
# Graduation Project

# Intelligent Mixed Reality Navigation Assistant for Smart Buildings with Multimodal AI Integration (CEREBRO)

**Submitted By**
**Team**

**1. Ahmed Mohamed Moussa - 222101392**
**2. Sandy Samy Samir – 222101524**
**3. Basma Ahmed Mahmoud - 221101164**

**Supervised by:**

**Associate Professor. Aya Zoghby**
**Dr. Waleed Mohamed**
**Dr.Mohamed Handosa**

**Faculty of Computer Science and Engineering**
**New Mansoura University**
**2025-2026**

**Graduation Project**

# CEREBRO

## Submitted By
## Team

| Student Name | Student Academic ID | Program | Track |
|---|---|---|---|
| Ahmed Mohamed Moussa | 222101392 | AIS | |
| Sandy Samy Samir | 222101524 | AIS | |
| Basma Ahmed Mahmoud | 221101164 | AIE | |

## ABSTRACT

Modern wearable technologies aim to enhance human–technology interaction; however, current smart glasses solutions remain limited in personalization, contextual awareness, and seamless multimodal integration. To address these limitations, we present an advanced AI-powered smart glasses system designed to improve daily communication, productivity, accessibility, and decision-making through intelligent, hands-free interaction.

The smart glasses integrate a wide range of cutting-edge technologies, including speech recognition, real-time multilingual translation, large language models, computer vision, augmented reality, GPS navigation, environmental sensing, and smart home connectivity. The system enables users to communicate across language barriers, record audio and video, capture images, take notes, browse the web, and access contextual information through a built-in heads-up display (HUD).

A YOLO11-based computer vision module, trained on custom datasets, performs real-time object detection and face recognition, allowing the system to identify members of the user's social circle and personalize interactions. Navigation is supported through a custom indoor GPS solution that combines manually mapped building layouts, graph-based routing, and Wi-Fi triangulation to deliver accurate guidance in complex indoor environments.

The speech interaction pipeline utilizes Whisper [14] for Arabic and English transcription, which is processed by a large language model to interpret user intent and generate structured JSON-based task instructions. Responses are delivered through a multilingual text-to-speech engine, ensuring natural and context-aware interaction in the user's preferred language. The hardware architecture is centered around an ESP32 module, providing Bluetooth and Wi-Fi connectivity to microphones, speakers, sensors, and the smart glasses interface.

A responsive mobile application and web dashboard complement the smart glasses by enabling smart home control, device management, analytics, and accessibility customization. The platform supports real-time data synchronization, user preferences, and system monitoring, and can function independently of the glasses for broader accessibility. The system is designed with inclusivity as a core principle, offering voice commands, haptic feedback, visual aids, and personalized assistance for users with disabilities or mobility challenges. It also enhances driver safety by minimizing manual interaction and improving navigation awareness.

This project delivers a modular, scalable, and future-ready smart glasses platform that leverages AI, AR, and multimodal human interaction to enhance daily life across education, healthcare, business, travel, and customer service domains. In addition, a comprehensive competitor analysis of Amazon Echo Frames [18] and Meta Ray-Ban [20] glasses highlights the technological gaps addressed by our system, particularly in advanced computer vision, indoor navigation, smart home integration, and AI-driven personalization.

## ACKNOWLEDGEMENTS

We would like to express our heartfelt and deepest gratitude to everyone who stood by our side and supported us throughout the journey of this project.

First and foremost, we extend our sincere thanks to our beloved supervisor, **Dr. Aya Zoghby**, whose wisdom, patience, and heartfelt support have made a lasting impact on both our project and our personal growth. Her tireless guidance and unwavering dedication were lights that guided us through every challenge we faced.

A special and warm thank you to **New Mansoura University**, whose unwavering support, encouragement, and belief in our potential have been a cornerstone of our progress.

We are truly honored and grateful to **Prof. Dr. Moawad El-Kholy**, President of the University, for his visionary leadership and continuous inspiration, which have always motivated us to aim higher and work harder.

Our heartfelt appreciation goes to **Prof. Dr. Khaled Fouad**, Dean of the Faculty of Computer Science and Engineering, for his steadfast support, kind guidance, and genuine care. His presence throughout this journey has been a true source of strength and motivation.

We also extend our sincere thanks to **Dr. Mohamed Handosa & Dr. Waleed Mohamed** for their close follow-ups, thoughtful suggestions, and constructive feedback. Their insightful remarks played a vital role in refining and improving the quality of our work.

We would also like to thank **all the faculty members** who enriched us with knowledge and helped shape our academic and professional identities. Their passion for education and excellence continues to inspire us every day.

Finally, we owe an immense debt of gratitude to **our families**, who stood by us with love, prayers, and encouragement in every step of the way. Their belief in us gave us the strength to move forward, even in the toughest moments.

And to our incredible teammates, thank you for your dedication, support, and shared dreams. We also acknowledge the early contributions of **Aya Tarek, Alaa Adel, Mariam Khaled and Aya Wael** during the initial phases of the project. This achievement would not have been possible without your collaboration and determination.

This project is not just a result of effort and time. It reflects the support, love, and belief of all those who walked with us along this journey.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES ??????????????????????????

# 1. INTRODUCTION

## 1.1. Problem Statement

In today's fast-paced world, technology has become an integral part of daily life. However, users face several challenges when interacting with various technological devices, particularly in daily communication, productivity, and navigation. Key problems include:

1. Language Barriers: Users often struggle to communicate effectively due to differences in spoken or written language, especially in international environments or while traveling.
2. Task Distraction: Managing multiple tasks at once—such as note-taking, capturing photos, recording audio, or checking messages—can overwhelm users and reduce efficiency.
3. Lack of Intelligent Assistance: Most devices do not provide a personal smart companion capable of understanding user needs and offering real-time guidance.
4. Indoor Navigation Limitations: In large or complex buildings like universities, hospitals, or corporate campuses, finding the desired destination can be difficult without accurate navigation support.

The proposed system, Smart Glasses, addresses these challenges by combining AI-driven smart assistance, speech recognition, real-time translation, augmented reality, indoor navigation, and environmental and health monitoring sensors into a single portable and ergonomic device.

## 1.2. Project Purpose

The primary goal of the Smart Glasses project is to create a productive, intelligent, and user-friendly environment that enhances daily life by:

- Enabling seamless hands-free communication through speech recognition and translation features.
- Providing a smart companion capable of understanding tasks, managing schedules, setting reminders, and offering personalized recommendations.
- Improving navigation in complex indoor and outdoor environments using GPS, computer vision, and graph-based pathfinding algorithms.
- Supporting users with disabilities through voice commands, visual aids, haptic feedback, and gesture recognition.
- Allowing immersive experiences with augmented reality overlays for both productivity and entertainment purposes.
- Continuously adapting to user behavior using machine learning algorithms for personalized interaction and decision support.

The project aims to reduce daily friction, enhance productivity, and improve the overall quality of life, providing a comprehensive solution that integrates communication, navigation, health monitoring, and intelligent assistance in one device.

