ANKARA SCIENCE UNIVERSITY FACULTY OF ENGINEERING
SOFTWARE ENGINEERING & ELECTRIC-ELECTRONICAL ENGINEERING DEPARTMENT
GRADUATION PROJECT - 1

GRADUATION PROJECT REPORT

ALAS - AI GLASSES

Prepared by 4 students

Mehmet ALGÜL, 210204004

Enes ÜNLÜ, 210202001

İlayda YAKAR, 210202008

Yasin YILDIRIM, 210204005

December 2025

ANKARA

The project titled "**ALAS**" prepared by Mehmet ALGUL, Yasin YILDIRIM, Enes UNLU and Ilayda Yakar. has been accepted as a GRADUATION PROJECT in Ankara Science University, Software Engineering Dept. by the following jury with VOTE UNANIMOUS / VOTE MAJORITY.

Advisor: Assoc. Prof. M. Nedim ALPDEMİR

Software Engineering Department

I confirm that this report is a Graduation Project in scope and quality.

Signature: …………………….

Member: ………………………………….

………………………………… Engineering Department

I confirm that this report is a Graduation Project in scope and quality.

Signature: …………………….

Member: ………………………………….

………………………………… Engineering Department

I confirm that this report is a Graduation Project in scope and quality.

Signature: …………………….

I confirm that this project, accepted by the jury, fulfills the requirements for being a Graduation Project. ..../……./……

…………………

Prof./Assoc.Dr….

Head of <DEPARTMENT Name>

CONFLICT OF INTEREST

In this project study that we prepared in accordance with Ankara Science University Engineering Faculty Project Writing Rules.

We declare that;

- We have obtained the data, information, and documents are presented in the project report within the framework of academic and ethical rules,
- We have presented all information, documents, evaluations, and results in accordance with the rules of scientific ethics and morality,
- We have cited all the works I have benefited from in the project study with appropriate attribution,
- We have not made any changes to the data used,
- This report is original and has not been presented elsewhere before,
- We have prepared within the framework of workplace training within the scope of the studies and observations,

We, all group members, hereby declare that we accept all loss of rights that may arise against us in any other case declared above. ..../……./……

Yasin YILDIRIM Signature………..…

İlayda YAKAR Signature………..…

Mehmet Algül Signature………..…

Enes ÜNLÜ Signature………..…

ALAS AI GLASS

(Graduation Project)

ABSTRACT

In this project, an AI-based smart glasses system called ALAS has been designed to support safer and more independent outdoor mobility for visually impaired individuals. The system has been structured as a wearable assistive platform that combines environmental perception, obstacle awareness, and offline route guidance through an embedded processing unit integrated with a camera module and motion/location sensors. Within the scope of the study, a literature review has been conducted, system requirements have been specified, and a modular architecture has been defined. A deep-learning-based perception approach suitable for execution on embedded hardware has been adopted to enable local scene understanding and hazard awareness. To reduce dependency on external services and connectivity, offline navigation has been planned using OpenStreetMap-based route generation. User interaction has been addressed through speech output (Text-to-Speech) and a button-activated speech input mechanism (Speech-to-Text) to minimize unintended activations and support intentional commands. The integration strategy for hardware and software components has been outlined, and scenario-based verification and performance evaluation steps have been defined to validate the proposed design. The resulting outcomes have been assessed within a prototype-oriented framework aligned with offline operation, privacy-preserving on-device processing, and real-time assistive navigation goals.

Keywords: smart glasses, assistive technology for the visually impaired, embedded AI, offline navigation, computer vision, speech feedback

ALAS YZ GÖZLÜK
(Mezuniyet Projesi)
ÖZET

Bu projesinde, görme engelli bireylerin dış ortamda daha güvenli ve bağımsız hareket edebilmesini desteklemek amacıyla ALAS adlı yapay zekâ tabanlı akıllı gözlük sistemi tasarlanmıştır. Sistem; giyilebilir bir kamera modülü, konum ve hareket sensörleri ile gömülü bir işlem birimi kullanılarak çevresel algı, engel farkındalığı ve çevrimdışı rota yönlendirmesini tek bir yapıda birleştirecek şekilde kurgulanmıştır. Çalışma kapsamında literatür incelemesi yapılmış, gereksinimler tanımlanmış ve modüler bir sistem mimarisi oluşturulmuştur. Çevresel algı için gömülü donanım üzerinde çalıştırılabilecek derin öğrenme tabanlı algılama yaklaşımı benimsenmiş; navigasyon için dış servislere bağımlılığı azaltmak üzere OpenStreetMap tabanlı çevrimdışı rota üretimi planlanmıştır. Kullanıcı etkileşimi, sesli geri bildirim (TTS) ve istenmeyen aktivasyonları azaltmak amacıyla butonla etkinleşen sesli komut girişi (STT) ile ele alınmıştır. Donanım ve yazılım bileşenlerinin entegrasyon yaklaşımı tanımlanmış; sistemin doğrulanması için senaryo tabanlı test ve performans değerlendirme adımları belirlenmiştir. Elde edilen çıktılar, çevrimdışı çalışma, gizlilik odaklı yerel işlem ve gerçek zamanlı yardımcı navigasyon hedefleri doğrultusunda bir prototip çerçevesinde değerlendirilmiştir

Anahtar Kelimeler : akıllı gözlük, görme engelliler için yardımcı teknoloji, gömülü yapay zekâ, çevrimdışı navigasyon, bilgisayarlı görü, sesli geri bildirim

TABLE OF CONTENTS

Table of Contents

TABLE OF CHARTS

TABLE OF FIGURES

SYMBOLS AND ABBREVIATIONS

The symbols and abbreviations used in this study are presented below with their explanations.

Symbols	**Explanations**
m^3	Comments should **not** be longer than one line
db	Decibel
hz	Hertz
m^2	Square meter

Abbreviations	**Explanations**
AB	Comments should not be longer than one line
ASHRAE	Abbreviations should be given in alphabetical order
ASTM	Abbreviations should be given in alphabetical order
BRE	Abbreviations should be given in alphabetical order
BREEAM	Abbreviations should be given in alphabetical order
BTK	Abbreviations should be given in alphabetical order
CFD	Abbreviations should be given in alphabetical order

MAPPING TO MÜDEK PROGRAM OUTCOMES

Section	Program Outputs	Learning Outputs
Section 1.4	P.O. (2) (a,b,c)	L.O.1
Section 2	P.O. (5) (a)	L.O.2
Section 1.4	P.O. (6) (a)	L.O.3
Section 8	P.Ç. (8) (a)	L.O.4
Section 8 & Presentation	P.Ç. (9) (a,b)	L.O.5
Section 7	P.Ç. (11) (a,b,c)	L.O.6

ALAS

(Graduation Project)

ABSTRACT

In this project, an AI-based smart glasses system called ALAS has been designed to support safer and more independent outdoor mobility for visually impaired individuals. The system has been structured as a wearable assistive platform that combines environmental perception, obstacle awareness, and offline route guidance through an embedded processing unit integrated with a camera module and motion/location sensors. Within the scope of the study, a literature review has been conducted, system requirements have been specified, and a modular architecture has been defined. A deep-learning-based perception approach suitable for execution on embedded hardware has been adopted to enable local scene understanding and hazard awareness. To reduce dependency on external services and connectivity, offline navigation has been planned using OpenStreetMap-based route generation. User interaction has been addressed through speech output (Text-to-Speech) and a button-activated speech input mechanism (Speech-to-Text) to minimize unintended activations and support intentional commands. The integration strategy for hardware and software components has been outlined, and scenario-based verification and performance evaluation steps have been defined to validate the proposed design. The resulting outcomes have been assessed within a prototype-oriented framework aligned with offline operation, privacy-preserving on-device processing, and real-time assistive navigation goals.

Keywords : smart glasses, assistive technology for the visually impaired, embedded AI, offline navigation, computer vision, speech feedback

ALAS YZ GÖZLÜK

(Mezuniyet Projesi)

ÖZET

Bu projesinde, görme engelli bireylerin dış ortamda daha güvenli ve bağımsız hareket edebilmesini desteklemek amacıyla ALAS adlı yapay zekâ tabanlı akıllı gözlük sistemi tasarlanmıştır. Sistem; giyilebilir bir kamera modülü, konum ve hareket sensörleri ile gömülü bir işlem birimi kullanılarak çevresel algı, engel farkındalığı ve çevrimdışı rota yönlendirmesini tek bir yapıda birleştirecek şekilde kurgulanmıştır. Çalışma kapsamında literatür incelemesi yapılmış, gereksinimler tanımlanmış ve modüler bir sistem mimarisi oluşturulmuştur. Çevresel algı için gömülü donanım üzerinde çalıştırılabilecek derin öğrenme tabanlı algılama yaklaşımı benimsenmiş; navigasyon için dış servislere bağımlılığı azaltmak üzere OpenStreetMap tabanlı çevrimdışı rota üretimi planlanmıştır. Kullanıcı etkileşimi, sesli geri bildirim (TTS) ve istenmeyen aktivasyonları azaltmak amacıyla butonla etkinleşen sesli komut girişi (STT) ile ele alınmıştır. Donanım ve yazılım bileşenlerinin entegrasyon yaklaşımı tanımlanmış; sistemin doğrulanması için senaryo tabanlı test ve performans değerlendirme adımları belirlenmiştir. Elde edilen çıktılar, çevrimdışı çalışma, gizlilik odaklı yerel işlem ve gerçek zamanlı yardımcı navigasyon hedefleri doğrultusunda bir prototip çerçevesinde değerlendirilmiştir

Anahtar Kelimeler : akıllı gözlük, görme engelliler için yardımcı teknoloji, gömülü yapay zekâ, çevrimdışı navigasyon, bilgisayarlı görü, sesli geri bildirim

1. Introduction

Visual impairment significantly limits an individual's mobility, independence, and social participation. Navigating outdoor environments poses severe challenges due to dynamic obstacles, complex traffic patterns, and the lack of accessible infrastructure. While traditional aids such as white canes provide immediate tactile feedback for ground-level obstacles, they cannot reliably detect head-level hazards or convey semantic information about the surroundings. Moreover, many existing electronic travel aids depend on continuous internet connectivity or high-cost proprietary hardware, making them inaccessible or unreliable in regions with poor network coverage.

This project introduces ALAS (AI Glasses), a wearable assistive technology designed to address these limitations. ALAS is built around a Jetson Nano based embedded system that integrates a monocular RGB camera embedded into the glasses and multiple sensors to provide real-time environmental perception. By leveraging edge AI and deep learning techniques, the system performs semantic segmentation and object-level awareness locally, without relying on cloud services. The overall aim is to bridge the gap between simple obstacle detection and comprehensive environmental understanding, enabling visually impaired individuals to navigate with greater confidence and safety.

1.1 Purpose

The primary purpose of the ALAS project is to design and prototype a wearable, AI-powered smart-glass system that enhances the independent mobility and quality of life of visually impaired individuals. The project seeks to go beyond traditional aids, which are effective for detecting ground-level obstacles but insufficient for recognizing head-level hazards, identifying specific objects, or navigating complex routes. ALAS aims to offer a comprehensive “second pair of eyes” through advanced computer vision and multi-sensor fusion technologies.

