VAVI-Voice Assistant for the Visually Impaired

Final Report
02/01/2026

Project Team:

- **Ceyda Kuşçuoğlu:16348076072**
- **Kıvanç Terzioğlu: 27233564574**
- **Berkay Kaan Karaca: 68317070956**

Supervisor:

- **Prof.Dr. Tansel Dökeroğlu**

---

1. Introduction

Visually impaired individuals face significant challenges in indoor navigation, particularly in unfamiliar environments such as university buildings, offices, or public institutions. Traditional navigation systems are often insufficient for indoor use due to limitations in GPS accuracy and the lack of contextual awareness of indoor obstacles and signage.

The VAVI project aims to address these challenges by developing an AI-assisted indoor navigation system that combines mobile technologies, computer vision, artificial intelligence, and audio feedback. The system enables visually impaired users to determine their location, select a source and target destination using either manual or voice-based input, and receive step-by-step navigation instructions through both textual and auditory guidance. In addition, real-time object detection is performed to increase environmental awareness and improve user safety during navigation.

By integrating multiple AI models, a hybrid system architecture, and accessibility-focused design principles, VAVI provides a low-cost, scalable, and user-centered solution for indoor navigation.

---

2. System Architecture Overview

The VAVI system is designed using a **hybrid architecture**, combining on-device processing with backend-supported AI services. This approach allows computationally intensive tasks to be handled efficiently while maintaining real-time responsiveness on the mobile device.

Main System Components

The system consists of the following core components:

- **Mobile Application (Flutter)**
- **Backend Server (FastAPI)**
- **Database (Microsoft SQL Server)**
- **AI Models (YOLO, Gemini API)**
- **Audio Feedback Module**
- **Sensor Module (Camera, Microphone)**
- **External AI APIs**

Each component is responsible for a specific function and communicates with others through well-defined interfaces.

---

3. Detailed System Design and Data Flow

3.1 Mobile Application Layer

The mobile application, developed using **Flutter**, serves as the primary interface between the user and the system. Accessibility is a key design consideration, and the application supports both **manual interaction** (dropdown lists) and **voice-based commands** for selecting the source and target nodes.

At the application entry point, the user selects:

- A **starting location (source node)**
- A **destination (target node)**

These selections can be made either through a dropdown menu or via speech recognition.

3.2 Location Detection via Visual Context

If the user chooses to determine their current location automatically, the **"Find My Location"** feature is activated. In this mode:

1. The mobile application opens the device camera.
2. Approximately **30 image frames** are captured over a short time interval.
3. These frames are sent to the **Gemini API** for visual analysis.
4. The AI model analyzes environmental cues such as **office signs, room labels, or textual indicators** visible in the frames.
5. Based on this analysis, the system estimates the user's current indoor location.

This vision-based location detection approach provides an alternative to traditional sensor-based localization methods, which were explored but later abandoned due to insufficient accuracy.

3.3 Navigation and Path Planning

Once the source and target nodes are determined, the system computes the **shortest path** between the two locations using pre-defined indoor graph data stored in the database. The calculated route is then:

- Displayed to the user as text
- Converted into **audio-based navigation instructions**

This dual-mode feedback ensures usability for visually impaired users with varying preferences.

3.4 Real-Time Object Detection

During navigation, the mobile application continuously displays a half-screen camera preview and activates **real-time object detection** using the **YOLO model**. This module detects nearby objects and potential obstacles, enhancing situational awareness and reducing the risk of collisions.

Detected objects are communicated to the user through **audio alerts**, allowing them to react promptly without needing visual confirmation.

3.5 Backend and Database Layer

The backend system is implemented using **FastAPI**, which handles:

- Data communication between the mobile application and the database
- Route computation requests
- AI service coordination

The **Microsoft SQL Server** database stores:

- Indoor node and edge data
- Location mappings
- Graph structures used for pathfinding

3.6 AI Models and Technologies Used

Multiple AI-based approaches were explored during the development process. For location tracking and prediction, traditional machine learning models such as:

- Random Forest
- K-Nearest Neighbors (KNN)
- Support Vector Machines (SVM)
- XGBoost
- Logistic Regression

were tested. However, due to inadequate accuracy and reliability in real-world indoor environments, these approaches were ultimately discarded in favor of **vision-based location estimation using the Gemini API**.