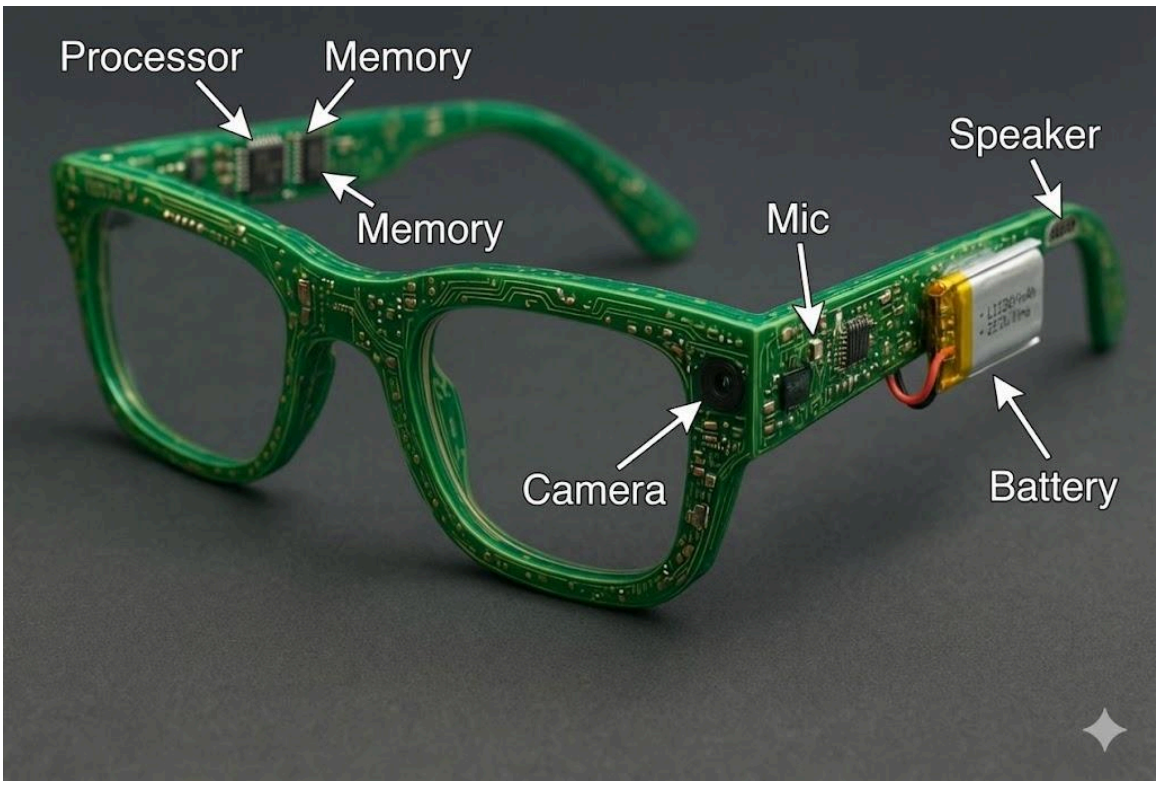



Figure 1: Project Vision

### 1.3. Project Scope

The scope of the Smart Glasses project includes:

1. Hardware Development:

    - Wearable device platform with microphone, speakers, camera, display, and sensors.
    - GPS module for location-based services and indoor positioning.
    - Low-latency wireless connectivity (Wi-Fi, Bluetooth).

2. Software Development:

    - Speech recognition and text-to-speech (TTS) systems.
    - Natural Language Processing (NLP) for task understanding.
    - AI-powered smart companion for scheduling, reminders, and assistance.

- Machine learning models for adaptive and personalized user experience.
- Augmented reality interface for immersive visualization and interaction.
- Web browsing and dashboard interface for remote management.

3. User Interaction Features:

- Hands-free operation with voice commands and gesture recognition.
- Real-time translation for multiple languages.
- Environmental and health monitoring with alerts and recommendations.
- Seamless communication with loved ones.

The project is designed to cater to a wide range of use cases including education, healthcare, business, entertainment, travel, customer service, and accessibility for users with disabilities.

## 1.4. Objectives and Success Criteria of the Project

Project Objectives (expanded)

a. Enhance Communication

Provide robust real-time transcription and translation to remove language friction in live conversations and recorded media.

What this means: Continuous low-latency speech-to-text (STT) in the user's active language, with optional simultaneous translation into a target language; support for multi-speaker scenarios and punctuation/formatting in transcripts.

Implementation notes: Use an on-device or hybrid STT pipeline (edge model + cloud fallback) with language detection, speaker diarization (who spoke when), and punctuation restoration. Post-process with NER & context-aware normalization for names, dates, and domain terms.

b. Improve Daily Productivity

Enable multitasking by letting users capture notes, voice memos, photos and video, and quickly retrieve contextual information via HUD and voice/gesture controls.

What this means: Voice/gesture-triggered note-taking, automatic summarization of recorded meetings, fast media capture (single-command photo/video), and searchable transcripts synchronized with timestamps.

c. Provide Smart Assistance

Build an AI companion that understands context, follows multi-turn dialogues, manages schedules, sets reminders, answers queries, and provides empathic micro-interactions.

What this means: Context retention across sessions (short-term and long-term), calendar/notification integration, task suggestions, and safe emotional-support responses (non-clinical).

d. Enable Accurate Navigation

Combine GPS (outdoor) and Indoor Positioning Systems (IPS) using camera-based positioning (visual localization), Wi-Fi/BLE triangulation, and graph-based path planning for reliable route guidance.

What this means: Outdoor turn-by-turn guidance plus indoor floor-plan localization with visual anchors, map matching, and shortest-path recommendations tailored to user preferences (e.g., step-free routes).

e. Ensure Accessibility

Design for inclusive use: voice commands, speech-to-text for deaf/hard-of-hearing users, text-to-speech for low-vision users, haptic feedback, and interaction alternatives to touch.

What this means: Multiple input/output modalities, adjustable font/UI scaling on HUD, caption customization, and integrations with hearing aids/assistive devices.

f. Offer Immersive AR Experiences

Overlay contextual, relevant information on the user's view (POI labels, navigation arrows, object labels, safety warnings) to improve situational awareness without cluttering the display.

What this means: Minimalist HUD design, context-aware content prioritization, and adaptive layering (show/hide based on user focus).

g. Continuously Adapt

Use machine learning to personalize behaviour and improve feature accuracy based on user interactions and anonymized usage patterns.

What this means: On-device fine-tuning / federated learning where appropriate, adaptive speech models for user accents, and personalization of recommendations and UI.

**Success Criteria (expanded & measurable)**

These are measurable acceptance criteria the project must meet to be considered successful:

1. System Reliability & Integration

- All hardware and software components must operate without critical failures during a standard 8-hour mixed-use scenario (recording, navigation, calls, AR use).
- Metric: < 2% system-crash rate per 100 user-hours; mean time between failures (MTBF) ≥ target.

2. Speech Recognition & Translation Accuracy

    - Metric (speech): Word Error Rate (WER) ≤ 12% in quiet environments, WER ≤ 20% in noisy environments (street/cafe).
    - Metric (translation): BLEU/chrF comparable to baseline production translation (expected acceptable threshold: intelligible, with >80% user-rated adequacy in evaluations).
    - Evaluation: Use standard datasets for benchmarking and controlled user studies with native speakers for subjective ratings.