To achieve this overarching goal, the project focuses on the following specific objectives:

> **Developing an Offline-First Architecture:** Unlike many commercial assistive devices that rely on cloud computing, a core objective of ALAS is to perform all critical processing (segmentation, object-level interpretation, and navigation calculations) on the edge (Jetson Nano). This ensures that the user is not dependent on internet connectivity, which is vital for safety in remote, underground, or low-coverage areas.
>
> **Providing Semantic Environmental Understanding:** The system aims to move beyond simple obstacle avoidance by offering semantic context. Using deep learning models (specifically U-Net or similar architectures for semantic

segmentation), the device will classify and convey the type of obstacles (for example, distinguishing a "car" from a "bench" or a "pedestrian") to the user via Text-to-Speech (TTS), improving their mental mapping of the surroundings.

Ensuring Cost-Effectiveness and Accessibility: High costs of existing electronic travel aids (ETAs) often create a barrier to entry. The project aims to utilize cost-effective, off-the-shelf hardware components and open-source software libraries to create a solution that is affordable without severely compromising performance, making advanced assistive technology accessible to a broader demographic.

Enhancing Safety with Multi-Sensor Fusion: To maximize reliability, the project will integrate data from an RGB-D camera (for depth and visual information) with IMU and GPS sensors. This multi-sensor approach is intended to provide accurate distance estimation and robust localization, reducing the risk of collisions and falls.

Contributing to the Literature: By implementing and optimizing deep neural networks on a resource-constrained embedded platform, the project also aims to contribute to the academic field of embedded AI, providing feasibility insights and performance benchmarks that can guide future research on assistive technologies.

1.2 Project Scope

The scope of this graduation project is to design, implement, and evaluate a wearable AI-based smart glasses system that supports visually impaired individuals during outdoor pedestrian navigation in urban environments. The system focuses on real-time environmental perception, obstacle awareness, and offline route guidance using an embedded computing platform based on Jetson Nano, without relying on cloud services or continuous internet connectivity.

Within this scope, the project will:

- Develop a lightweight semantic segmentation model (U-Net–based or similar CNN architecture like YOLO) that runs locally on Jetson Nano and can distinguish key scene elements such as road, sidewalk, obstacles, and other critical classes from RGB-D camera input.
- Integrate depth, IMU, and GPS data to support real-time obstacle detection and basic situational awareness (direction of movement, approximate distance to obstacles, and user position on the route).
- Implement an offline navigation module that uses pre-processed OpenStreetMap (OSM) data and A* algorithm to generate pedestrian routes and update guidance as the user moves.
- Design a user interaction layer based on Text-to-Speech (TTS) for guidance and warnings, and a button-triggered Speech-to-Text (STT) interface for simple commands (for example "nearest pharmacy", "cancel route", "status"), operating fully on-device to preserve privacy.

- Build a working prototype that integrates the RGB-D camera, Jetson Nano, IMU, GPS, microphone, speaker, power unit, and basic housing into a portable form factor suitable for field tests, and evaluate it using metrics such as frame rate, model accuracy, TTS latency, navigation success rate, and battery life.

The main deliverables of the project are:

- A system-level requirements and design specification for the ALAS smart glasses;
- A trained and optimized segmentation model exported in an embedded-ready format (e.g., TFLite) and integrated into the perception pipeline;
- Navigation, perception, and TTS/STT software modules running together on Jetson Nano OS;
- A hardware prototype of the smart glasses system with at least several hours of autonomous operation;
- Experimental results and analysis based on laboratory and limited outdoor tests, documented in the graduation project report.

The project is intentionally limited in the following ways:

- The system is designed for outdoor pedestrian scenarios such as sidewalks, crosswalks, and simple open areas; complex indoor navigation (malls, metro stations, multi-floor buildings) is out of scope.
- The prototype targets a single user and a single device; multi-user coordination, remote monitoring, or cloud-connected services are not considered.
- The system is a digital assistive technology and does not aim to replace medical devices, clinical diagnosis, or certified rehabilitation tools.
- Only a limited set of voice commands and one primary output language will be supported in the first prototype; advanced multilingual, personalization, or large-scale usability studies are outside the current scope.
- The project will use publicly available datasets and a limited amount of locally collected data; large-scale data collection campaigns and productization activities (industrial design, mass production) are not targeted within this graduation project period.

1.3 Innovative Aspect of the Project

The proposed ALAS system combines several features that are rarely integrated within a single assistive device for visually impaired users. First, unlike many commercial electronic travel aids that depend on smartphones or cloud-based processing, ALAS adopts a fully offline, edge-AI architecture. All critical tasks—semantic segmentation, obstacle interpretation, and route planning—are executed locally on a Jetson Nano, reducing latency and eliminating dependency on continuous internet access.

Second, the project leverages RGB semantic segmentation together with IMU- and GPS-based localization to provide context-aware navigation. Instead of only detecting the presence of an obstacle, the system distinguishes between different object types (such as vehicles, pedestrians, and street furniture) and relates them to the user's current route. This integrated perception-and-navigation pipeline is designed specifically for outdoor pedestrian use and is tailored to the computational constraints of embedded hardware.

Third, ALAS introduces a privacy-preserving interaction model based on fully local Text-to-Speech (TTS) and button-triggered Speech-to-Text (STT). Voice commands are processed on-device, preventing sensitive data from leaving the system while maintaining a natural, hands-free interface. Finally, the modular, low-cost hardware design and open-source–oriented software stack aim to facilitate reproducibility and further research on embedded AI for assistive technologies.
Finally, beyond its technical novelty, the project addresses a critical socio-economic barrier. By utilizing off-the-shelf, low-cost hardware components and open-source software, ALAS aims to offer a high-performance alternative to prohibitively expensive commercial ETAs. This cost-effective design not only facilitates reproducibility for further academic research but also aims to democratize access to advanced assistive technology, fostering greater independence and social inclusion for visually impaired individuals.

1.4 Global Goals for Sustainable Development

This project is aligned with several United Nations Sustainable Development Goals (SDGs):

- **SDG 3 – Good Health and Well-Being:**
  ALAS contributes to safer mobility and reduced risk of accidents for visually impaired individuals by providing real-time warnings and structured navigation support. Enhancing independent mobility supports mental well-being and overall quality of life.
- **SDG 8 – Decent Work and Economic Growth (Optional):** Independent mobility is a prerequisite for consistent employment. By facilitating safe and reliable daily commuting, ALAS aids in removing physical barriers to the workforce for visually impaired individuals, thereby supporting their economic independence and professional growth.
- **SDG 9 – Industry, Innovation and Infrastructure (secondary contribution):**
  Through the development of an embedded AI platform for assistive navigation, the project demonstrates how locally produced, low-power hardware and software solutions can address societal needs. This supports innovation capacity in digital health and smart assistive technologies at a national level.
- **SDG 10 – Reduced Inequalities:**
  By focusing on cost-effective, off-the-shelf components and offline operation, the system is designed to be accessible to users who may not afford high-end commercial devices or continuous mobile data plans. This helps reduce technological and social barriers faced by people with visual impairments.

- **SDG 11 – Sustainable Cities and Communities:**
  The project promotes inclusive urban environments by enabling visually impaired citizens to participate more actively in daily city life—such as commuting, shopping, or accessing public services—without requiring major changes to existing infrastructure. The use of open geospatial data (e.g., OpenStreetMap) further supports sustainable, community-driven urban information systems.

2. Literature Review

The landscape of assistive technologies for the visually impaired has evolved into two primary categories: mobile application-based software and standalone wearable hardware. While existing solutions provide significant support, there remains a critical gap in providing a cost-effective, fully offline system that integrates high-level environmental perception with robust navigation.

2.1.1 Comparative Analysis of Commercial Solutions

A review of the current market reveals several benchmarks against which the **ALAS** system is positioned:

- **High-End Standalone Wearables:** Devices such as OrCam MyEye 3 Pro and Envision Glasses offer advanced object recognition and real-time navigation. However, these systems are categorized by high costs and often rely on proprietary ecosystems that can limit user flexibility.
- **App-Based Assistants:** Solutions like Seeing AI and Aira utilize powerful cloud-based AI to describe surroundings. While accessible (often free or subscription-based), their dependency on continuous internet connectivity makes them unreliable in low-signal areas or regions with expensive data plans.
- **Navigation-Specific Aids:** Tools such as BlindSquare and Nearby Explorer focus on GPS-based guidance. While effective for broad localization, they often lack the "head-level" obstacle detection and semantic scene understanding provided by the ALAS perception layer.

Commercial Name	Manufacture	Functionality	Mode of Use	Supported OS	Cost
ALAS	US	RT Navigation and Object Detection	Outdoor:through GPS and Camera	Standalone Device	Medium
eSight 4.	eSight Co.	Smart glasses with Augmented Reality	Camera	Standalone device	High to Very High

OrCam MyEye 3 Pro	OrCam Technologies	Reading Objects and people recognitionReal-time	Clips onto the user's eyeglass frame navigation	Standalone device	High
Seeing AI	Microsoft	Object and people recognation , Barcode Scanning	Through AI app	iOS Android	Free
Aira	Aira Tech Corp.	On-line assistance on demand	Through live agent service	iOS Android	Monthly Subscription
BlindSquare	Matapo A.S.	Simplified smartphone	Through physical keyboard and voice commands	Standalone device	Low to Medium
Nearby Explorer	American Printing House Inc.	Navigation	Indoor: through BPS Outdoor:— through GPS	Android iOS	Low
WeWalk Smart Cane	WeWALK LTD	Intelligent voice assistant Navigation instructions	As a smart cane	Standalone device	Low

Table 2.1. Comparative Analysis of Commercial Solutions

2.1.2 Technical Approaches Used in Existing Assistive Technologies

- **High-End Wearables (eSight, OrCam)**

High-end wearable systems such as OrCam MyEye 3 Pro and eSight 4 rely primarily on camera-based computer vision pipelines executed on proprietary embedded hardware. These systems capture RGB images at head level and process them using on-device or semi-embedded neural networks for object recognition, text reading, and scene description

Navigation functionality in such devices is typically limited or secondary, as the primary focus is visual augmentation or object identification rather than graph-based route planning. While these systems achieve high perception accuracy, their closed architectures and specialized hardware significantly increase cost and limit extensibility.

- **App-Based Assistive Systems (Seeing AI, Aira)**

Application-based assistive systems such as Seeing AI and Aira adopt a cloud-centric processing architecture. Visual and audio data captured through the smartphone's camera and microphone are transmitted to remote servers, where large-scale deep learning models perform image understanding, optical character recognition, and natural language processing.

This architecture enables access to computationally intensive models and frequent updates; however, it introduces strong dependency on continuous internet connectivity. As a result, system responsiveness is subject to network latency, and user privacy may be affected due to external data transmission. In the case of Aira, perception and decision-making are partially human-in-the-loop, with remote agents interpreting the scene and providing real-time guidance.

- **Navigation-Focused Assistive Tools (BlindSquare, Nearby Explorer)**

Navigation-focused assistive tools such as BlindSquare and Nearby Explorer primarily rely on GPS-based localization combined with digital map services and point-of-interest databases. Route guidance is generated using positional data and predefined geographic information, typically without incorporating real-time visual perception of the surrounding environment.

While these systems are effective for macro-level navigation and spatial orientation, they lack head-level obstacle detection and semantic scene understanding. Consequently, their guidance is limited in complex urban environments where dynamic obstacles and fine-grained environmental awareness are critical for pedestrian safety.

