Software Requirements Specifications

ENVIRONMENT DETECTION FOR BLIND USER

Project Code:

EDBP-2025

Internal Advisor:

Prof. Muhammad Fahad

Project Manager:

Dr. Muhammad Ilyas

Project Team:

Ali Raza Ansari (BSCS51F22S008) — Team Leader

Abdullah Zammad (BSCS51F22S036) — Team Member

M. Hassan (BSCS51F22S040) — Team Member

Submission Date:

October 20, 2025

Project Manager's Signature

Document Information

Category	Information
Customer	CS
Project	Environment detection for blind user
Document	Requirement Specifications
Document Version	1.0
Identifier	BSCS-51-F-22-S
Status	Draft
Author(s)	Ali Raza Ansari (BSCS51F22S008) Abdullah zammad (BSCS51F22S036) M. Hassan (BSCS51F22S040)
Approver(s)	PM
Issue Date	Oct. 15, 2025
Document Location	
Distribution	1. Advisor 2. PM 3. Project Office

Definition of Terms, Acronyms and Abbreviations.

Term	Description
TF	TensorFlow
CV	OpenCV
RTOD	Real Time Object Detection
TTS	Text-To-Speech

Table of Content

1. Introduction

1.1 Purpose of Document

This document outlines the objectives, scope, and features of the “Smart Assistant for Visually Impaired” project, which uses AI and audio feedback to help visually impaired individuals identify objects. It is intended for supervisors, evaluators, developers, and stakeholders involved in the project’s design and development.

1.2 Project Overview

“Smart Assistant for Visually Impaired” is an AI-powered system that detects objects through a camera and provides real-time audio feedback, helping visually impaired users navigate safely and confidently. It aims to enhance independence, mobility, and awareness in everyday life.

1.3 Scope

Included:

- Real-time object detection using smartphone camera
- Image preprocessing (resizing, normalization, noise reduction)
- Audio feedback generation using offline Text-to-Speech (TTS)
- Direction identification (Left, Center, Right)
- Continuous real-time operation
- Simple and accessible mobile interface

Not Included:

- Distance estimation between user and objects
- Object size measurement
- Navigation or path planning features
- IoT or external hardware integration
- Facial recognition or text reading (OCR)

2. Overall System Description

The Smart Assistant for Visually Impaired is an AI-based system designed to help visually impaired individuals recognize objects in their surroundings through real-time audio feedback. The system uses a camera to capture visuals, processes them with AI algorithms, and provides speech output to guide the user safely and independently.

2.1 User characteristics

The primary users are visually impaired individuals who rely on audio feedback for object recognition and navigation.
Secondary users include developers, testers, and system administrators responsible for installation, updates, and maintenance.
The most critical users are visually impaired individuals, as the system is built specifically to enhance their independence and mobility.

2.2 Operating environment

Frontend: Flutter
Backend: Python (Django + REST API)
AI Model: Computer Vision
Hosting: Cloud-based or local server deployment

2.3 System constraints

- **Software Constraints:** Real-time object detection may be affected by processing speed and system performance.
- **Hardware Constraints:** Requires a functional camera, earphones, and sufficient memory for AI processing.
- **Cultural Constraints:** Currently supports English; multilingual support may be added later.
- **Legal Constraints:** Must ensure user data privacy and comply with accessibility standards.
- **Environmental Constraints:** Performance may decrease in low-light or noisy surroundings.
- **User Constraints:** Interface must be fully audio-based for accessibility.
- **Off-the-Shelf Components:** Depends on open-source AI models and libraries with specific licensing terms.

3. External Interface Requirements

The Smart Assistant for Visually Impaired system interacts with several external components, including hardware devices (camera, earphones), software frameworks (AI models, libraries), and communication networks (Wi-Fi or mobile data). These interfaces ensure smooth integration and data flow between all components.

3.1 Hardware Interfaces

The system interfaces with cameras, and earphones.

- The camera captures images of the surroundings for object detection.
- The earphones enable audio interaction between the user and the system.
- Data is transferred digitally between the hardware and software through device drivers and APIs.

3.2 Software Interfaces

The system connects multiple software components:

- **Frontend:** Developed using Flutter, providing an accessible and interactive interface.
- **Backend:** Built with Python (Django + REST API) for managing data and communication between modules.
- **AI Model:** Uses Computer Vision algorithms via TensorFlow and OpenCV for object detection.
- **Operating Systems:** Compatible with Android platforms.