3. Navigation & IPS Performance

    - Outdoor: GPS-guided routing with arrival error ≤ 5 m under normal conditions.
    - Indoor: Localization accuracy ≤ 2–3 m in mapped indoor spaces; path guidance success rate ≥ 90% for target destinations (in tested buildings).
    - Evaluation: Field tests in multiple indoor environments, measuring localization error and successful arrival rate.

4. Latency & Responsiveness

    - Voice command latency (voice → action) ≤ 1.0 s for on-device actions; transcription streaming latency ≤ 500–800 ms (per chunk).
    - AR overlay update rate ≥ 30 FPS nominal for smooth visuals.
    - Evaluation: Automated profiling + real-world user latency tests.

5. Usability, Comfort & Acceptance

    - Metric: System must achieve an average SUS (System Usability Scale) score ≥ 70 in pilot studies and positive qualitative feedback on comfort for at least 80% of test participants wearing the device for 1 hour.
    - Evaluation: Usability tests, comfort questionnaires, and cognitive-load assessments.

6. Accessibility Effectiveness

    - Provide configurable captioning and TTS with user satisfaction ≥ 80% among participants with relevant disabilities in targeted trials.
    - Support alternative input (gesture, physical buttons) with ≥ 90% task-completion rate in accessibility-focused tasks.

7. Privacy & Security

    - Sensitive personal data processed locally by default; cloud storage uses end-to-end encryption and explicit user consent.
    - Metric: No critical privacy violations in penetration test; compliance checks (documented flow of sensitive data).

8. Continuous Improvement

    - Measurable improvement in at least one personalization metric (e.g., speech recognition WER for the user) after 4 weeks of adaptive usage.
    - Evaluation: Compare pre- and post-adaptation model metrics.

**Domain-Specific Additions (Education, Travel, Accessibility)**

Below are expanded details and success criteria for the three domains you asked to add.

A. Education — Features, Use Cases & Success Criteria

**Key features for education use-cases**

- Live lecture transcription with speaker diarization and timestamped notes.
- Real-time translation for multilingual classrooms.
- Live summarization and study-guide generation after class.
- AR overlays for visual aids (labeling diagrams, 3D models).
- Hands-free photo capture of whiteboards + auto-enhanced cropping.

**Example user flows**

- Student activates "Lecture Mode" → device records audio, displays live captions on HUD, and marks important moments when the student says "mark" or uses a gesture → after class, AI generates a summary and suggested study flashcards.

**Education success criteria**

- Lecture transcription WER ≤ 10% in lecture hall conditions.
- Automated summarization scores (ROUGE/BLEU-like proxy) reach acceptable readability in educator review (> 80% acceptability).
- Student acceptance: ≥ 75% of pilot students find generated notes usable for studying.

**Implementation notes**

- Use domain-adaptive language models for educational vocabulary (STEM terms).
- Provide export to common LMS formats (PDF, DOCX, CSV of timestamps).

B. **Travel — Features, Use Cases & Success Criteria**

**Key features for travel**

- Turn-by-turn outdoor navigation via GPS with HUD arrows and voice prompts.
- Indoor navigation in transit hubs (airports, malls) using IPS + CV anchors.
- Real-time POI labeling and quick translation assistance in conversations with locals.
- Travel-safety overlays: hazard warnings, environment alerts (e.g., high temperature).

**Example user flows**

- Tourist enables "Travel Mode" → device shows route overlays on HUD, reads next steps aloud, and suggests POIs nearby based on user preferences; in foreign language interactions, automatic live translation is offered.

**Travel success criteria**

- Outdoor navigation arrival error ≤ 5 m; route-follow success ≥ 90%.
- Indoor localization median error ≤ 3 m across tested transit hubs.
- At least 85% user satisfaction in pilot travel scenarios (usability, safety, helpfulness).

**Implementation notes**

- Provide offline maps and cached translations for areas with poor connectivity.
- Prioritize low-latency voice translation for conversational use.

### C. Accessibility for Users with Disabilities — Features, Use Cases & Success Criteria

**Key accessibility features**

- For deaf/hard-of-hearing: Live captions, speaker diarization, transcript export, vibrational/haptic alerts for incoming calls/events.
- For low-vision: Clear TTS, high-contrast HUD mode, audio scene descriptions (object labels read aloud).

- For motor impairments: Full voice and gesture alternatives; customizable single-gesture triggers; physical button fallback.
- Cognitive accessibility: Simplified UI mode, step-by-step guidance, and context reminders.

**Example user flows**

- A deaf user attends a family gathering: device displays live captions for multiple speakers and marks each speaker with a name when recognized.
- A visually impaired user navigates a crowded station: device announces upcoming turns and nearby POIs, and warns about obstacles.

**Accessibility success criteria**

- Caption accuracy and readability judged acceptable by ≥ 80% of participants with hearing impairments.
- TTS naturalness & clarity rated ≥ 4/5 by visually impaired testers.
- Alternative input methods achieve ≥ 90% task-completion rate for key actions across users with motor impairments.

**Implementation notes**

- Co-design with accessibility experts and users with disabilities during prototyping.
- Provide robust customization (font size, speech rate, haptic intensity, color schemes).

**Evaluation & Testing Plan (brief)**

1) Benchmarks & Lab Tests

   - Evaluate STT, translation, object detection (mAP), and localization on standard datasets and synthetic noisy conditions.
   - Measure latency, FPS, battery, and thermal performance in controlled lab runs.

2) Field Trials

   - Multi-scenario field tests (indoor venues, streets, transit hubs, classrooms). Collect telemetry, logs, and objective metrics.

3) User Studies

   - Usability studies (SUS), domain-specific trials for Education/Travel/Accessibility, and qualitative interviews.

4) Security & Privacy Audits

   - Threat modeling, penetration testing, and privacy impact assessment.

5) Iterative Improvements

   - Use pilot feedback and logged metrics to prioritize fixes; re-evaluate acceptance metrics after 2–4 sprint cycles.

**Deliverables (related to Objectives and Success Criteria)**

- Functional prototype with: STT + translation, HUD overlay, object detection demo (YOLO11), indoor/outdoor navigation demo, and basic smart companion features.
- Evaluation report with benchmark numbers (WER, localization error, SUS scores).
- Accessibility compliance checklist and pilot study results.
- Privacy & security documentation (data flow diagrams + consent UX).

### 1.5. Report Outline

## 2. RELATED WORK

This chapter explores the background research and technologies that form the foundation of the Smart Glasses System. The project integrates multiple fields—computer vision, indoor navigation, wearable hardware, natural language processing, intelligent speech interaction, smart home automation, and multimodal user interfaces. To provide a comprehensive review, this chapter examines existing systems, identifies their limitations, evaluates competitor technologies, and highlights the gaps addressed by the proposed solution.

## 2.1. Existing Systems

Comparison with:
**Cerebro , hololens** , **Meta Ray-Ban Glasses** [20] **and Amazon Amelia Smart Glasses** 4

1. **Microsoft HoloLens**

**Overview**

HoloLens [19] is a high-end Mixed Reality (MR) headset designed mainly for industrial, medical, and engineering environments. It offers advanced holographic projection, spatial mapping, and hand/eye tracking.