- **Smart Cane Systems (WeWalk Smart Cane)**

Smart cane solutions such as the WeWalk Smart Cane integrate ultrasonic sensors and inertial measurement units to detect nearby obstacles and estimate relative distances. Proximity-based alerts are generated through audio or haptic feedback, enabling basic obstacle awareness during walking.

Although this sensing approach is energy-efficient and robust, ultrasonic measurements provide limited contextual information. The system can detect the presence of an obstacle but cannot distinguish object type, motion, or semantic relevance, resulting in binary rather than context-aware environmental perception.

2.2. Module-Specific Justification and State-of-the-Art Comparison

The ALAS system is designed to improve upon these existing technologies through a modular, edge-centric approach. Below is the justification for the specific technologies selected for the project.

2.2.1. User Interaction: Button-Activated TTS/STT vs. Continuous Interfaces

Traditional interfaces developed for visually impaired users include solutions based on simple audio alerts (such as buzzer or beep signals) or physical keyboards (e.g., BlindShell).

- **ALAS Approach**: The project utilizes a single-button-activated Speech-to-Text (STT) input and Text-to-Speech (TTS) output.
- **Rationale**: Unlike "always-listening" devices that raise privacy concerns and consume significant battery power, a button-activated interface ensures that the system processes audio only when intended. Compared to haptic-only systems like Wayband, TTS provides richer, semantic information (example: "Turning right toward the pharmacy" vs. a simple vibration), which is essential for complex urban **navigation.**

2.2.2. Navigation: Offline OSM Data and Graph-Based Pathfinding

Most contemporary navigation aids rely on online APIs (Google Maps, Apple Maps) for route generation.

- **ALAS Approach:** The system implements an offline navigation module using OpenStreetMap (OSM) data and graph-based algorithms (such as Dijkstra or A*).
- **Rationale:** While cloud-based APIs offer real-time traffic data, they fail in "dead zones" like subways or rural paths. By pre-loading OSM graphs, ALAS ensures zero-latency route recalculation and maintains user privacy by not transmitting location data to external servers. This mirrors the reliability of high-end standalone devices but at a fraction of the cost by using open-source geospatial data.

2.2.3. Perception: Advanced Deep Learning and RGB Fusion

Traditional electronic travel aids (ETAs) often use ultrasonic sensors to detect proximity, similar to the **WeWalk Smart Cane**.

- **ALAS Approach:** The project leverages Advanced Deep Learning models for semantic segmentation combined with RGB camera data.
- **Rationale:** Ultrasonic sensors can detect that an object is present but cannot distinguish between a "bench" (safe to sit) and a "moving vehicle" (danger). By using depth-aware computer vision, ALAS provides contextual awareness, allowing the system to prioritize warnings based on the *type* and *distance* of the obstacle, a feature typically reserved for much more expensive systems like eSight 4.

2.3. Position Within the Body of Knowledge

In conclusion, **ALAS** establishes a distinctive positioning within the assistive technology landscape by systematically addressing the dual constraints of cloud

dependency found in mobile applications and the prohibitive costs associated with high-end standalone hardware.
By integrating high-fidelity environmental perception with robust, graph-based navigation on a localized Edge AI platform (Nvidia Jetson Nano), the project overcomes the reliability gaps typical of internet-dependent systems. The strategic fusion of RGB data with offline OpenStreetMap (OSM) intelligence ensures that users receive continuous, real-time guidance even in network-deprived urban environments.
Furthermore, the implementation of a locally-processed, button-activated interaction model serves as a vital safeguard for user privacy while optimizing energy consumption for prolonged outdoor use. Ultimately, ALAS demonstrates that through the intelligent integration of off-the-shelf components and optimized open-source algorithms, it is possible to deliver a high-performance, private, and economically accessible solution that fosters independent mobility and social inclusion for the visually impaired.

3. Overall Description

This chapter provides a high-level description of the ALAS system, including its product perspective, object-oriented structure, user characteristics, operating environment, design constraints, and underlying assumptions. The purpose of this section is to position the system within its technical and operational context before detailing functional and non-functional requirements.

3.1 Product Perspective

- This project is developed within the domain of AI-based assistive technologies. Its main objective is to enhance environmental awareness for visually impaired individuals, enabling them to move independently, aware, and safely. The proposed system aims to perform by locally real-time scene analysis using deep learning, sensors, and computer vision, providing adaptive audio guidance to the user.
- The project is positioned within the market of smart glasses and assistive navigation systems. Current commercial and research-orieted solutions mostly rely on ultrasonic sensors for simple obstacle detection. However, these approaches cannot distinguish between object types or interpret complex environmental structures because of that these also cannot provide one of our main goals environmental awareness.
- In contrast, the proposed system introduces a more advanced perception mechanism by integrating deep learning-based semantic segmentation and algorithms. This approach enables the device not only to detect obstacles but also to identify and classify them as pedestrians, vehicles, sidewalks,

tactile pavements or other environmental elements. Furthermore, unlike other systems, the proposed system provides not only detection but also navigate the user where they want to go. Furthermore, traditional systems often provide limited communication with user typically vibration, buttons or simple buzzer sounds. Its cause not to fully understand the message. Our systems create feedback and input mostly using Text-to-Speech technologies.

- From a hardware perspective, previous solutions are mostly had dependence on smartphones, online service or connection with internet which restrict computational capability and accessibility. The proposed design operates on a Jetson Nano based embedded AI platform, allowing on-device inference with real-time performance while maintaining portability. Additionally, its low-cost, modular and local structure makes it more accessible and suitable for usage and further technological development

The ALAS software architecture follows an object-oriented and modular design to ensure clarity, maintainability, and scalability. The system is structured around a set of core components that communicate through well-defined interfaces.

At a high level, the system consists of the following main objects and modules:

- **CameraManager**: Responsible for acquiring synchronized RGB and depth frames from the RGB-D camera and forwarding them to the preprocessing pipeline.
- **SensorFusionManager**: Collects and synchronizes IMU and GPS data, providing orientation, motion, and position information to the navigation and perception modules.
- **PreprocessingPipeline**: Performs image resizing, normalization, and noise reduction on RGB frames before inference.
- **EnvironmentPerceiver**: Executes the semantic segmentation model on the Jetson Nano and produces pixel-level classification outputs.
- **ObstacleAnalyzer**: Combines segmentation results with depth information to estimate obstacle distance, position, and risk level.
- **NavigationManager**: Handles offline route generation, progress tracking, and route deviation detection using OpenStreetMap data.
- **TTSController**: Converts system messages, navigation instructions, and warnings into audio feedback.
- **STTController**: Processes button-activated voice commands and converts them into structured control actions.

- **SystemMonitor**: Oversees system health, monitors sensor availability, battery status, and error conditions, and triggers fail-safe behavior when required.

Data exchange between these components is performed using structured data objects such as timestamped frame packets, obstacle descriptors, and route nodes. This object-oriented approach supports clean separation of concerns and simplifies debugging and future development.

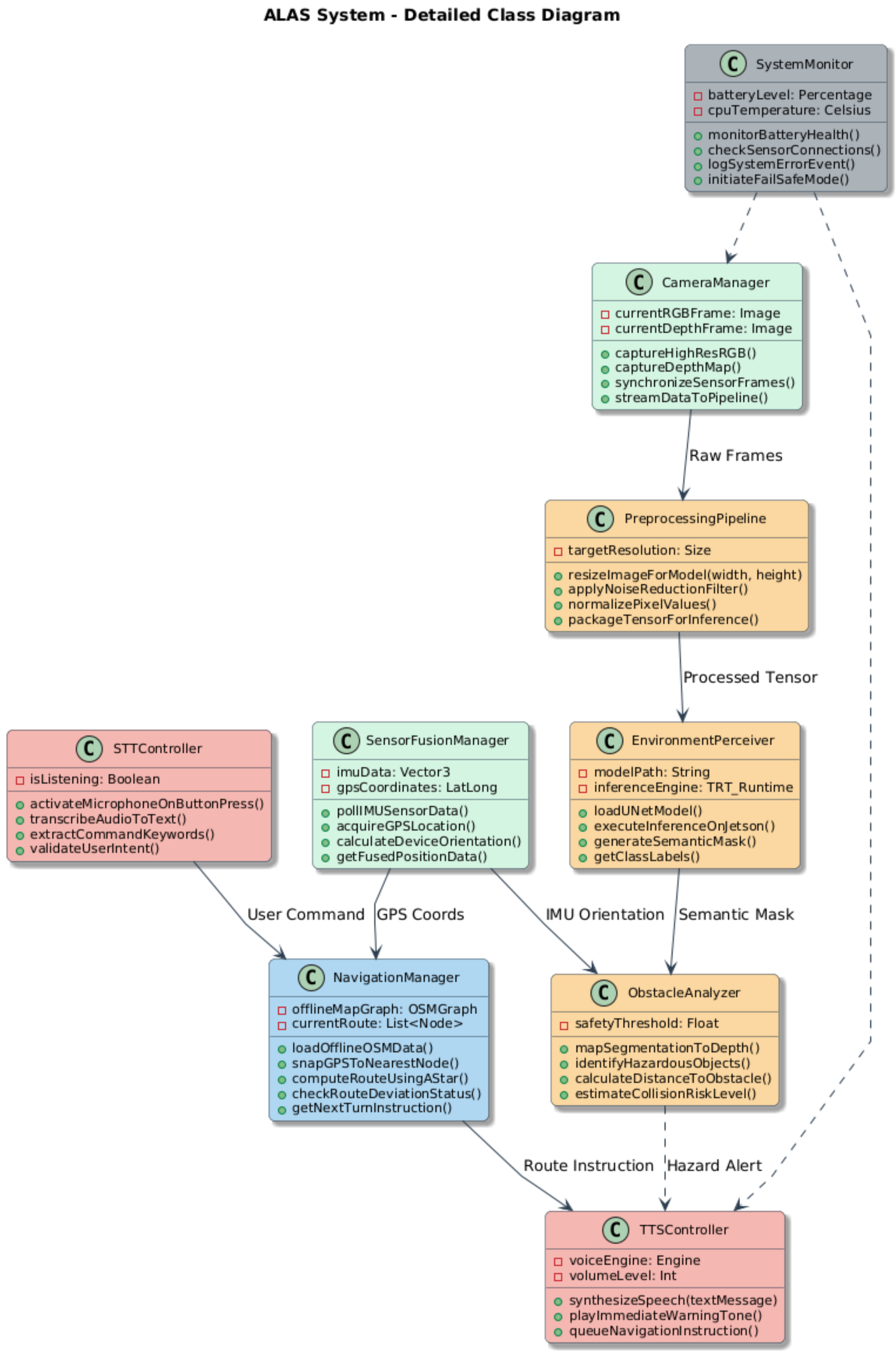

Diagram 3.1. Detailed Class Diagram

3.2 User Classes and Characteristics

- The primary users are visually impaired individuals who rely on assistance for safe and independent navigation in outdoor and semi-structured environments. Their main responsibility is to operate the device in daily life by following the auditory and haptic feedback provided by the system.

- **User Characteristics**:
  - Require hands-free operation and low-latency feedback.
  - Prefer simple, natural voice guidance instead of complex commands.
  - Rely on feedback alerts for obstacle proximity and direction cues.
  - Expect stable performance in noisy or bright outdoor conditions.
  - Desire compact, lightweight and comfortable wearable hardware.
  - Need to be easy for setup and initiation.
  - Should be privacy-preserving design which works locally.