3.3 Communications Interfaces

The system communicates over Wi-Fi or mobile data for updates, cloud access, and model synchronization.
It uses HTTP/HTTPS protocols for secure API communication between the Flutter frontend and Django REST backend.
Data exchanged is formatted in JSON and encrypted for privacy protection.

4. Functional Requirements

- **Object Detection:** The system must capture images using the camera and identify surrounding objects using AI-based computer vision models.
- **Audio Feedback:** The system must convert detected objects into speech output through a Text-to-Speech (TTS) module.
- **Real-Time Processing:** The system must process data and deliver feedback instantly for effective navigation.
- **User Interface:** The mobile app (Flutter) must offer an easy-to-use interface with minimal interaction, suitable for visually impaired users.
- **Data Management:** The backend (Django + REST API) should manage communication, data flow, and logs efficiently.
- **Language Support:** The system should provide audio feedback in English, with options to add more languages in future versions.
- **System Updates:** The system should allow periodic updates to AI models and application features when connected online.

5. Non-functional Requirements

5.1 Performance Requirements

The system shall provide **real**-time object detection and audio feedback with a maximum response delay of one second. It must maintain high accuracy and precision in recognizing objects under standard environmental conditions. The application shall demonstrate reliability and stability during continuous operation and be optimized for low power consumption and efficient memory usage on Android-based devices and portable hardware.

5.2 Safety Requirements

The system shall prioritize user safety by ensuring accurate and timely feedback to prevent potential hazards during navigation. In the event of low light, camera obstruction, or system malfunction, the system shall notify the user through audio alerts. All hardware components (camera, earphones, and power modules) must comply with

electronic safety and ergonomic standards. The system shall avoid generating misleading or delayed information that could compromise user well-being.

5.3 Security Requirements

The system shall ensure data security, integrity, and user privacy through encrypted communication protocols (HTTPS) and secure data handling practices. Access to system settings and updates shall be restricted to authorized users only. No personal data shall be collected or transmitted without explicit user consent. The system shall comply with data protection and accessibility standards, ensuring confidentiality and safeguarding all stored and transmitted information.

6. Assumptions and Dependencies

6.1 Assumptions

- Users will have access to compatible Android devices equipped with a camera and internet connectivity (for updates).
- The AI models and libraries (e.g., TensorFlow, OpenCV) will remain available, stable, and compatible with future software updates.
- The hardware components (camera, earphones) will function correctly and meet minimum performance standards.
- Users will operate the system in normal environmental conditions with adequate lighting for object detection.
- The availability of cloud services or local storage for model updates and data logs will remain uninterrupted.

6.2 Dependencies

- The project depends on third-party frameworks and APIs such as TensorFlow, OpenCV, and Django REST API for object detection and communication.
- The Text-to-Speech (TTS) engine and Computer Vision models are external components whose updates or licensing changes may affect system performance.
- The system’s success depends on timely hardware integration (e.g smartphones).

- The project relies on stable network connections for online features, updates, and data synchronization.
- The development timeline depends on the availability of trained AI models and dataset quality for accurate object recognition.

7. References

[1] “Guide Dogs vs. White Canes: The Comprehensive Comparison,” CloverNook, Sept. 18, 2020. Available: https://clovernook.org/2020/09/18/guide-dogs-vs-white-canes-the-comprehensive-comparison/

[2] A. Tesfaye, “Enhancing Mobility and Safety: A Smart Walking Cane for Visually Impaired Individuals with Ultrasonic Sensor, Infrared, and GSM Module,” Journal of Computational Science and Data Analytics, 2024. Available: https://journal.aastu.edu.et

[3] “Design of Smart Cane with Integrated Camera Module for Visually Impaired People,” International Journal for Research in Applied Science and Engineering Technology (IJRASET), 2023. Available: https://www.ijraset.com

[4] “Hybrid Approach Using Ultrasonic and Vision-Based Detection for Obstacle Recognition,” International Journal of Engineering and Advanced Technology (IJEAT), vol. 8, no. 4, 2019. Available: https://www.ijeat.org

[5] “Efficacy and Patients’ Satisfaction with the ORCAM MyEye Device Among Visually Impaired People: A Multicenter Study,” *PubMed*, 2024 . Available: https://pubmed.ncbi.nlm.nih.gov/36645535/