**Strengths**
- Industry-grade holograms and spatial mapping
- Accurate hand and eye tracking
- Suitable for training, simulation, and engineering visualization
- Stand-alone computing with no external device needed

**Weaknesses**
- Heavy and not comfortable for daily, long-term use
- Very expensive
- Limited battery life under continuous MR use
- Not optimized for real-time translation or lightweight everyday tasks

2. **Meta Ray-Ban Smart Glasses** [20]

**Overview**

Meta Ray-Ban focuses on lifestyle use: capturing photos, recording videos, making calls, and using Meta AI through voice commands.

**Strengths**
- Stylish, socially acceptable design
- High-quality camera and audio system
- Good battery life for light use
- Strong voice assistant through Meta AI

**Weaknesses**

- No visual display or HUD
- No AR overlays
- Limited on-device AI processing
- Not built for navigation, translation, or object understanding

3. **Amazon Amelia / Amazon Echo Frames** [18]

**Overview**

Amazon's glasses are voice-first devices built mainly around Alexa interactions. They emphasize audio tasks and smart home control rather than visual computing.

**Strengths**

- Excellent voice assistant (Alexa)
- Lightweight and comfortable
- Strong integration with smart home features
- Good for reminders, calls, and simple everyday tasks

**Weaknesses**

- No screen, HUD, or visual interface
- No camera or computer vision
- No AR or visual overlays
- Heavy dependence on cloud processing

**Final Comparison Summary:**

TABLE 1: COMPARES FOUR SMART GLASSES PRODUCTS

| Feature | Our Smart Glasses (Cerebro) | HoloLens | Meta Ray-Ban | Amazon Amelia |
|---|---|---|---|---|
| **Daily Comfort** | High | Low | High | High |
| **Display** | HUD + AR | Full MR | None | None |
| **Object Detection** | Yes (built-in) | With customization | Basic | No |
| **Indoor Navigation** | Yes | No | No | No |
| **Translation** | Real-time | No | No | No |
| **Social AI (faces, memory)** | Yes | Very limited | No | No |
| **On-Device Processing** | On Device + Cloud | On Device | Cloud | Cloud |
| **Target Use** | Everyday AI assistant + AR navigation | Industrial MR | Lifestyle | Voice assistant |

- Our Smart Glasses (Cerebro): Positioned as an everyday AI assistant with strong comfort, built-in object detection, navigation, real-time translation, and social AI features.

- HoloLens [19] : Focused on industrial mixed reality (MR) with a full MR display but lower comfort and fewer everyday AI features.

- Meta Ray-Ban [20] : A lifestyle product with high comfort but no display and only basic object detection.

- Amazon Amelia: Primarily a voice assistant with no display or object detection, focused on voice interaction.

Cerebro is highlighted for its balance of comfort, AI capabilities, and AR navigation, making it suited for daily use.

- Existing wearable and smart assistant technologies focus primarily on isolated functionalities. They offer speech interaction or augmented reality, but few combine multimodal AI capabilities in one consistent and user-friendly system.

### 2.1.1 Amazon Amelia Smart Glasses

Figure 2.1: Amazon Amelia Smart Glasses (Amazon Echo Frames) [18] – smart glasses with integrated voice assistant.(Source: Amazon official product page)

Amazon Amelia Smart Glasses are AI-powered smart glasses designed primarily for voice-based interaction through an integrated virtual assistant. The system focuses on productivity tasks, reminders, and hands-free voice commands, targeting lightweight daily assistance rather than advanced visual intelligence.

• Designed as AI-powered smart glasses with integrated voice assistant "Amelia."
• Focused mainly on voice-based interaction, reminders, and productivity tasks.
• Offer basic multimodal support (voice + simple visual cues), but no advanced computer vision.
• No built-in object detection, scene understanding, or real-time AR processing.
• Limited navigation support — no indoor mapping or route guidance.
• No advanced AI reasoning beyond cloud-based responses.
• Do not support AR overlays on the real-world environment.
• Smart home integration is minimal and voice-only.

Limitations:
Amelia Smart Glasses lack computer vision, AR overlays, indoor navigation, context-aware AI reasoning, and multimodal perception, making them less capable for education, accessibility, or complex travel assistance.

### 2.1.2 HoloLens 2

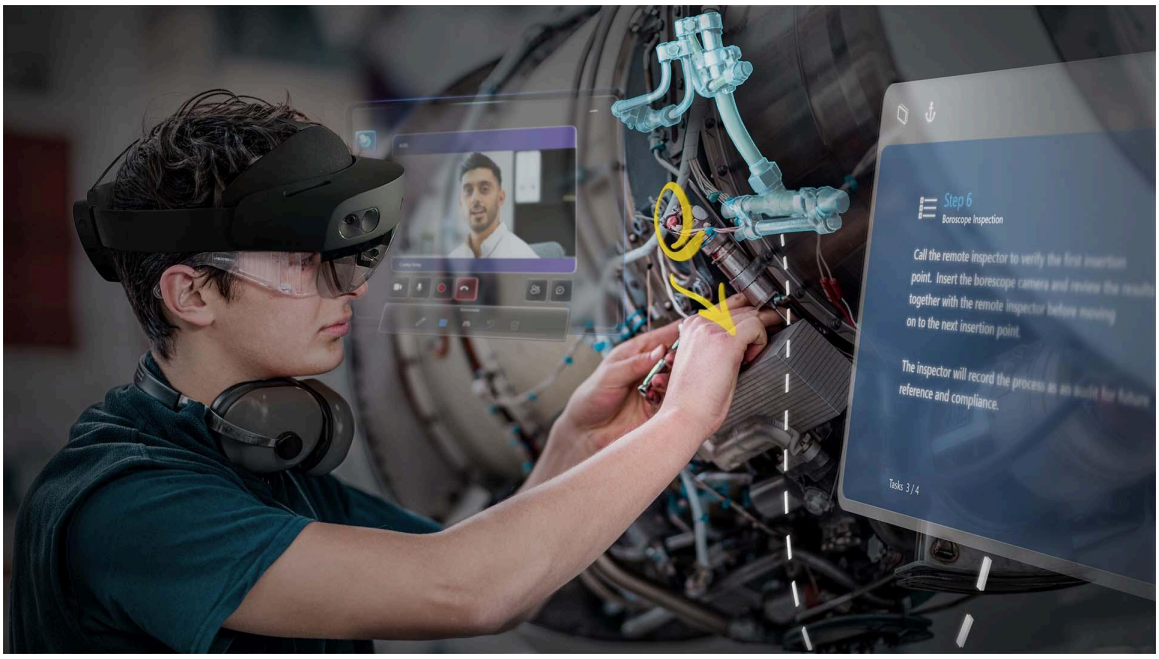



Figure 2.2: Microsoft HoloLens 2 mixed reality headset.
(Source: Microsoft official product page).

Microsoft HoloLens 2 [19] is an advanced mixed reality headset developed by Microsoft, designed to overlay digital holograms onto the real-world environment. It is primarily targeted toward industrial, medical, and enterprise applications that require spatial mapping and immersive augmented reality interaction.