- **User Preferences:**
  - Language: Primarily English voice assistance, with optional Turkish support.
  - Output modality: Audio (Text-to-Speech) as the primary channel, haptic as secondary.
  - Interface simplicity: Voice-based and several button control.
  - Feedback priority: Obstacle alert, direction directives and describe environmental awareness.

- For speaking technical capabilities of the system most users are expected to have basic experience with smartphones or simple electronic devices. They can handle operations such as powering the device on/off, charging it, pressing buttons, and giving voice commands. Advanced configuration or setup tasks assumed to be done by system automatically.

3.3 Operating Environment

- ALAS operates on a Jetson Nano–based embedded computing platform running a Linux-based operating system. The software stack is primarily implemented in Python and uses optimized libraries for AI inference, sensor handling, and audio processing.

- The system is intended for outdoor pedestrian environments, including sidewalks, crosswalks, pedestrian paths, and open urban areas. It is designed to function under varying environmental conditions such as changing lighting, moderate background noise, and GPS accuracy fluctuations.

- The device operates without network connectivity during runtime. All required data, including OpenStreetMap graphs and AI models, are stored locally on the device. During development and testing phases, a USB interface may be used for debugging and software updates.

3.4 Design and Implementation Constraints

The design and implementation of ALAS are subject to several constraints that directly influence architectural and technological decisions:

- **Computational constraints**: The system must operate within the limited processing and memory resources of an embedded platform, requiring lightweight AI models and efficient pipelines.
- **Energy constraints**: Battery-powered operation limits continuous processing time and necessitates power-efficient algorithms and hardware usage.
- **Real-time constraints**: Navigation guidance and obstacle warnings must be delivered with low latency to ensure user safety.
- **Form factor constraints**: The system must remain lightweight, portable, and suitable for wearable use.
- **Ethical and privacy constraints**: All user data must be processed locally to prevent privacy risks associated with cloud-based solutions.
- These constraints motivate the use of edge AI, modular software design, and offline navigation strategies.

3.5 Assumptions and Dependencies

The following assumptions are made for the successful operation of the ALAS system:

- Users have basic familiarity with simple voice commands and button-based interaction.
- OpenStreetMap data is preprocessed and stored locally on the device prior to use.
- GPS accuracy is sufficient to map the user's position to the nearest pedestrian route node.
- The operating environment is primarily outdoor and pedestrian-oriented.

System dependencies include:

- Jetson Nano hardware and compatible Linux drivers.
- RGB camera, IMU, GPS, microphone, speaker, and battery modules.

- AI inference frameworks such as TFLite or ONNX runtime.
- Locally stored OpenStreetMap data and routing libraries.

4. System Features

4.1 Functional Requirements

4.1.1 Camera & Sensor Acquisition Module

FR-1.1 System must collect synchronized color and depth frames from the RGB-D camera.

FR-1.2 Depth sensor shall measure at least 0.2m – 4m distance.

FR-1.3 GPS location data must be automatically added to the route when a new location is acquired.

FR-1.4 If the camera or GPS/IMU connection is lost.

FR-1 .5 If one of the sensors cannot be read the system must create a log and system alert the user with an audible signal within at least 1 second.

FR-1.6 Data must be timestamped and synchronized in the pipeline.

FR-1.7 All sensor and camera data shall be timestamped and synchronized across modules.

4.1.2 Image Preprocessing Module

FR-2.1 RGB an depth images should be scaled to ABCxABC.

FR-2.2 A Gaussian/median filter should be used for noise reduction.

FR-2.3 The depth channel should be normalized and given to the model as an additional channel.

4.1.3 Navigation and Route Creation Module

FR-3.1 The system must operate with the local database without an internet connection.

FR-3.2 The user's GPS location must be mapped to the nearest node on the network.

FR-3.3 The route must be calculated using Dijkstra algorithm.

FR-3.4 The once route is created, the road should be saved in memory for saving CPU cycle.

FR-3.5 The system must create feedback like "Turn right in 10 meters".

FR-3.6 The once feedback is created, the system should send feedback to TTS module.

FR-3.7 The system must check if the user is in the route or not in every 3 seconds.

FR-3.8 The navigation engine shall operate fully offline and generate routes using OSM and A* algorithm.

4.1.4 User Interaction Module

FR-4.1 TTS output must be arrangeable.

FR-4.2 STT must be active only when the button is pressed.

FR-4.3 STT outputs should create by same frame shape (Turn ***Right/Left*** after ***X*** meter, Route calculated).

FR-4.4 STT inputs must be created by keywords (Nearest ***X***, Cancel Route, Status, Stop/Pause).

FR-4.5 If the command is not understood TTS can output “Command not understood” but if understood “Command understood”.

FR-4.6 STT shall activate only when the button is pressed and shall achieve ≥75% accuracy.

4.1.5 System Management and Telemetry Module

FR-5.1 The system must automatically initialize all modules upon startup.

FR-5.2 Hardware status (Camera/IMU/GPS) must be continuously monitored.

FR-5.3 An audible warning through TTS must be given when system detect any error or critical message (Battery Level Drop, Sensor Connection Loss).

FR-5.4 The log system must create time-stamped error and command history.

FR-5.5 The system must be able to decide when it should switch to low power mode or high-power mode according to system status.

FR-5.6 Critical errors shall be logged and immediately announced to the user via TTS.

4.1.6 Hardware Integration Module

FR-6.1 The camera, GPS, IMU, speaker, microphone and battery must be fully integrated with the Jetson Nano.

FR-6.3 The system must enter a fail-safe state in the event of hardware failures.

4.2 Non-Functional Requirements

4.2.1 Performance Requirements

NFR-P1. Real-time inference

The semantic segmentation model shall run on the Jetson Nano at a minimum of 20 FPS, with a target operational range of 20–30 FPS.

NFR-P2. TTS latency

Text-to-Speech (TTS) output shall have a total end-to-end latency of less than 1 second.

NFR-P3. Sensor fusion latency

End-to-end processing of IMU, GPS, and camera data shall have a latency of less than 500 ms.

NFR-P4. Navigation update rate

Route deviation checks shall be executed at least once every 3 seconds.

NFR-P5. Preprocessing pipeline limit

Image preprocessing latency shall not exceed 20 ms.

NFR-P6. Power-efficient operation

The system shall operate under continuous load without entering thermal throttling conditions.

NFR-P7. Camera FPS

The camera data stream must be at least 10 FPS, yet the target must be 25 FPS.

NFR-P8. Sensor Latency

Sensor data (IMU, GPS) must be processed with a latency of less than 500 ms.

NFR-P9. Pipeline Latency

The pipeline delay should be less than 20 ms in total.

NFR-P10. CPU Optimization

The system must be designed to use less CPU cycle for AI modules can operate easily.

NFR-P11. TTS Delay

The TTS delay must be <1 second.

NFR-P12. STT Accuracy

STT performance rate must be at least 75%.

NFR-P13. Filters for Performance

For increasing STT performance system should use some filters.

NFR-P14. Operation Continuity
The prototype must provide at least 2 hours of continuous operation.

NFR-P15. Robustness of Modules
Cabling and connection modules must be resistant to vibration and drops in the field.

NFR-P16. FPS Boundary

The semantic segmentation model shall run at ≥20 FPS on Jetson Nano.

NFR-P17. Pipeline Limit

The image preprocessing pipeline shall complete in <20 ms.

NFR-P18. Latency

TTS output latency shall be <1 second.

4.2.2 Reliability Requirements

NFR-R1. Continuous operation

The prototype shall operate for at least 2 hours on battery power.

NFR-R2. Fault tolerance

If the camera, GPS, or IMU becomes temporarily unavailable, the system shall enter a fail-safe mode and notify the user via audio feedback.

NFR-R3. Data integrity

All camera and sensor data shall be timestamped and synchronized with a maximum error margin of less than 1%.

NFR-R4. Hardware robustness

Cables and connectors shall withstand vibration and accidental drops; a controlled shutdown shall occur in case of power failure.

NFR-R5. Crash resilience

If any module (perception, navigation, or TTS/STT) crashes, the system shall automatically recover the affected module without requiring a full system reboot.

4.2.3 Usability Requirements

NFR-U1. Minimal interaction

The system shall be operable using only one physical button and voice-based feedback.

NFR-U2. Clear spoken feedback

TTS outputs shall be produced at a sound level of 60–75 dB with high intelligibility.

NFR-U3. Accessible output format

Guidance commands shall follow a short and fixed structure, such as "Turn right in 10 meters."

NFR-U4. Easy learning curve

The system shall be learnable by an average user within less than 5 minutes.

NFR-U5. Noise robustness

STT accuracy shall remain at or above 75% under typical outdoor noise conditions (approximately 70 dB).

4.2.4 Security Requirements

NFR-S1. Local data processing

All image, audio, and sensor data shall be processed entirely on the device; no cloud transmission shall be permitted.

NFR-S2. Command security

STT shall activate only while the physical button is pressed, preventing unintended or malicious voice input.

NFR-S3. File system protection

System logs and route files shall be accessible only to internal system processes and not exposed to the user.

NFR-S4. Restricted system access

Root-level access and external software installation shall be disabled except in developer mode.

4.2.5 Maintainability Requirements

NFR-M1. Modular architecture

Perception, navigation, TTS/STT, and hardware I/O modules shall be independently maintainable and replaceable.

NFR-M2. Open-source compatibility

All modules shall be documented and designed to support open-source libraries and development tools.

NFR-M3. Logging accuracy

The logging subsystem shall record system events and errors with timestamps accurate to within 10 seconds.

NFR-M4. Hardware serviceability

The camera, IMU, GPS, and battery shall be replaceable without requiring modifications to other system components.

4.2.6 Portability and Scalability Requirements

NFR-PS1. Physical portability

The total weight of the wearable system shall remain below 300 grams.

NFR-PS2. Model interchangeability

Different TFLite models shall be swappable without retraining or major software modifications.

NFR-PS3. Expandable architecture

The system shall allow easy integration of additional sensors (e.g., LiDAR or ultrasonic sensors).

NFR-PS4. Platform flexibility

The software shall run on Linux-based embedded platforms with minimal modification.

4.2.7 Safety Requirements

NFR-SF1. Critical obstacle warnings

For hazards exceeding a predefined danger threshold, the system shall issue an audio warning within 500 ms.

NFR-SF2. Low-battery protection

When battery level drops below 10%, the system shall warn the user and enter low-power mode.

NFR-SF3. Navigation safety

If a computed route conflicts with OSM data or becomes invalid, the system shall automatically recalculate a valid route.

NFR-SF4. Thermal safety

If the Jetson Nano temperature exceeds 80 °C, the system shall reduce CPU frequency and camera frame rate.

5. Requirement Analysis and Concept of Operations

5.1 Requirement Analysis:

5.1.1 User Requirements Analysis

UR-1: The user shall always receive clear and low-latency audio feedback.

UR-2: The system shall start all modules automatically upon power-on (plug-and-play operation).

UR-3: Primary interaction shall be performed using a single physical button and simple voice commands.

UR-4: The user shall operate the device without needing configuration, calibration, or technical adjustments.

UR-5: The device shall be lightweight, portable, and ergonomically suitable for daily outdoor use.