The final system relies on:

- **YOLO** for real-time object detection
- **Gemini API** for visual scene understanding and location inference

---

4. Impact of Engineering Solutions

4.1 Societal Impact

The primary societal impact of the VAVI project is its contribution to improving the quality of life of visually impaired individuals. Independent indoor navigation is one of the major challenges faced by visually impaired people in daily life, particularly in complex environments such as university buildings, offices, hospitals, and public institutions.

By providing real-time navigation assistance, location awareness, and object detection through audio feedback, VAVI enables visually impaired users to move more confidently and independently without relying on external assistance. This increases personal autonomy, accessibility, and social inclusion, which are essential components of an inclusive and equitable society.

4.2 Economic Impact

From an economic perspective, VAVI is designed as a low-cost solution. The system does not impose any direct financial burden on visually impaired users, as it relies on widely available consumer devices such as smartphones and does not require specialized or expensive hardware.

For organizations and companies that wish to integrate the system into indoor environments, such as universities or office buildings, the deployment cost is relatively low. The solution mainly requires software integration and basic infrastructure data (nodes and paths), making it significantly more affordable compared to commercial indoor navigation systems that depend on proprietary hardware or sensor networks. This cost efficiency increases the feasibility of large-scale adoption.

4.3 Global Impact

The problem addressed by VAVI is universal and not limited to a specific country or region. Visually impaired individuals around the world face similar challenges when navigating indoor environments.

Since the system is software-based, language-independent, and adaptable to different indoor layouts, it can be integrated into various environments globally. With appropriate localization (language and building data), VAVI can be used in different countries, cultures, and architectural contexts, making it a globally applicable assistive technology solution.

4.4 Environmental Impact

The environmental impact of the VAVI project is minimal. The system does not require additional physical infrastructure, sensors, or hardware installations beyond existing mobile devices. This reduces electronic waste and material consumption.

Additionally, most of the processing is handled efficiently through a hybrid architecture, minimizing unnecessary energy consumption. As a result, the project does not introduce any significant negative environmental effects and can be considered environmentally neutral.

---

5. Contemporary Issues

5.1 AI Ethics in Assistive Technologies

The increasing use of artificial intelligence in assistive technologies raises important ethical considerations. In the VAVI project, AI models are responsible for critical tasks such as location inference and real-time object detection, which directly affect user safety and navigation decisions.

One of the primary ethical concerns is the **reliability of AI-driven decision-making** in environments where incorrect guidance may lead to confusion or unsafe situations. To address this issue, the system is designed to provide supportive guidance rather than fully autonomous decision-making. The user remains in control, and AI-generated outputs are presented as assistive recommendations through audio and textual feedback.

Additionally, the project prioritizes transparency in system behavior. Users are informed when AI-based analysis is being used (e.g., during visual location detection), which aligns with ethical principles of informed usage and responsible AI deployment.

5.2 Privacy and Camera Data Concerns

Privacy is a critical contemporary issue, particularly for systems that rely on camera input and cloud-based AI services. The VAVI system temporarily captures camera frames to analyze indoor visual cues such as signs or labels for location detection purposes.

To mitigate privacy risks, the system follows a **minimal data usage principle**. Only a limited number of frames are captured, and these frames are used solely for location inference. No personal data is intentionally collected, stored, or shared beyond what is strictly necessary for system functionality.

Furthermore, the system avoids persistent storage of camera images. Visual data is processed either in real time or transmitted securely to external AI services (such as the Gemini API) and discarded immediately after analysis. This approach reduces the risk of unauthorized data access and complies with general data protection principles.

By incorporating privacy-aware design decisions, the VAVI project demonstrates that AI-powered navigation systems can be developed responsibly while respecting user privacy and ethical constraints.

---

6. New Tools and Technologies Used

During the development of the VAVI project, several modern frameworks, APIs, AI models, and hardware components were utilized. Some of these tools were used for the first time by the project team and played a crucial role in achieving the project objectives.

6.1 Framework: Flutter

Flutter was used as the primary framework for developing the mobile application. It is an open-source UI framework that allows cross-platform mobile application development using a single codebase.

Purpose:
Flutter was used to design and implement an accessible mobile interface that supports both visual and audio interaction, including dropdown-based selection and voice-assisted commands.

Reason for Selection:
Flutter was chosen due to its high performance, cross-platform support, rich UI capabilities, and strong community support. Its ability to rapidly prototype and develop user interfaces made it particularly suitable for an accessibility-focused application.

6.2 APIs: Gemini API

The Gemini API was integrated into the system for advanced visual understanding and scene interpretation.