• Advanced mixed reality headset with full AR overlays and spatial mapping.
• Supports hand tracking, eye tracking, and holographic interaction.
• Offers strong computer vision but optimized mostly for industrial and medical applications.
• Bulky form factor — not suitable for everyday travel, campus use, or continuous wear.
• Limited outdoor usability due to sunlight reflection and heavy optics.
• No built-in indoor navigation for public buildings (requires enterprise-level setup).
• Very expensive, not consumer-friendly, and not designed for accessibility-focused tasks.
• No native smart home integration for casual users.

Limitations:
Despite powerful AR, HoloLens 2 [19] is not practical as lightweight smart glasses, lacks portability, is unsuitable for travel, and doesn't provide AI-driven multimodal assistance like continuous speech translation, real-time accessibility tools, or small-form navigation support.

### 2.1.3 Ray-Ban Meta Smart Glasses (2024)

Figure 2.3: Ray-Ban Meta Smart Glasses in multiple styles.
(Source: The Verge product images)

Ray-Ban Meta Smart Glasses integrating a built-in camera, microphones, and open-ear speakers to enable voice-based AI interaction, live streaming, and social media content capture.

- Meta AI integration
- Voice-controlled assistant
- Built-in camera
- Live streaming support
- No indoor GPS or pathfinding
- No smart home dashboard
- No VR interaction
- Weak NLP action planning

Limitations:
Meta Smart Glasses provide strong social media integration but have no multimodal AI pipeline like the proposed system.

### 2.1.4 Indoor Navigation Apps (Google Maps Indoor, HERE Indoor Positioning)


Figure 2.4: Mobidev guide on indoor navigation

Example of an indoor navigation interface showing floor-level routing and positioning inside a large commercial space.

Indoor navigation applications such as Google Maps Indoor and HERE Indoor Positioning are software-based solutions designed to guide users inside large commercial and public buildings. These systems extend traditional outdoor navigation by providing floor-level positioning and indoor route guidance.

- Provide floor-level guidance in commercial spaces.
- Require Wi-Fi fingerprinting or BLE beacons.
- Not suitable for custom buildings without dedicated infrastructure.

Limitations:
They cannot be embedded into wearable glasses and require expensive sensor setups.

### 2.1.5 Smart Home Systems (Google Home, Alexa, HomeKit)


2.5 Smart Home Systems (Google Home, Alexa, HomeKit)

Example of a smart home control dashboard interface illustrating connected devices and settings accessible through smart home systems.

- Rely mainly on voice commands.
- No wearable integration.
- No AR or VR control interfaces.
- No customized device control or dashboards.

**2.2.Overall Problems of Existing Systems**

Across all competitors, several major limitations emerge:

2.2.1 Lack of Multimodal Interaction

Most systems either:

- Use speech only (Echo Frames) [18]
- Use AR only (Google Glass)
- Or use AI only (Meta Glasses)

No existing system combines:

✔ Speech → Text → LLM → Action → AR/VR
✔ Real-time vision understanding
✔ Indoor navigation
✔ Smart home control
✔ Mobile + Web dashboards

2.2.2 No Indoor Navigation

Existing glasses rely on GPS, which:

- Fails indoors
- Cannot detect stairs/elevators/rooms
- Cannot be customized for small buildings
- This project solves the problem by:
- Creating a manual building map
- Converting it to a graph representation
- Using JSON path planning
- Supporting stairs, elevators, and corridors

2.2.3 Limited Hardware Capabilities

Market devices lack:

- Affordable microcontrollers
- Customizable sensors
- Full integration with apps
- Real-time speech processing locally

The proposed system uses:

- ESP32 with WiFi/Bluetooth
- Microphone + speakers
- On-device fast communication
- Direct connection with app and web

2.2.4 Weak NLP Reasoning

Most competitors do NOT turn user commands into structured actions.

Our project uses:

- MCP NLP pipeline [1], [5], [6], [8]
- Converts natural commands → steps → JSON → model execution
- Works with speech, vision, navigation, and smart home

2.2.5 No Computer Vision

Competitors rarely include:

- YOLO object detection
- Real-time local inference
- Custom dataset training

Our system uses:

- YOLO11
- Custom dataset
- Real-time feature extraction
- Object detection integrated with navigation

### 2.3.Comparison Between Existing and Proposed Method

TABLE 2: COMPARES THE CAPABILITIES OF VARIOUS SMART GLASSES

| Feature | Amazon Echo Frames | Google Glass EE | Meta Glasses | Indoor Nav Apps | Proposed Smart Glasses |
|---|---|---|---|---|---|
| **Computer Vision** | ❌ | ❌ Limited | ✔ Basic | ❌ | ✔ YOLO11 (custom) |
| **Indoor Navigation** | ❌ | ❌ | ❌ | ✔ Only app-based | ✔ Full indoor routing |

| | | | | | |
|---|---|---|---|---|---|
| **Hardware Control** | ❌ | ❌ | ❌ | ❌ | ✔ ESP32 integration |
| **Smart Home** | Limited via Alexa | ❌ | ❌ | ❌ | ✔ Full dashboard |
| **Speech Recognition** | ✔ Alexa | ✔ Basic | ✔ Meta AI | ✔ | ✔ Whisper multilingual |
| **Text-to-Speech** | ✔ | ✔ | ✔ | ❌ | ✔ Bilingual |
| **LLM Reasoning** | Limited | ❌ | Moderate | ❌ | ✔ MCP + LLM pipeline |
| **AR Interaction** | ❌ | ✔ | ❌ | ❌ | ✔ AR / VR modes |

| | | | | | |
|---|---|---|---|---|---|
| **Customizatio** | Very limited | Limited | Very limited | Medium | ✔ Full customization |

- The table compares five different approaches or products across nine key features, highlighting what they can and cannot do.
- Amazon Echo Frames [18] : Primarily a voice assistant (Alexa) with basic smart home control and no vision or navigation features.
- Google Glass EE (Enterprise Edition): Offers basic speech recognition and AR interaction but lacks advanced computer vision, indoor navigation, and smart home integration.
- Meta Glasses: Have basic computer vision and Meta AI voice features but lack navigation, hardware control, and deep LLM reasoning.
- Indoor Nav Apps: These are software-only solutions (like phone apps) that provide navigation but lack hardware integration, AR, and most AI features.
- Proposed Smart Glasses: Outperforms all others in every category. It uniquely combines advanced, custom features like the YOLO11 computer vision model, full indoor navigation routing, ESP32 hardware integration for device control, a full smart home dashboard, multilingual Whisper [14] speech recognition, and an advanced LLM reasoning pipeline. It also offers full customization, which others lack.