UR-6: The system shall operate reliably under outdoor conditions such as varying noise and lighting.

UR-7: All processing shall occur locally on the Jetson Nano to protect user privacy; no data shall be transmitted externally.

5.1.2 Environmental Requirements Analysis

ER-1: The RGB camera shall operate under varying lighting conditions (bright sunlight, shade, low-light).

ER-2: The system shall maintain navigation functionality even under fluctuations in GPS accuracy.

ER-3: The STT module shall remain functional under typical outdoor noise levels (60–75 dB).

ER-4: Hardware components shall withstand light impact, vibration, and temperature variations.

ER-5: The system shall operate continuously outdoors for at least 2 hours.

5.1.3 Hardware Requirements Analysis

HR-1: The RGB camera shall provide synchronized RGB data.

HR-2: The IMU shall deliver orientation and motion data with low latency.

HR-3: The GPS sensor shall supply a location accurate enough for mapping to the nearest node in the OSM graph.

HR-4: The speaker shall output intelligible TTS audio at 60–75 dB.

HR-5: All perception, preprocessing, inference, and navigation operations shall run on the Jetson Nano.

HR-6: The power system shall support at least 2 hours of continuous operation.

5.1.4 Reliability and Safety Requirements Analysis

RSR-1: If the camera, GPS, or IMU becomes temporarily unavailable, the system shall switch to fail-safe mode and warn the user.

RSR-2: If any software module crashes, it shall automatically restart without requiring a full system reboot.

RSR-3: Sensor timestamps shall be synchronized with a maximum error margin of <1%.

RSR-4: Physical connectors shall withstand movement, vibration, and everyday handling.

RSR-5: Battery-critical or hardware-fault conditions shall trigger immediate audio warnings.

5.1.5 Performance Requirements Analysis

PR-1: Semantic segmentation inference shall run at ≥20 FPS (target: 25–30 FPS).

PR-2: Sensor fusion latency (camera + IMU + GPS) shall be <500 ms.

PR-3: Route-deviation checks shall occur at least every 3 seconds.

PR-4: The system shall operate without thermal throttling under continuous load.

PR-5: Total TTS delay shall remain <1 second.

5.1.6 Constraints, Assumptions and Dependencies

Constraints

C-1: The system is designed only for outdoor pedestrian navigation.

C-2: Data collection will be limited; publicly available datasets will be combined with small-scale local samples.

C-3: The prototype is intended for a single user; cloud services and multi-user systems are out of scope.

Assumptions

A-1: The user has basic familiarity with voice commands and simple button interactions.

A-2: OSM graph data is preloaded on the device.

A-3: GPS provides an accuracy level suitable for pedestrian-level navigation.

Dependencies

D-1: Preprocessed OSM road graph and offline map data.

D-2: Jetson Nano OS, driver support for camera/GPS/IMU, and necessary libraries.

D-3: TFLite/ONNX runtime for executing the segmentation model.

5.2 Concept of Operations (ConOps):

5.2.1 User Roles – Primary User

- Executes destination selection, navigation start/stop and status query operations by issuing voice commands.
- Activates STT mode using the physical button on the device.
- Acts according to the TTS instructions provided by the system.

Figure 5.1. Example usage of STT mode start

5.2.2 High-Level Operational Description

The ALAS system provides real-time, offline navigation and obstacle awareness by processing sensor and camera data during user movement.

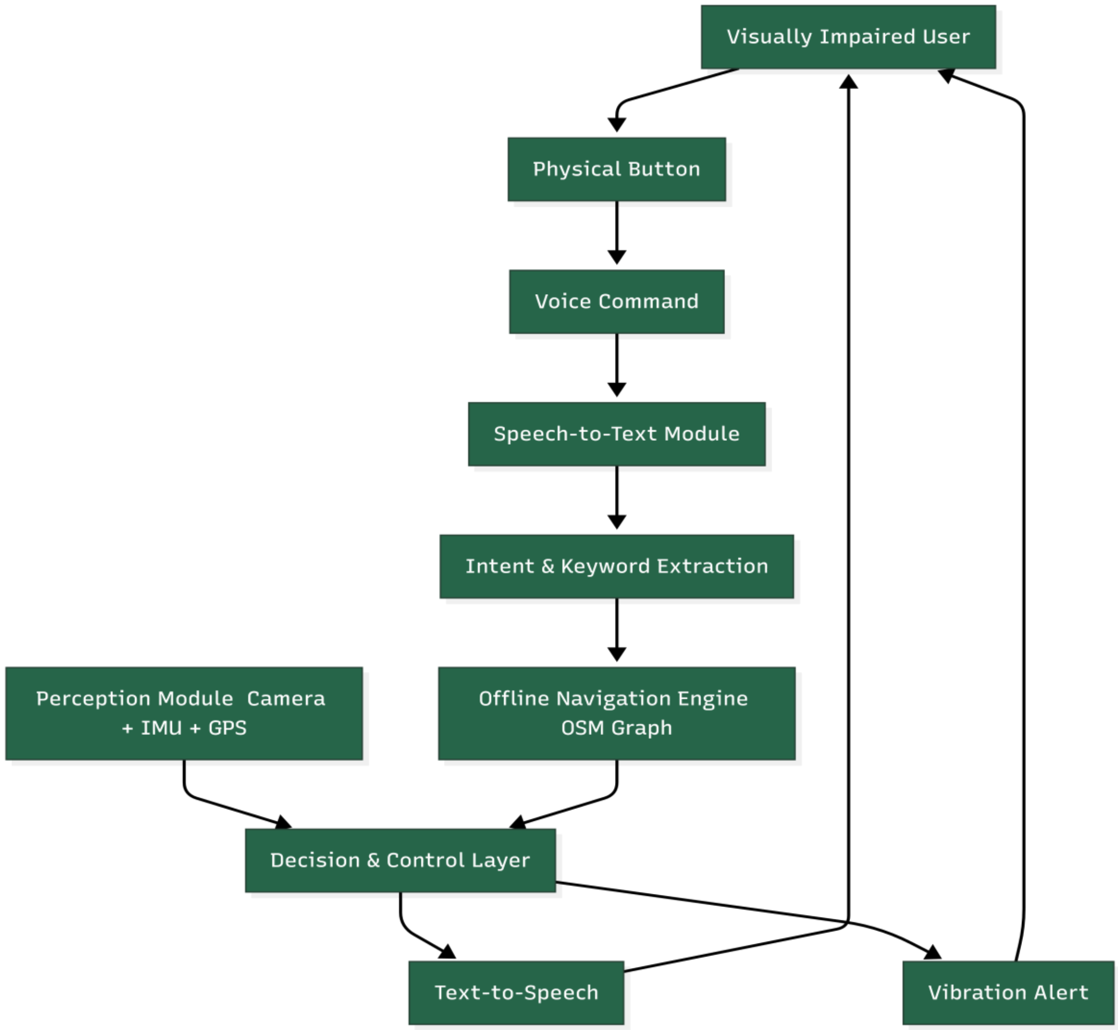

Table 5.1. Concept of Operation

The system operates fully offline and receives only two user inputs: voice commands and a physical button used to activate Speech-to-Text (STT) mode or power control.

System outputs include Text-to-Speech (TTS) guidance, obstacle warnings, status notifications, and optional vibration alerts.

Camera, IMU, and GPS data are continuously processed in real time during operation, requiring no manual configuration by the user.

5.2.2.1 Typical Use Scenario (End-to-End)

This diagram illustrates a typical end-to-end operational scenario of the ALAS system, starting from device activation and destination selection to real-time navigation, obstacle handling, route correction, and arrival confirmation.

The system continuously operates perception, navigation, and feedback loops during user movement, ensuring safe and uninterrupted guidance until the destination is reached.

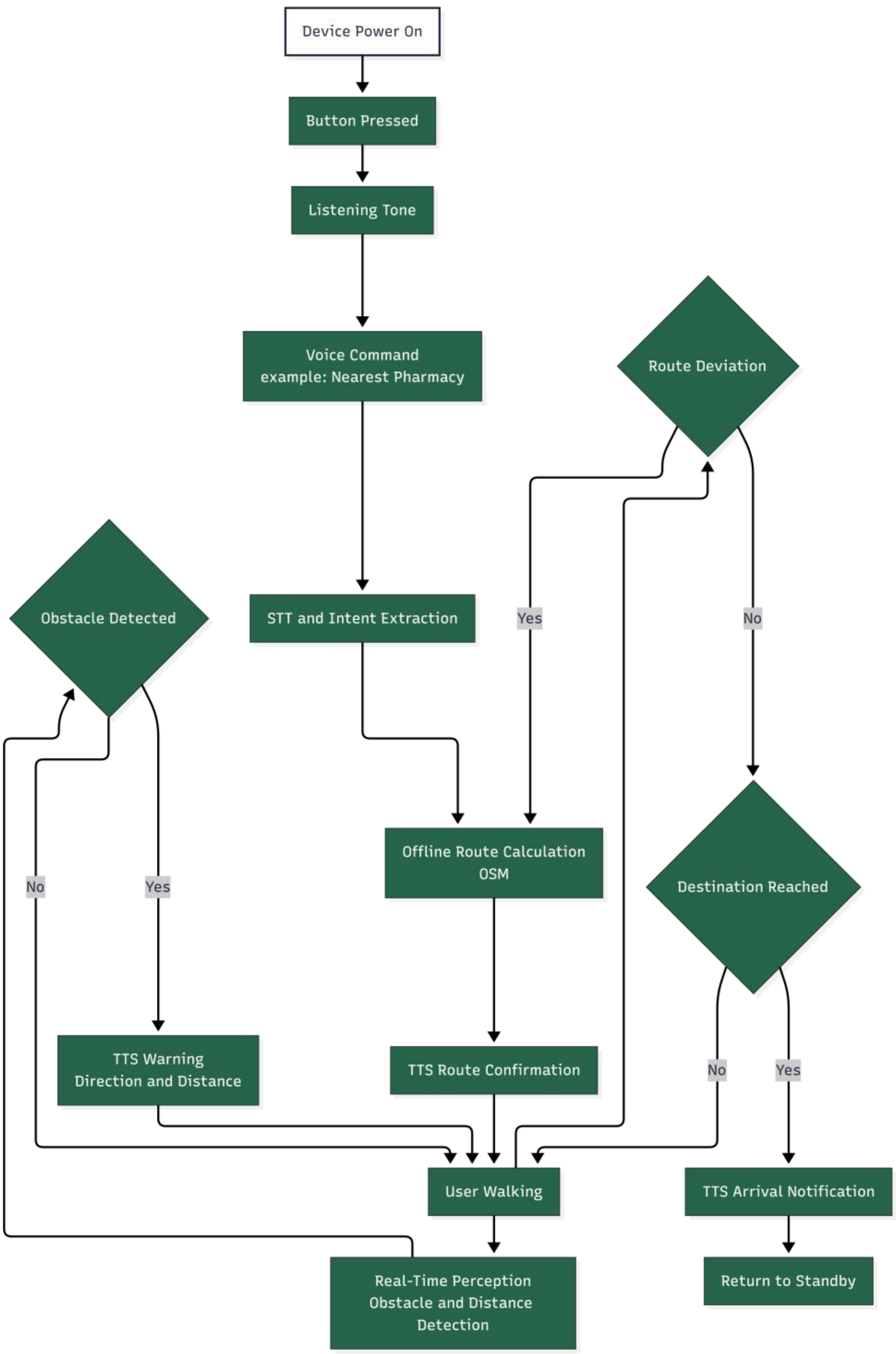

Table 5.2. Typical Use Scenerio

5.2.3 Operational Modes

The system operates in the following modes:

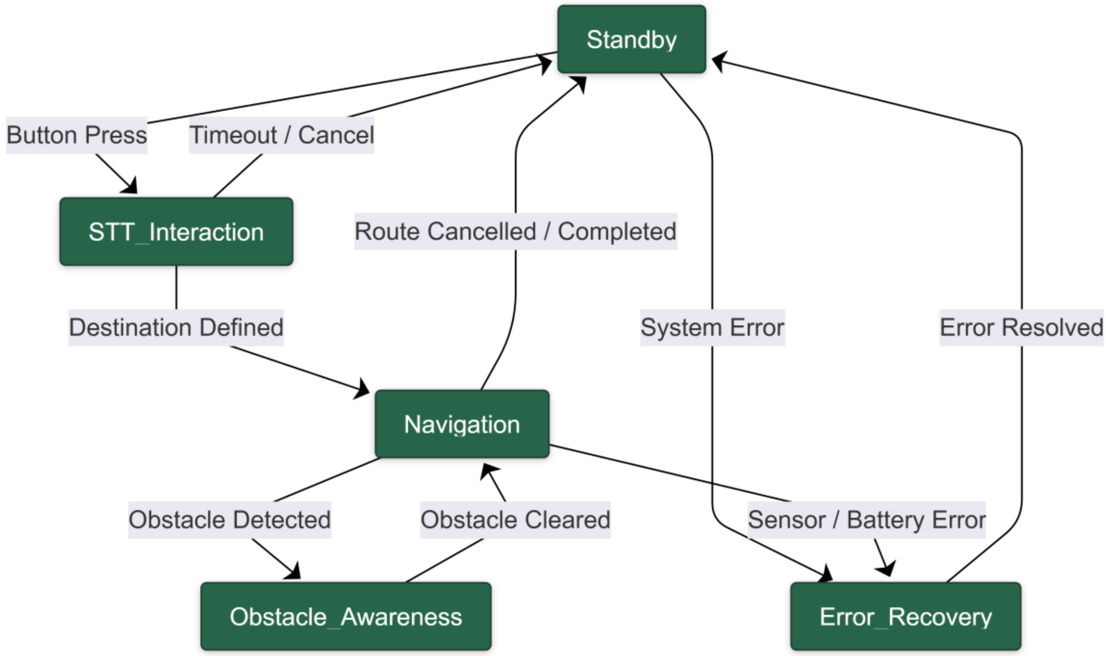

Table 5.3. Operational Modes

Mode-1 Standby Mode

- The system is powered on but not actively navigating.
- Sensors operate in low-power mode, and high-load processing tasks are paused.
- The user can initiate interaction by pressing the physical button.

Mode-2 STT Interaction Mode

- Activated when the user presses the button.
- The device emits a short "Listening Tone" when ready to listen.
- The user speaks the command.
- The spoken command is processed and the corresponding action is triggered.
- A TTS confirmation request may be generated for misunderstood commands.

Mode-3 Navigation Mode

- Starts after the user defines the destination.
- The system calculates an offline route and initiates guidance.
- The TTS periodically provides direction, turn and distance information.
- The route is updated as the user moves.
- If a route deviation occurs, a new route is generated.

Figure 5.1. Navigation Demonstration

Mode-4 Obstacle Awareness Mode

- Operates concurrently with Navigation Mode.
- Obstacles are detected using camera and depth data.
- According to obstacle distance and size system warn the user with speech.
- Obstacle direction (left, right, front) is indicated audibly.

Figure 5.2. Segmentation Demonstration

Mode-5 Error/Recovery Mode

- Activates in situations such as camera disconnection, GPS signal loss, insufficient battery level and sensor errors.
- After reporting the error via TTS, the system enters the safest possible state.

5.2.4 Major User Interactions

- **Start navigation:** Button press → "Nearest X / Go to X" command → route calculation → TTS confirmation.
- **Pause / resume:** "Pause" stops navigation prompts; "Continue" resumes guidance.
- **Cancel route:** "Cancel route" clears the active route and returns to standby.
- **Status query:** "Status" announces battery level, GPS availability, and whether navigation is active.
- **Error acknowledgement:** When an error is announced, the user can request "Repeat" to hear the last message again.

6. System Architecture and Preliminary Design

6.1 System Architecture

6.1.1 Hardware & Edge Computing Layer

This layer forms the physical foundation of the device, centered around the **NVIDIA Jetson Nano** to leverage its CUDA-accelerated processing capabilities for real-time AI inference.

- **RGB Camera:** Captures high-resolution visual data for environment mapping.
- **Inertial Measurement Unit (IMU):** Provides 6-DOF motion tracking and orientation data to refine localization.
- **GPS Module:** Facilitates global positioning for long-range navigation and route mapping.
- **Audio Interface (Mic & Speaker):** Serves as the primary HMI (Human-Machine Interface) for Speech-to-Text (STT) and Text-to-Speech (TTS).
- **Power Management:** An integrated battery system designed for high-current edge computing demands.

6.1.2 Perception & Computer Vision Layer

This layer transforms raw sensor data into actionable environmental intelligence.

- **Image Preprocessing:** High-speed normalization and resizing of RGB frames (target latency 20 ms).

- **Semantic Segmentation (U-Net Variant):** Executes pixel-level classification to identify walkable surfaces, sidewalks, and potential hazards.
- **Spatial Obstacle Analysis:** Fuses depth maps with segmentation masks to calculate the precise distance of objects, enabling proactive hazard avoidance.

6.1.3 Geospatial Navigation Layer

Leveraging offline capabilities, this layer ensures reliable pathfinding without requiring a constant cellular data connection.

- **OSM Integration:** Maps real-time GPS coordinates onto the nearest OpenStreetMap (OSM) graph nodes.
- **Pathfinding Engine:** Utilizes **A* Algorithm** to calculate optimal routes based on user-defined destinations.
- **Dynamic Progress Tracking:** Monitors deviations every 3 seconds, triggering automatic route recalculation if the user veers off-course.
- **Guidance Logic:** Translates spatial coordinates into intuitive, natural language voice commands.

6.1.4 User Interaction & Natural Language Layer

Focused on accessibility, this layer manages the multimodal communication between the user and the system.

- **TTS & STT Engines:** Optimized for low-latency performance to ensure real-time responsiveness.
- **Contextual Feedback:** A logic-based controller that confirms user commands and provides system status updates (e.g., "Route Found," "Searching for Signal").

6.1.5 System Orchestration & Telemetry Layer

Ensures operational integrity and fault tolerance across all software modules.

- **Module Initialization:** Manages the sequential boot-up of hardware drivers and AI models.
- **Health Monitoring:** Continuously audits sensor connectivity and thermal status of the Jetson Nano.
- **Telemetry & Logging:** Synchronizes event logs and error reports with timestamps for post-operation analysis.

6.1.6 Operational Workflow and Logic Flow

The following workflow describes the sequential logic of the system, from the initial power-on trigger to real-time navigation and safety intervention. This process is split into two parallel execution pipelines: **Global Navigation** and **Local Obstacle Avoidance.**

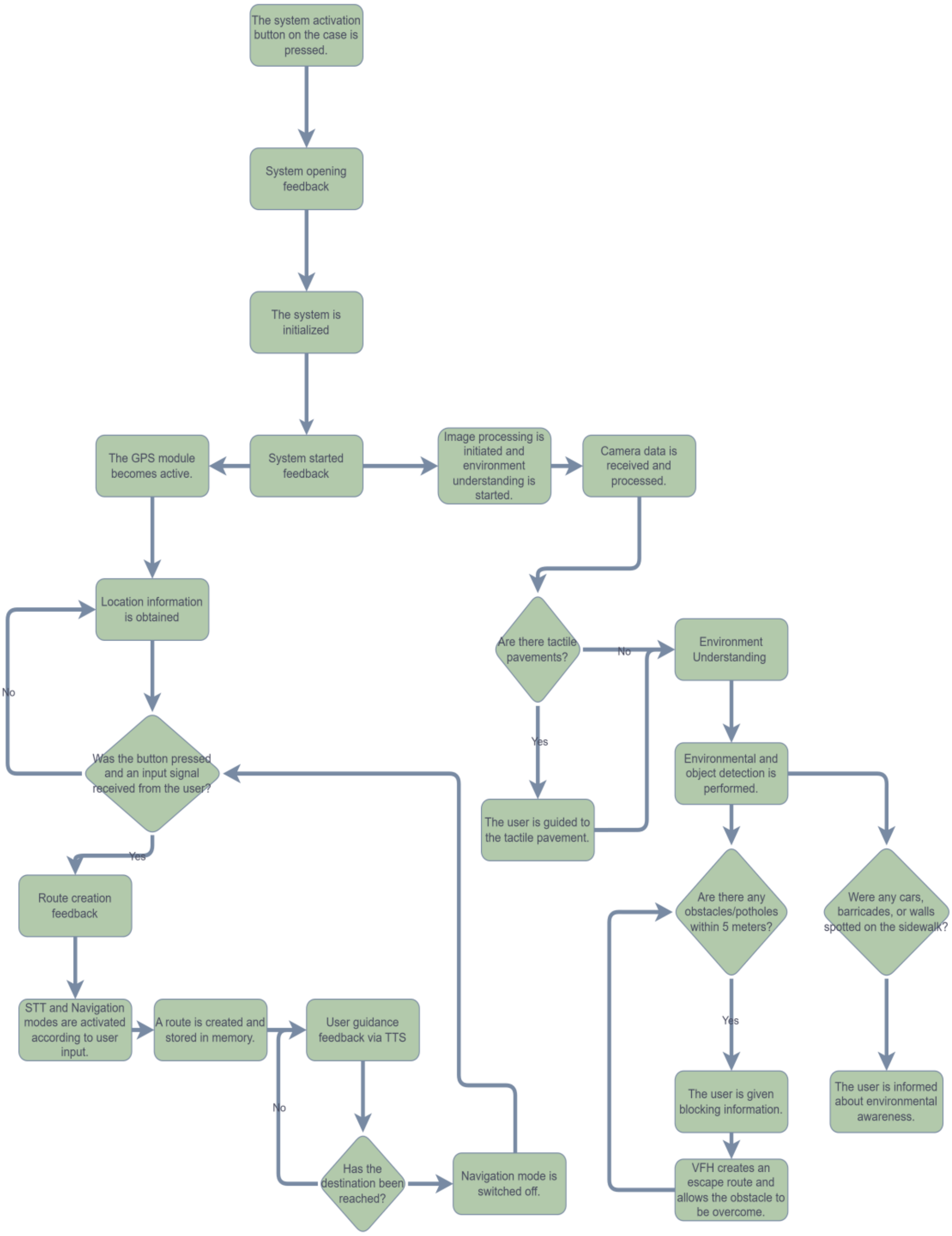

Diagram 6.1. Workflow Diagram

6.2 Design Trade-offs and Decisions

This section documents the major engineering trade-offs encountered during the design of the ALAS system. Each trade-off reflects a conscious balance between competing objectives such as real-time performance, safety, cost, usability, aesthetics, and system complexity. The final design choices prioritize reliability and user safety over maximal functionality or visual refinement.