Purpose:
The Gemini API analyzes multiple camera frames captured by the mobile device to identify visual cues such as office signs, labels, or textual indicators. Based on this analysis, it helps infer the user's indoor location.

Reason for Selection:
Gemini API was preferred due to its strong multimodal capabilities, particularly its ability to interpret visual data and extract semantic meaning from images. This made it a suitable alternative after traditional machine learning-based location prediction models failed to achieve the desired accuracy.

6.3 AI Models: YOLO and Classical Machine Learning Models

YOLO (You Only Look Once) was used for real-time object detection within the mobile application.

Purpose:
YOLO enables the system to detect surrounding objects and potential obstacles in real time, enhancing user safety during navigation.

Reason for Selection:
YOLO was selected because of its high detection speed and suitability for real-time applications, which is essential for continuous camera-based object detection.

In addition to YOLO, several classical machine learning models were experimentally evaluated for location tracking purposes, including:

- Random Forest
- K-Nearest Neighbors (KNN)
- Support Vector Machines (SVM)
- XGBoost
- Logistic Regression

These models were ultimately discarded due to insufficient accuracy and reliability in complex indoor environments.

6.4 Backend Technology: FastAPI

FastAPI was used to implement the backend services of the system.

Purpose:
The backend handles communication between the mobile application and the database, manages navigation requests, and supports AI-related operations.

Reason for Selection:
FastAPI provides high performance, simplicity, and automatic API documentation, making it well-suited for rapid backend development and integration with AI-based systems.

6.5 Database: Microsoft SQL Server

Microsoft SQL Server was used as the main database system.

Purpose:
The database stores indoor navigation data, including nodes, edges, and graph structures required for shortest path calculation.

Reason for Selection:
It was chosen due to its reliability, structured data support, and compatibility with backend technologies used in the project.

6.6 Hardware Components

The project primarily relies on **standard smartphone hardware**, including:

- Camera
- Microphone
- Speaker or headphones

Purpose:
The camera is used for location detection and object recognition, the microphone enables voice-based interaction, and audio output devices provide navigation instructions and alerts.

Reason for Selection:
Using existing smartphone hardware eliminates the need for additional devices or sensors, reducing cost and increasing accessibility. This design choice aligns with the goal of creating a practical and scalable assistive technology.

---

7. Test Results and Assessment

This section presents the test results of the VAVI system based on the previously prepared Test Plan document

TestPlanReport

. The testing activities were conducted to evaluate the functional correctness, performance, usability, accuracy, and overall reliability of the system in real-world indoor environments.

7.1 Test Case Summary

Testing activities were carried out across multiple test categories as defined in the test plan:

- **Functional Tests**
- **Performance Tests**
- **Usability Tests**
- **Accuracy and Reliability Tests**
- **Integration and System-Level Tests**

A total of **approximately 30 test cases** were executed throughout the testing phases, including unit, integration, system, performance, user acceptance, and beta testing.

7.2 Functional Test Results

Functional tests focused on verifying the correct operation of core system features such as navigation, object detection, audio feedback, and backend communication.

Key Results:

- Most functional test cases **passed successfully**.
- Source and target node selection (manual and voice-based) worked as expected.
- Shortest path computation and navigation guidance were generated correctly.
- YOLO-based object detection triggered appropriate audio warnings.

Status:
✓ Passed: ~90%
✕ Failed: ~10%

Minor failures were mostly related to edge cases such as unstable Wi-Fi signals or delayed sensor updates.

7.3 Performance Test Results

Performance testing was conducted according to the target metrics defined in the test plan

TestPlanReport

.

Component	**Target**	**Observed Result**
YOLO inference latency	< 120 ms	~100–130 ms
Camera preview	≥ 15 FPS	~15–20 FPS
Wi-Fi scan duration	< 300 ms	~250–350 ms
Fusion update rate	≥ 3 Hz	~3–4 Hz
Navigation accuracy	±2 m	~1.5–2.5 m
Server response time	< 200 ms	~150–220 ms

Status:
✓ Passed with minor deviations

Some performance metrics slightly exceeded target thresholds under adverse conditions (crowded corridors, weak Wi-Fi), but the system remained usable and stable.

7.4 Usability and User Acceptance Test Results

User Acceptance Testing (UAT) was conducted with non-visually impaired users during the initial phase, followed by beta testing in a real indoor environment.

Key Observations:

- Users were able to reach their selected destination with acceptable accuracy.
- Audio-based navigation instructions were clear and timely.
- Directional audio feedback for obstacle detection was intuitive and effective.