**TABLE 2.1: COMPARISON OF PRODUCTS**

| **Category** | **Apple Vision Pro** | **Oppo Air Glass 3** | **Xiaomi Wireless AR** | **Huawei Vision Glass** | **Meta Quest 3** |
|---|---|---|---|---|---|
| **Device Type** | MR Headset | AR Glasses | AR Glasses (wireless) | Smart Display Glasses | Mixed Reality Headset |
| **Price** | $3,499+ | ~Estimated $800 | — | ~$430 | $499 (128GB) |
| **Release Year** | 2024–2025 | 2024 | 2023 Prototype | 2023 | 2023 |

| **Weight** | 750–800 g + 353 g battery | ~50 g | 40–120 g | Lightweigh t | ~515 g |
|---|---|---|---|---|---|
| **OS / Platform** | visionOS | ColorOS (phone) | Xiaomi AR OS | N/A | Meta OS (Android-b ased) |
| **CPU / Chipset** | Apple M5 + R1 | Phone-bas ed | Snapdrago n XR2 | Phone-base d | Snapdragon XR2 Gen 2 |
| **GPU** | 10-core GPU | Phone GPU | Adreno | Phone GPU | Adreno 740 |
| **RAM** | 16GB unified | — | — | — | — |
| **Storage** | 256–1TB | — | — | — | 128–512G B |
| **Display Type** | Micro-OLED | Waveguide | Micro-OLE D waveguide | Micro-OLE D | LCD + Pancake lenses |
| **Resolution (per eye)** | 23M pixels total | — | High PPD | 1080p | 2064×2208 |
| **Refresh Rate** | 90/96/100/120 Hz | — | — | N/A | 120 Hz |
| **Field of View** | — | — | — | N/A | — |
| **Brightness** | — | 1000+ nits | ~1200 nits | High | — |
| **Sensors** | 14+ sensors: LiDAR, IMUs, eye tracking | Touch, mic | 3 cams + IMU | Basic | RGB/IR cams, depth |
| **Cameras** | Stereo 3D + passthrough | — | Tracking cams | — | 2 RGB + 4 IR |
| **Tracking** | Hand + eye + head | Basic | Hand + head | None | Inside-out 6DoF |
| **Hand Tracking** | Yes | Yes | Yes | No | Yes |
| **Eye Tracking** | Yes | No | No | No | No |
| **Authenticatio n** | Optic-ID (iris) | — | — | — | — |
| **Audio** | Spatial Audio | On-ear speakers | Spatial audio | Stereo | 3D spatial |
| **Battery** | 2.5 h (external) | Unknown | Concept | None | 2–3 h |
| **Charging** | USB-C | — | — | USB-C | USB-C 3.2 |
| **Connectivity** | Wi-Fi 6, BT 5.3 | BT, Wi-Fi | Wireless low-latency | USB-C | Wi-Fi 6E, BT |
| **Input Methods** | Eye/hand/voic e | Touch | Hand/voice | Buttons | Controllers + hand |
| **IPD Range** | 51–75 mm | — | — | — | — |
| **Use Cases** | Full MR apps | Smart overlay | AR apps | Virtual screen | VR/MR gaming |

| | | | | | |
|---|---|---|---|---|---|
| **Special Features** | Spatial video, OpticID | High portability | Wireless AR | Cinema display | Full-color passthrough |

COMPLETE TABLE **2.1**

| **Category** | **Magic Leap 2** | **HoloLens 2** | **Meta Ray-Ban Glasses** | **Lenovo ThinkReality A3** | **VIVE XR Elite** | **Amazon Echo Frames (3rd Gen)** | **Amazon Amelia Smart Glasses (Best-Estimate)** | **Smart Glasses** |
|---|---|---|---|---|---|---|---|---|
| **Device Type** | Enterprise AR Headset | Enterprise MR Headset | Smart Glasses | AR Tethered Glasses | XR Headset | Smart audio glasses (no AR display) | Enterprise AR smart glasses for delivery workers | Smart glasses (AR + sensor-rich wearable) |
| **Price** | $3,600–$5,500 | ~$3,500 | $299–$379 | $1,499 | $1,099 | ~$269 | Not announced (expected enterprise-rate ~$1,200–$1,800) | Estimated $200 (without EoS) |
| **Release Year** | 2022 | 2019 | 2023 | 2021 | 2023 | 2023 (3rd Gen) | 2025 | 2026 |
| **Weight** | 260 g (headset) | 566 g | 48–50 g | 130 g | Modular, lightweight | ~31–36 g depending on lens type | 70–110 g EST. (glasses only; battery offloaded to vest) | 70–120 g |
| **OS / Platform** | Magic Leap OS | Windows Holographic | Meta firmware | Custom Android | Android XR | Amazon custom firmware | Custom Amazon OS (Android-based, estimated) | Linux (Raspberry Pi OS / other Linux distro) |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **CPU / Chipset** | AMD Zen 2 + RDNA2 | Snapdragon 850 + HPU 2.0 | Qualcomm mobile SoC | PC/Phone CPU | Snapdragon XR2 | Low-power embedded audio chipset | Snapdragon XR1 / XR2-class (estimated) | Processor Server-based (PC offload) |
| **GPU** | RDNA2 | Adreno + HPU | Integrated | PC/phone GPU | Adreno 650 | None | Integrated Adreno GPU EST | Server-based (PC GPU) |
| **RAM** | 16GB | 4GB | — | PC RAM | 12GB | Not specified (low-power embedded RAM) | 2–4 GB EST. | 8 GB onboard + PC RAM |
| **Storage** | 256GB NVMe | 64GB | — | PC storage | 128GB | Not applicable | 32–64 GB onboard EST. | 128 GB onboard + PC storage |
| **Display Type** | Waveguide | Holographic waveguide | None | Micro-OLED | LCD + pancake | (no AR / no HUD) | Monocular or binocular micro-LED / micro-OLED HUD EST | TBD |
| **Resolution (per eye)** | 1440×1760 | 2048×1080 | — | 1920×1080 | 1920×1920 | — | 640×400 – 1280×720 | TBD |
| **Refresh Rate** | 120 Hz | 60 Hz | — | 60 Hz | 90 Hz | __ | | |
| **Field of View** | 70° | 52° | — | ~40° | 110° | __ | 10–20° EST | TBD |
| **Brightness** | 20–2000 nits | ~500 nits | — | 200–400 nits | — | __ | 1000+ nits (for outdoor delivery use) EST | TBD |
| **Sensors** | 3 cams + depth + eye | Depth + IR eye tracking | IMU, touch, mics | IMU, tracking cams | Depth + RGB + IMU | Accelerometer | IMU (accelerometer + gyro) | IMU-Eye tracking-Stereo 3D-Touc |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | h-Microphone-Camera-Speaker-Bluetooth radio |
| **Cameras** | 12.6MP RGB + depth | RGB + depth | 12MP | 8MP RGB | 16MP RGB | None | based CV hazard detection | |
| **Tracking** | SLAM + eye tracking | 6DoF + hand + eye | None | 6DoF | Inside-out 6DoF | Basic motion tracking (IMU only) | Type No full 6DoF; uses basic IMU + visual tracking for alignment EST. | TBD |
| **Hand Tracking** | Yes | Yes | No | Yes | Yes | No | None | Yes |
| **Eye Tracking** | Yes | Yes | No | No | No | No | None | Yes |
| **Authentication** | Iris ID | — | LED indicator only | — | — | — | | |
| **Audio** | Spatial | Spatial | Open-ear speakers | Stereo | Integrated speakers | Open-ear directional speakers (4-micro speaker array) | Small open-ear speakers or bone-conduction EST | Unknown |
| **Battery** | 3.5 h | 2–3 h | 3–4 h | Tethered | ~2 h (swap) | ~6 hours continuous playback / full day mixed use | Life 8–12 hours using vest battery pack | Life Unknown |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Charging** | USB-C | Fast charge | Case | USB-C | USB-C | ~2 hours | Time 1–2 hours for vest battery EST | USB-C |
| **Connectivity** | Wi-Fi 6, BT | Wi-Fi 5 | Bluetooth | USB-C tether | Wi-Fi 6E, BT | Proprietary charging cable | Bluetooth + Wi-Fi (tethered to vest/phone/hub) | Wi-Fi, Bluetooth, Wired (PC link) |
| **Input Methods** | Hand/eye/voice | Hand/eye | Voice/touch | Controller optional | Controllers | Voice (Alexa) | Vest-mounted controller + Voice commands | Eye, Hand, Voice, Touch, Buttons |
| **IPD Range** | — | Auto | — | — | 54–73 mm | __ | N/A (fixed HUD, no IPD adjustment) | TBD |
| **Use Cases** | Enterprise AR | Enterprise MR | Camera + audio | Enterprise workflows | XR gaming + MR | | Delivery routing | Camera + Audio + Controller + AR |
| **Special Features** | Dynamic dimming, SLAM | Holo UI | Social capture | PC-class AR | Hot-swap battery | Auto-off when removed | Hazard detection, Privacy mode | Hot-swap battery |