6.2.1 Local Edge Processing vs. Cloud-Based Processing

Competing Objectives:

- High computational capability (cloud)
- Deterministic real-time response and reliability (edge)

Trade-off Analysis:
Cloud-based processing enables the use of large-scale AI models and virtually unlimited computational resources. However, this advantage directly conflicts with the real-time and safety requirements of an assistive navigation system. Transmitting camera frames and sensor data to the cloud introduces network latency and jitter, which are unpredictable and environment-dependent. During walking, even sub-second delays in obstacle warnings can significantly reduce user safety.

Local edge processing on Jetson Nano limits model complexity and requires careful optimization, but it guarantees deterministic latency, continuous operation without connectivity, and full user data privacy.

Selected Option: Local edge processing on Jetson Nano

Accepted Loss: Reduced model size and AI complexity
Accepted Gain: Predictable real-time performance, safety, and privacy

6.2.2 Jetson Nano vs. Raspberry Pi 5 AI Kit

Competing Objectives:

- Lower cost and general-purpose flexibility (Raspberry Pi 5)
- Stable AI acceleration and real-time vision performance (Jetson Nano)

Trade-off Analysis:
The Raspberry Pi 5 AI Kit offers an attractive cost profile and improved CPU performance; however, its AI acceleration relies on external or less mature software stacks. Under continuous perception workloads, this introduces uncertainty in inference latency and thermal stability.

Jetson Nano provides an integrated GPU and a mature CUDA/TensorRT ecosystem, enabling more predictable execution of computer vision pipelines. This comes at the cost of higher power consumption and slightly increased hardware cost.

Selected Option: Jetson Nano

Accepted Loss: Higher power usage and hardware cost
Accepted Gain: Reliable, sustained real-time AI performance

6.2.3 RGB-D Camera vs. RGB-Only Glasses-Integrated Camera

Competing Objectives:

- Accurate depth perception and simplified obstacle distance estimation (RGB-D)
- Aesthetic design, minimal form factor, and production-level cost efficiency (RGB-only)

Trade-off Analysis:
RGB-D cameras provide direct and reliable depth measurements, which simplify obstacle distance estimation and reduce uncertainty in safety-critical perception tasks. However, integrating an external RGB-D camera into a wearable glasses form factor introduces significant drawbacks. The physical size of such cameras results in a bulky appearance, requiring a large and visually intrusive module mounted on top of the glasses. This negatively affects wearability, aesthetics, and social acceptance of the product.

In addition to form factor limitations, RGB-D cameras substantially increase hardware cost. Their price makes large-scale production economically unfavorable and limits the feasibility of transitioning the prototype into an accessible consumer product.

As an alternative, glasses with a built-in RGB camera embedded directly into the frame were evaluated. This approach enables a significantly more compact and visually acceptable design, closely resembling conventional sunglasses. From a production perspective, RGB-only camera modules are considerably more cost-effective, making the system more suitable for large-scale manufacturing and broader user adoption. The main drawback of this approach is the absence of direct depth information, requiring depth cues to be inferred indirectly through computer vision techniques, which may be less robust under certain outdoor conditions.

Selected Option:
RGB-only glasses-integrated camera

Accepted Loss:
Reduced depth estimation accuracy and increased algorithmic complexity for distance inference

Accepted Gain:
Improved aesthetics, reduced hardware cost, enhanced wearability, and greater feasibility for product-level production

6.2.4 U-Net–Based Segmentation vs. YOLOv8-Based Perception

Competing Objectives:

- Pixel-level accuracy (U-Net)
- High frame rate and low latency (YOLOv8)

Trade-off Analysis:
U-Net architectures provide precise semantic segmentation masks, which are advantageous for detailed scene understanding. However, on embedded hardware, they require aggressive optimization to meet real-time constraints, often resulting in reduced frame rates.

YOLOv8-based perception offers faster inference and stable real-time performance, enabling object-level awareness (e.g., pedestrian, vehicle, obstacle) with lower computational overhead. While mask precision may be lower, the improved responsiveness is more critical for navigation safety.

Selected Option: YOLOv8-based perception

Accepted Loss: Reduced pixel-level segmentation accuracy
Accepted Gain: Higher frame rate and lower end-to-end latency

6.2.5 Offline OSM Navigation vs. Online Navigation APIs

Competing Objectives:

- Real-time map updates and traffic awareness (online APIs)
- Availability, privacy, and determinism (offline OSM)

Trade-off Analysis:
Online navigation services provide dynamic updates but depend on continuous internet access and external data sharing. For assistive devices, this dependency introduces unacceptable availability and privacy risks.

Offline OpenStreetMap-based navigation sacrifices live updates but ensures consistent route planning, immediate recalculation, and complete independence from network connectivity.

Selected Option: Offline navigation using OpenStreetMap (OSM)

Accepted Loss: No real-time traffic updates
Accepted Gain: Guaranteed availability and user privacy

6.2.6 Button-Activated STT vs. Always-Listening STT

Competing Objectives:

- Seamless interaction (always-listening)
- Power efficiency and privacy (button-activated)

Trade-off Analysis:
Always-listening STT systems increase power consumption and risk unintended activations, particularly in noisy outdoor environments. Button-activated STT enforces intentional user input, reduces unnecessary computation, and aligns with privacy-preserving design principles.

Selected Option: Button-activated STT

Accepted Loss: Slightly reduced interaction convenience
Accepted Gain: Improved privacy, power efficiency, and robustness

6.2.7 Audio-Centric Feedback vs. Haptic-Only Interaction

Competing Objectives:

- Rich semantic communication (audio)
- Minimal sensory load and power usage (haptics)

Trade-off Analysis:
Haptic feedback alone cannot convey complex navigation instructions or environmental semantics. Audio feedback enables detailed guidance at the cost of sensitivity to ambient noise. Given the navigation-focused use case, semantic clarity is prioritized.

Selected Option: Audio-centric (Text-to-Speech) feedback

Accepted Loss: Sensitivity to environmental noise
Accepted Gain: Clear and expressive navigation guidance

Summary of Trade-off Strategy

Across all design decisions, ALAS intentionally selects solutions that favor deterministic latency, operational safety, and system reliability, even when this requires accepting higher hardware cost, increased power consumption, or reduced model complexity. These trade-offs are aligned with the primary objective of supporting safe and independent navigation for visually impaired users in real-world outdoor environments.

7. Professional Development And Lifelong Learning

The ALAS project provides an extensive environment for the development of professional competencies and lifelong learning skills required in modern engineering practice. Throughout the project, team members engaged with real-world problems faced by visually impaired individuals and applied interdisciplinary knowledge spanning embedded systems, artificial intelligence, computer vision, and human–computer interaction. This process not only strengthened technical expertise but also fostered a mindset oriented toward continuous improvement, research, and ethical engineering practice.

From a professional standpoint, the project required the use of industry-standard tools, such as Jetson Nano for edge computing, Python-based AI frameworks, and offline navigation technologies derived from OpenStreetMap. Working with these tools improved the team's ability to design and implement resource-efficient algorithms, optimize deep learning models for embedded hardware, and structure modular architectures that follow engineering best practices. These skills reflect real industry requirements for developing deployable, reliable, and low-latency embedded AI systems.

The project also contributed significantly to the development of research literacy. As the solution involved semantic segmentation, obstacle analysis, and multimodal sensor fusion, it required the team to explore academic literature, evaluate methodological trade-offs, and adapt existing models to a new application domain. This iterative exploration enhanced the team's ability to interpret scientific findings and integrate them into practical engineering solutions—an essential component of lifelong learning.

Ethical and social considerations further shaped professional growth. Designing assistive technology demands awareness of privacy, accessibility, inclusiveness, and user safety. The requirement to process all data locally on Jetson Nano emerged not only as a technical choice but also as a response to ethical principles regarding user privacy. This alignment with international engineering ethics standards reinforces responsible decision-making habits that extend beyond the scope of the project.

The collaborative nature of the work encouraged continuous learning within the team. Members were required to communicate effectively, document system interfaces, and conduct systematic testing in real-world conditions. These activities strengthened soft skills such as teamwork, technical communication, and problem-solving under constraints—all essential qualities for long-term professional development.

Finally, the rapidly evolving field of artificial intelligence necessitates that engineers maintain an ongoing commitment to learning. By working on a system heavily dependent on AI and embedded technologies, the team gained awareness of the need to follow emerging techniques in lightweight model design, real-time inference optimization, and assistive technology regulations. This project therefore served not only as a technical achievement but also as a foundation for sustained lifelong learning in a continuously shifting technological landscape.

8. Teamwork and Communication

Effective teamwork and structured communication played a central role in the successful execution of the ALAS project. The multidisciplinary nature of the system—combining embedded hardware, computer vision, artificial intelligence, offline navigation, and user-interaction design—required the team to coordinate tasks efficiently while maintaining a shared understanding of the project's objectives and constraints.

9. Team Roles and Collaboration

The team was organized into complementary roles to ensure balanced workload distribution and domain specialization:

9.1 Embedded Systems & Hardware Integration:

Responsible for Jetson Nano configuration, sensor interfacing (RGB-D camera, IMU, GPS), power management, and physical prototyping.

9.2 Computer Vision & AI Development:

Focused on dataset preparation, model training and optimization, preprocessing pipeline design, and real-time inference integration on the Jetson Nano platform.

9.3 Navigation & Software Architecture:

Managed OSM data processing, Dijkstra-based routing, system modularization, synchronization between perception, navigation, and TTS/STT modules, and ensuring software scalability.

9.4 User Interaction & System Usability:

Designed the audio feedback structure, defined user interaction flows, implemented TTS and STT mechanisms, and evaluated usability from the perspective of visually impaired users.

Collaboration was maintained through iterative development cycles. Weekly progress reviews allowed members to present their updates, synchronize tasks, resolve blockers, and refine design decisions. This approach fostered accountability and ensured that the system evolved cohesively across all modules.

Team members were encouraged to share knowledge across roles—computer vision developers supported navigation design, while embedded systems developers contributed to preprocessing and optimization strategies. This cross-functional cooperation enhanced overall system quality and strengthened the team's engineering versatility.

9.5 Communication and Reporting:

Clear communication practices were essential to coordinate parallel tasks and maintain consistency across modules. The team adopted structured communication channels:

9.5.1 Weekly Meetings:

Formal sessions were conducted to evaluate progress, assign new tasks, and review integration risks. These meetings also served as decision-making checkpoints for architecture changes and module-level trade-offs.

9.5.2 Shared Documentation Environment:

Technical documents, requirements lists, interface definitions, model performance logs, and test results were stored in a collaborative platform accessible to all members. This ensured transparency and avoided duplicated work.

9.5.3 Version Control Practices:

A shared Git repository facilitated code management, modular integration, and consistent software version tracking. Branching strategies were applied to isolate development tasks and enable smooth merging during integration phases.

9.5.4 Continuous Communication Channels:

Instant messaging groups allowed rapid exchange of technical questions, debugging discussions, and real-time coordination during field tests.

Reporting activities included preparing requirement specifications, writing design documentation, generating test reports, and producing periodic summaries for the academic advisor. These structured reporting practices reinforced professional communication habits and supported traceability across the project lifecycle.

The combined effect of clear communication, well-defined roles, and collaborative problem-solving allowed the team to integrate complex modules—perception, navigation, interaction, and system management—into a coherent and functional assistive technology prototype.