Status:
✓ Passed

The system was generally perceived as easy to use, safe, and suitable for visually impaired users.

7.5 Integration and System Test Results

Integration tests verified the interaction between major system components, including:

- Flutter mobile app ↔ FastAPI backend
- Sensor modules ↔ fusion pipeline
- Object detection ↔ audio feedback module

Status:
✓ Passed

No critical integration failures were observed. Minor synchronization delays were resolved during regression testing.

7.6 Failed Tests, Bugs, and Limitations

A small number of test cases failed or produced inconsistent results:

- Temporary navigation inaccuracies due to **Wi-Fi signal instability**
- **IMU drift** during prolonged movement
- Slight delays in YOLO inference under heavy load

These issues did not cause system crashes but affected accuracy and response time in specific scenarios.

7.7 Test Assessment and Future Improvements

Overall, the testing results indicate that the VAVI system meets its primary functional and performance objectives. The majority of test cases passed successfully, and no critical safety-related defects were identified.

Potential future improvements include:

- Further optimization of YOLO inference performance
- Improved sensor fusion techniques to reduce IMU drift
- Enhanced robustness against unstable Wi-Fi conditions
- Expanded testing with visually impaired users
- Long-duration stress testing and multi-building deployment

---

8. Project Repository and Public Availability

All project artifacts have been made publicly available in accordance with the course requirements. The source code, documentation, and demonstration materials are accessible through the following platforms:

- **Project Website:**
  https://berkaykkaraca.github.io/VAVI/

The project website provides an overview of the VAVI system, including project objectives, system architecture, features, and demonstration information. It serves as a public showcase of the project and allows external users to understand the system design and functionality.

- **GitHub Repository (Mobile Application):**
  https://github.com/ceydakuscuoglu/VaviApp

The GitHub repository contains the source code of the Flutter-based mobile application, including user interface components, camera integration, audio feedback mechanisms, and AI model integration. Version control and issue tracking were managed through GitHub throughout the development process.

These resources ensure transparency, reproducibility, and accessibility of the project, and they comply with the requirement to make the project publicly available before submission.

---

9. Conclusion

In this project, VAVI, an AI-assisted indoor navigation and object detection system for visually impaired individuals, was successfully designed, implemented, and evaluated. The system integrates mobile technologies, computer vision, artificial intelligence, and audio-based interaction to address a real-world accessibility problem.

A hybrid system architecture was adopted to balance real-time performance and computational efficiency. Vision-based location detection using the Gemini API, real-time object detection with YOLO, and shortest path navigation collectively enabled accurate and user-friendly indoor guidance. The system was developed with a strong focus on accessibility, cost efficiency, and scalability.

Comprehensive testing activities demonstrated that the system meets its primary functional and performance objectives under real-world conditions. While minor limitations related to sensor reliability and environmental variability were observed, these issues do not prevent effective system usage and can be addressed through future enhancements.

Overall, the VAVI project demonstrates how modern AI-driven engineering solutions can be applied responsibly to improve the independence and quality of life of visually impaired individuals. The project provides a strong foundation for further research and development in assistive indoor navigation systems.

---

10. References

1. IEEE Computer Society, *IEEE Standard for Software and System Test Documentation*, IEEE Std 829-2008, 2008.
2. ISO/IEC/IEEE, *ISO/IEC/IEEE 29119-1:2013 — Software and Systems Engineering — Software Testing*, 2013.
3. G. J. Myers, C. Sandler, and T. Badgett, *The Art of Software Testing*, 3rd ed., Wiley, 2011.
4. B. Beizer, *Software Testing Techniques*, 2nd ed., Van Nostrand Reinhold, 1990.
5. C. Kaner, J. Bach, and B. Pettichord, *Lessons Learned in Software Testing*, Wiley, 2002.
6. Google, *Flutter Documentation*, 2024. https://docs.flutter.dev/
7. Sebastián Ramírez, *FastAPI Official Documentation*, 2024. https://fastapi.tiangolo.com/
8. Ultralytics, *YOLO Object Detection Documentation*, 2024. https://docs.ultralytics.com/
9. Google, *Android Sensor API Documentation*, 2024. https://developer.android.com/guide/topics/sensors
10. R. E. Kalman, *A New Approach to Linear Filtering and Prediction Problems*, Journal of Basic Engineering, 1960.
11. Google AI, *Gemini API Documentation*, 2024.
12. Microsoft, *SQL Server Documentation*, 2024. https://learn.microsoft.com/sql/