This is a comprehensive comparison table detailing the specifications and features of numerous Augmented Reality (AR), Mixed Reality (MR), and smart glasses devices on the market. The table is divided into two main sections for clarity.

**Section 1: Consumer & Prosumer Devices**

This section focuses on devices aimed at general consumers, tech enthusiasts, and professional creators.

- Apple Vision Pro: Positioned as a premium, high-performance Mixed Reality Headset. It boasts top-tier specs like a powerful M5 chip, ultra-high-resolution

Micro-OLED displays, advanced hand and eye tracking (Optic ID), and spatial audio, but at a very high price and with significant weight.

- Oppo Air Glass 3 & Xiaomi Wireless AR: Represent lightweight AR Glasses concepts. They prioritize portability and basic information overlays, often relying on a connected smartphone for processing (phone-based CPU/GPU).

- Huawei Vision Glass: Functions primarily as a portable virtual screen (Smart Display Glasses) for media consumption, lacking interactive tracking features.

- Meta Quest 3: A versatile Mixed Reality Headset focused on accessible VR/MR gaming and applications, offering strong performance with inside-out tracking and full-color passthrough at a mid-range price.

**Section 2: Enterprise & Specialized Devices**

This section highlights devices designed for professional, industrial, and specific commercial use cases.

- Magic Leap 2 & Microsoft HoloLens 2: High-end Enterprise AR/MR Headsets. They feature robust construction, secure operating systems (Magic Leap OS, Windows Holographic), professional-grade SLAM and hand/eye tracking, and are built for complex industrial design, training, and field service workflows.

- Meta Ray-Ban Glasses & Amazon Echo Frames: These are Smart Audio Glasses with no AR display. Their primary functions are hands-free audio, photography/video capture (Meta), and voice-assistant interaction (Alexa), emphasizing all-day comfort and style.

- Lenovo ThinkReality A3 & VIVE XR Elite: These devices serve professional and prosumer XR needs. The Lenovo model is often tethered to a PC for enterprise applications, while the VIVE offers a balance of gaming and business features with modular design.

- Amazon Amelia Smart Glasses (Estimated): A concept for enterprise logistics (e.g., delivery workers). Key estimated features include long battery life via a vest pack, a monocular HUD for navigation and hazard detection, and durability for outdoor use.

- Smart Glasses (Cerebro): is a modular, sensor-rich prototype using a hybrid processing model (onboard + PC offload). It aims to be a versatile platform supporting multiple input methods (eye, hand, voice) for development and research.

Overall Summary:

The table illustrates the clear segmentation in the market:

- Headsets (Vision Pro, Quest 3, Magic Leap 2, HoloLens): Offer full immersive experiences with high computing power but are bulkier and often higher-priced.
- True AR Glasses (Oppo, Xiaomi concepts): Aim for everyday wearability, sacrificing processing power and display immersion for comfort.
- Audio/Smart Glasses (Meta Ray-Ban, Echo Frames): Forgo visual AR entirely to focus on discrete audio and camera functions.
- Enterprise Solutions: Prioritize specific professional needs like tracking accuracy, ruggedness, long battery life, and secure software over consumer-friendly design and price.

---

Below are the sub-sections that must appear inside Related Work because they directly support the technology foundation.

2.4 **Computer Vision in Wearable Systems**

Computer vision has been used in mobile and robotics systems, but rarely in smart glasses. Using **YOLO11**, the proposed solution provides:

- Feature extraction
- Object detection
- Identifying obstacles
- Enhancing indoor navigation

This improves:

- Safety
- Context awareness
- Accessibility for people of determination

**2.5 Indoor Navigation Using Graph-Based Mapping**

Traditional GPS fails indoors due to:

- No satellite visibility
- Weak signal penetration
- Multipath interference

Existing research shows that indoor navigation requires:

- Graph-based modeling
- Manual map digitization
- JSON-based routing
- Customized building layouts

Our system maps **Building 2 (Computer Science Building)** manually, converts rooms, stairs, and elevators into nodes and edges, and uses it for indoor navigation.

**2.6 Hardware in Smart Glasses**

Existing systems use expensive hardware. By contrast, ESP32 offers:

- Low cost

- WiFi/Bluetooth
- Compact size
- Easy integration with glasses
- Real-time communication

Microphone + speakers allow:

- Real-time speech capture
- Playback of TTS responses
- Audio feedback for navigation

**2.7 Speech Recognition (Whisper)** [14] **in Wearables**

Whisper is used because:

- It supports Arabic and English
- High accuracy
- Can transcribe audio/video
- Converts speech to .txt file
- Works offline or online

This enables:

- Hands-free interaction
- Fast processing
- Reliable commands for LLM

**2.8 Text-to-Speech Systems**

TTS research shows bilingual support improves accessibility. Our system uses two voices:

- Arabic TTS
- English TTS

Smart Glasses decide the output voice depending on LLM language.

**2.9 Large Language Models**

LLMs like GPT and LLaMA transform commands into structured actions. They provide:

- Context-aware reasoning
- Intelligent decision making
- Multi-step interpretation

This is enhanced by the **MCP NLP pipeline** [1], [5], [6], [8], which performs:

- Intent classification
- Task decomposition
- Action-to-JSON transformation
- Model execution routing

**2.10 Smart Home Integration**

Existing smart homes lack wearable integration.

Our system introduces:

- Mobile app dashboard
- Web dashboard
- Device management
- Controllers
- Logs

- Statistics
- AR/VR interaction

This unifies all control methods.

**2.11 User Flow and Multimodal Interaction**

Compared to existing products, our system includes:

- Natural speech → text → commands
- Vision-based alerts
- AR navigation
- App dashboards
- Web dashboards

This creates a consistent and intuitive experience.