10. External Interface Requirements

10.1 User Interfaces

The user interface is designed for visually impaired individuals and therefore prioritizes accessibility, simplicity, and minimal physical interaction.

Audio Output Interface (TTS)

- The primary communication channel is spoken feedback generated by the Text-to-Speech module.
- Navigation instructions follow a structured format (e.g., *"Turn right in 10 meters"*).
- System warnings (low battery, sensor errors, route recalculation) are delivered immediately and clearly.

Button-Based STT Activation

- A single tactile button activates Speech-to-Text mode.
- The button must be easily identifiable by touch and operable with one hand.
- Button press triggers an audible tone confirming activation.

Voice Command Interface

- Supported commands include keywords such as *"nearest pharmacy"*, *"cancel route"*, *"status"*, and *"stop"*.
- Invalid commands generate a neutral response such as *"Command not understood."*

Haptic Feedback (Optional Feature)

- Simple vibration signals may be used for critical alerts if required, though the main UI remains audio-based.
- These interfaces ensure that the system remains fully usable without visual feedback.

10.2 Hardware Interfaces

The system integrates multiple sensors and peripherals through Jetson Nano's hardware connectivity options. All hardware interfaces must remain stable under motion and outdoor conditions.

Camera Interface (RGB)

- Connected via USB or CSI depending on the camera model.
- Provides synchronized RGB and depth frames at 10–25 FPS.
- The interface must support continuous streaming without frame drops.

IMU Interface

- Communicates via I²C or SPI.
- Supplies accelerometer and gyroscope data used for orientation tracking.
- Low-latency data transfer is required for real-time sensor fusion.

GPS Interface

- Connected through UART serial communication.
- Outputs NMEA sentences for real-time position tracking.
- The interface must maintain stable throughput even in low-signal environments.

Audio Interface

- Microphone: Connected via USB or audio jack for STT input.
- Speaker: Connected via audio output interface, required to deliver 60–75 dB TTS output.

Power Interface

- The battery module supplies power to the Jetson Nano and all sensors.
- Must support hot-plug-safe connectors and short-circuit protection.

10.3 Software Interfaces

Software interfaces define how internal modules exchange data and how algorithms interact with external software resources.

Perception API

- Provides segmentation masks, depth-based obstacle distances, and environment classifications.
- Outputs standardized data structures consumed by the navigation module.

Navigation API

- Accepts GPS coordinates and perception outputs.
- Returns structured navigation commands, route progress updates, and deviation alerts.

TTS/STT Interface

- TTS receives text commands via a simple function-based API and returns audio playback.

- STT returns keyword tokens or structured command objects to the system controller.

Logging Interface

- All modules push timestamped logs to a shared storage directory.
- Supports event types such as errors, warnings, commands, system states, and module restarts.

Model Execution Interface (ONNX/TFLite Runtime)

- Loads the segmentation model into the Jetson Nano inference engine.
- Provides standard functions such as *load_model*, *run_inference*, and *optimize_output*.
- These software interfaces ensure modularity, allowing each subsystem to be developed and tested independently.

10.4 Communication Interfaces

- Explicitly describe communication protocols, standards, or methods your system uses for data exchange

The ALAS system is primarily designed for offline operation; therefore, external network interfaces are limited. However, certain internal communication channels are essential.

Internal Module Communication

Inter-process communication (IPC) or shared memory is used for high-frequency data exchange (e.g., camera → perception → navigation).

Lightweight messaging queues or sockets may be used for asynchronous events.

GPS Serial Communication

UART-based serial link provides continuous NMEA messages to the navigation system.

Sensor Communication (IMU)

I^2C/SPI interface transfers orientation data at high sampling rates for real-time fusion.

Offline Map Data Interface

OSM data is accessed locally from the file system via a read-only API used by the routing engine.

No internet connectivity is required at runtime.

Optional USB Interface for Development

Used during testing for debugging, logging, and firmware updates.

These communication interfaces ensure the system remains functional without external connectivity, maintaining reliability and user privacy.

11 Conclusions And Future Work

11.1 Conclusions

The ALAS project demonstrates the feasibility of developing an assistive navigation system as a complete product-level solution rather than a standalone academic prototype. Throughout the project, extensive literature research was conducted on existing assistive navigation systems and wearable guidance technologies. This preliminary research phase enabled the team to understand current approaches, technical limitations, and real-world deployment challenges, and highlighted the importance of grounding system design on prior scientific and industrial work.

The project provided hands-on experience in end-to-end product development, including requirements analysis, system architecture design, hardware–software co-design, and integration of perception, navigation, and interaction modules into a unified wearable platform. The system was implemented on an embedded edge platform and validated as a functional prototype capable of real-time perception, obstacle awareness, and offline navigation.

A modular system architecture was designed, enabling independent development and testing of perception, navigation, interaction, and system management subsystems. This modularity allowed parallel development and reduced integration complexity. Key engineering trade-offs were evaluated throughout the design process, particularly between real-time performance, power consumption, hardware constraints, and user usability requirements.

The project also provided substantial experience in collaborative engineering and project management. Team communication and coordination were maintained through regular interaction on WhatsApp and Microsoft Teams. A dedicated GitHub repository was established for version control, documentation, and progress tracking. Weekly meeting summaries and development reports were uploaded to the repository to ensure traceability and transparency. On average, biweekly online meetings were conducted to evaluate design decisions, review technical challenges, and assess project progress.

In parallel, the team actively participated in the HAVELSAN SUIT program and benefited from continuous mentorship throughout the development process. Regular biweekly meetings were held with the assigned mentor, during which system architecture, design decisions, and implementation strategies were critically reviewed. Feedback and technical critiques received during these sessions directly influenced subsequent design iterations, leading to several architectural refinements and performance optimizations. This mentorship-driven review process significantly strengthened the engineering rigor of the project and improved the overall system quality.

In addition to technical development, the team gained practical experience in time management, scope management, and milestone-driven project execution. Design alternatives were systematically evaluated and trade-off analyses were performed to support informed engineering decisions. This process strengthened the team's understanding of how real-world engineering projects are shaped by technical constraints, user needs, and resource limitations.

Several challenges were encountered during development, particularly in balancing real-time AI performance with power efficiency on embedded hardware and in achieving robust multi-sensor integration in outdoor environments. These challenges emphasized the importance of early prototyping, iterative testing, and continuous system validation. While the current implementation represents a usable and functional prototype, further engineering effort is required to transform the system into a fully deployable consumer-grade product.

Overall, the ALAS project resulted not only in a working assistive navigation prototype but also in a comprehensive learning experience in product-oriented system engineering. The project established a solid technical and methodological foundation for future development in wearable assistive technologies.

10.2 Future Work

While the current prototype successfully validates the core functional and architectural design of ALAS, additional engineering effort is required to transition the system from a research prototype into a mature, field-deployable assistive product. Several directions for future improvement and expansion have been identified:

10.2.1 Enhanced Perception and AI Models

Incorporating lightweight transformer-based models or attention-enhanced segmentation networks could improve accuracy without significantly increasing computational load. Collecting a domain-specific dataset from real outdoor environments would support better model generalization and robustness.

10.2.2 Multi-Sensor Fusion

Adding ultrasonic or LiDAR sensors could provide redundancy and improve obstacle detection reliability under challenging lighting conditions.
Advanced fusion techniques (e.g., Kalman filtering, probabilistic modeling) could enhance localization stability.

10.2.3 Indoor–Outdoor Hybrid Navigation

Extending the system to support indoor navigation would require alternative localization methods such as Wi-Fi fingerprinting, BLE beacons, or visual positioning.
A hybrid routing strategy could seamlessly transition between indoor and outdoor environments.

10.2.4 Improved User Interaction Models

Incorporating haptic feedback or spatial audio cues could create a more intuitive navigation experience.
Expanding the STT system to include natural-language understanding (NLU) would enable more flexible verbal commands.

10.2.5 Hardware Optimization and Form Factor

Developing a custom PCB and power-management module would reduce weight and increase battery efficiency.

Transitioning from a prototype configuration to a wearable glasses form factor would improve ergonomics and user comfort.

10.2.6 Safety Validation and Field Testing

Conducting structured usability studies with visually impaired participants would provide empirical data on system effectiveness.

Formal safety testing under varied outdoor conditions would help refine emergency responses and fail-safe mechanisms.

REFERENCES/SOURCES

1. Internet: TUBITAK 2242 University Students Research Project Competitions Project Guide, 2019, https://www.tubitak.gov.tr/sites/default/files/2750/2242-2019_proje-rehberi.pdf , Last Accessed: 04.04.2021.
2. Cetin, A.; Bulbul, E. (2020) Hierarchical Behavior Model for Multi-Agent System with Evasion Capabilities and Dynamic Memory. *ISPRS Int. J. Geo-Inf.*, 9, 279. https://doi.org/10.3390/ijgi9040279
3. Wolfe, J. M., Kluender, K. R., Dennis, L. M., Bartoshuk, L. M., Herz, R. S., Lederman, S. J., & Merfeld, D. M. (2015). *Sensation & perception (4th ed.)*. Sunderland, MA: Sinauer Associates.
4. State Planning Organization (2005). *Economic and social indicators* (1950-2004). Ankara: State Planning Organization, 312-314.
5. Bulut, H. (2001). Mass media and the spiral of silence. *Gazi University Journal of Faculty of Educational Sciences*, 32 (1-2), 1382-1385.
6. Freire, Paulo (1991). *Pedagogy of the oppressed*. (Trans. D. Hattatoğlu and Erol Özbek). Istanbul: Ayrıntı Publishing House. (The original work was published in 1982). 12-18.
7. Borman, W. C., Hanson, M. A., Oppler, S. H., Pulakos, E. D., & White, L. A. (1993, May). The people in organization. *Organizational Management*, 76-79.
8. Kahraman R. C., Borman, C., Hanımgil, M., Özler, H., Perçin, D., & Sergen, L. (1993). Factors playing a role in the determination of chronic heart disease. *Health Psychology*, 12(2), 76-80.
9. Lavric, A., Beguni, C., Zadobrischi, E., Căilean, A.-M., & Avătămăniței, S.-A. (2024). A Comprehensive Survey on Emerging Assistive Technologies for Visually Impaired Persons: Lighting the Path with Visible Light Communications and Artificial Intelligence Innovations. Sensors, 24(15), 4834. https://doi.org/10.3390/s24154834
10. eSight. (n.d.). *eSight 4*. Retrieved June 3, 2026, from https://www.esighteyewear.com/esight-4/
11. OrCam. (n.d.). *OrCam MyEye 3 Pro*. Retrieved June 3, 2026, from https://www.orcam.com/en-us/orcam-myeye-3-pro
12. Microsoft. (n.d.). *Seeing AI*. Retrieved June 3, 2026, from https://www.seeingai.com/
13. Aira. (n.d.). *Aira*. Retrieved June 3, 2026, from https://aira.io/
14. BlindSquare. (n.d.). *BlindSquare*. Retrieved June 3, 2026, from https://blindsquare.com
15. American Printing House for the Blind. (n.d.). *Nearby Explorer*. Retrieved June 3, 2026, from https://tech.aph.org/

16. WeWALK. (n.d.). *WeWALK smart cane*. Retrieved June 3, 2026, from https://wewalk.io/en/product/