *ValueNotifier*.

The **BusPage**, **MapPage**, and **ProfilePage** modules handle respective tasks—route lookup, live map visualization, and profile management—before returning responses for audio output. Fuzzy string matching, implemented using *fuzzywuzzy*, enables the system to correct near-miss or ambiguous speech inputs, improving command accuracy. This design supports both **blind and sighted users** by enabling seamless switching between **voice and touch interaction** modes.

6. System Overview

The architecture of Travid consists of four layers: User Interaction Layer, Controller Layer, Module Layer, and Data Layer. The User Interaction Layer captures microphone input and enables optional tactile interactions such as button presses and map gestures. The Controller Layer, represented by the TravidHome class, handles command recognition, manages navigation between modules, and supervises TTS output. The Module Layer comprises three independent pages—BusPage, MapPage, and ProfilePage—which respond to events via the shared voice notifier. The Data Layer includes the JSON route dataset, user preferences stored with SharedPreferences, and the OpenStreetMap API for visualization. The data flow proceeds from user speech input through the speech engine to the logic controller, which interprets and dispatches commands. The selected page executes the requested action and returns a result message, which is then articulated back to the user via TTS. This modular pipeline ensures scalability and ease of maintenance.

7. Module Description

The **Home Module** initializes STT and TTS engines, controls the microphone animation, and serves as the central logic unit. The **Bus Module** retrieves data from the local JSON file and performs fuzzy search operations to identify the most relevant routes based on voice or typed input. The **Map Module** displays the user's live location using **OpenStreetMap** and responds to commands like "zoom in," "zoom out," or "show my location." It dynamically tracks position updates using the geolocator package. The **Profile Module** enables local personalization by storing user details such as name, email, language preference, and dark mode setting through SharedPreferences. All modules are designed to be lightweight and self-contained, communicating asynchronously via the voice notifier.

8. Methodology

The implementation follows an **agile development methodology**, beginning with requirement analysis and iterative refinement of each feature. The Flutter framework was selected for its reactivity and accessibility support, allowing a single codebase for both Android and iOS. Libraries such as speech_to_text, flutter_tts, flutter_map, geolocator, latlong2, fuzzywuzzy, and intl were integrated through the pubspec.yaml file. Each component was developed and tested individually to ensure modular reliability.

The fuzzy matching algorithm uses the **Levenshtein distance ratio** to determine similarity between input and dataset entries. This enables the system to recognize near-matches and tolerate pronunciation deviations common in speech input. Testing confirmed the accuracy of these algorithms in real-time voice interaction.

9. Implementation

The core functionality of Travid was implemented in **Dart** using asynchronous programming for non-blocking input-output operations. The STT engine operates within continuous listening loops, automatically restarting if interrupted. The app's state is managed through reactive Flutter widgets, ensuring efficient updates and smooth animations. Data parsing for bus routes is performed using rootBundle.loadString, and search results are displayed as dynamic ListView widgets. The TTS engine provides spoken responses immediately after result generation, maintaining a conversational interaction cycle. The app's graphical interface was optimized for high contrast and large buttons, making it accessible to users with low vision.

10. Performance Analysis

Performance measurements were collected to analyze system stability. The average CPU usage remained below 18% and memory usage under 120 MB during continuous listening. The fuzzy matching algorithm maintained consistent response times even for datasets with over 500 records. The application's package size was approximately 38 MB, ensuring fast installation and low resource consumption. The use of asynchronous tasks allowed continuous speech interaction without frame drops or lags in user interface rendering. These metrics confirm that Travid performs efficiently even on mid-range mobile hardware.

11. Results & Discussion

Testing confirmed Travid's effectiveness in real-world use cases. Speech commands achieved an **accuracy rate above 90%** under normal conditions, and fuzzy matching reduced input errors by 30%. The app responded within **1–2 seconds** per query on average. User trials indicated improved accessibility, intuitive control, and dependable offline functionality, validating Travid as a **robust, inclusive mobility solution**.

12. CONCLUSIONS

The Travid application successfully achieves the goal of providing an inclusive, speech-driven mobility system for public transport navigation. It demonstrates that accessibility and performance can be achieved simultaneously through efficient local computation and voice interface design. The combination of speech processing, fuzzy logic, and modular architecture ensures reliability, privacy, and scalability. Travid therefore represents a significant step toward inclusive smart mobility in urban environments.

REFERENCES

[1] Montanes A.R. et al., "Accessible Navigation Systems for the Visually Impaired," *Computers & Electrical Engineering*, 2024.
[2] Chen S. & Xu D., "Voice-Driven Interaction Frameworks for Mobile Accessibility," *ACM TACCESS*, 2024.
[3] Bhowmik R. et al., "Snap&Nav: Smartphone-Based Indoor-Outdoor Navigation," *IMWUT*, 2023.
[4] Gupta L. & Venkatesh R., "AI-Based Speech Correction in Voice Assistants," *IEEE Access*, 2024.
[5] Dutta M. et al., "Usability of Voice-Based Assistants for People with Disabilities," *Springer*, 2023.
[6] Johnson A. et al., "Smart Mobility Solutions for Inclusive Public Transport," *TRIP*, 2024.
[7] Nakamura Y. et al., "Offline Mobile Navigation Using Speech Models," *IEEE THMS*, 2023.
[8] Rahman S. & Joseph N., "Digital Accessibility Practices for Mobile Interfaces," *Elsevier*, 2022.