# 3. METHODOLOGY

## 3.1. Requirement Analysis

### 3.1.1 Textual Requirements

## Functional Requirements

1. **Real-time Object Detection**
   - The system must detect objects using a custom-trained YOLO model.
   - The model should identify indoor landmarks (doors, elevators, room numbers, stairs, etc.).
2. **Indoor Navigation**
   - The system must compute optimal paths inside Building 2 using a custom indoor map.
   - The system should provide turn-by-turn navigation visually in AR and via speech.
3. **Multimodal Speech Interaction**
   - The glasses must capture audio and run Whisper [14] for speech-to-text.
   - The extracted text should be processed by an LLM to provide intelligent responses.
4. **Smart Home & IoT Control**
   - Users must be able to control home devices (lights, AC, door lock) through voice.
   - The ESP32 microcontroller must receive commands over Wi-Fi and trigger actuators.
5. **Multilingual Communication**
   - The system must transcribe, translate, and generate speech output in real time.
6. **Mixed Reality Rendering**

- The AR module must render virtual arrows, labels, and icons anchored to detected objects.

7. **User-Friendly Interaction**
   - Users can interact with the system through voice, gestures, or UI element

## Non-Functional Requirements

1. **Performance**
   - Object detection must run at ≥ 20 FPS.
   - Speech recognition latency must be ≤ 2 seconds.
2. **Accuracy**
   - YOLO model accuracy must exceed 90% mAP.
   - Whisper transcription accuracy must exceed 95% for English and Arabic.
3. **Usability**
   - The interface must be simple for visually impaired and hearing-impaired users.
4. **Reliability**
   - The system should maintain stable operation for long sessions.
5. **Security**
   - Device commands must be encrypted before being sent to the ESP32 module.
6. **Portability**
   - The system should run on mobile devices and AR glasses.

### 3.1.2 Use Case Diagram

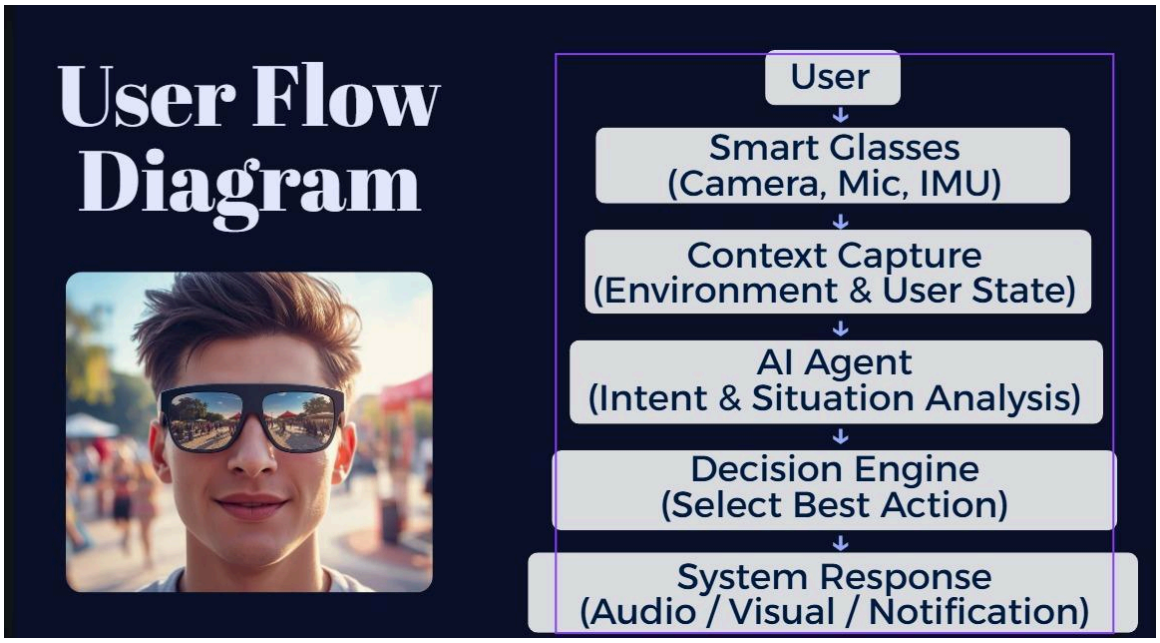



Figure 3.1: Use Case Diagram

### 3.1.3 Use Case Scenarios

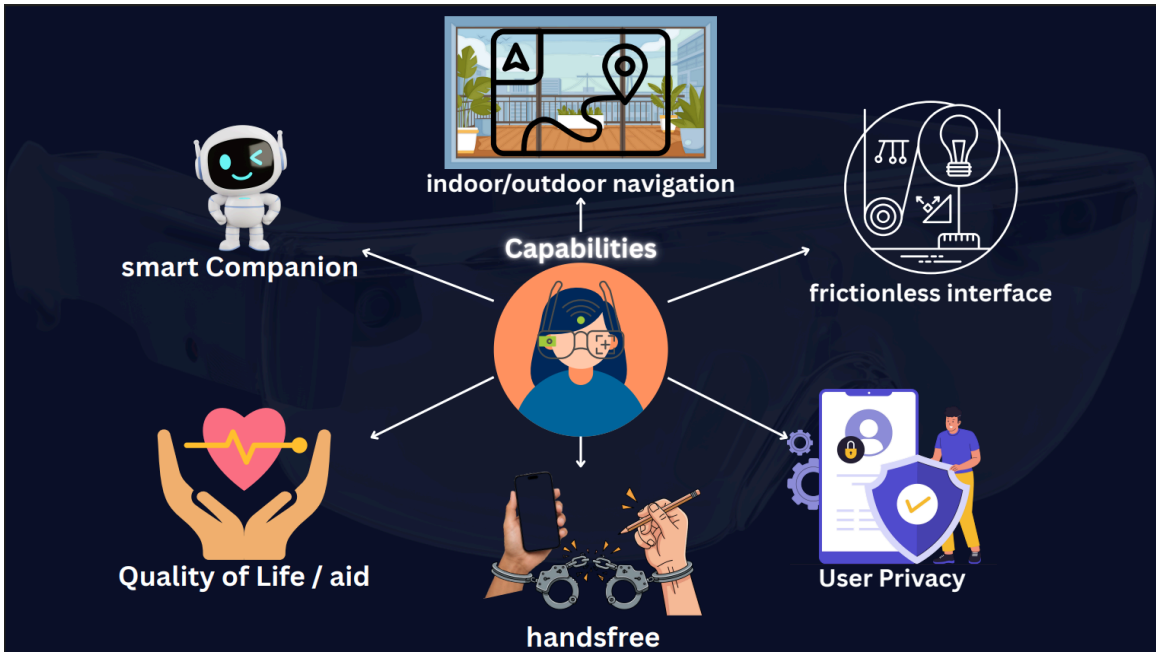



Figure 3.2: Use Case Scenarios

**UC1 – Real-Time Object Detection**

| Step | Description |
|---|---|
| 1 | User starts camera mode. |
| 2 | YOLO processes the current frame. |
| 3 | Model returns detected objects + bounding boxes. |
| 4 | AR module displays labels and icons. |

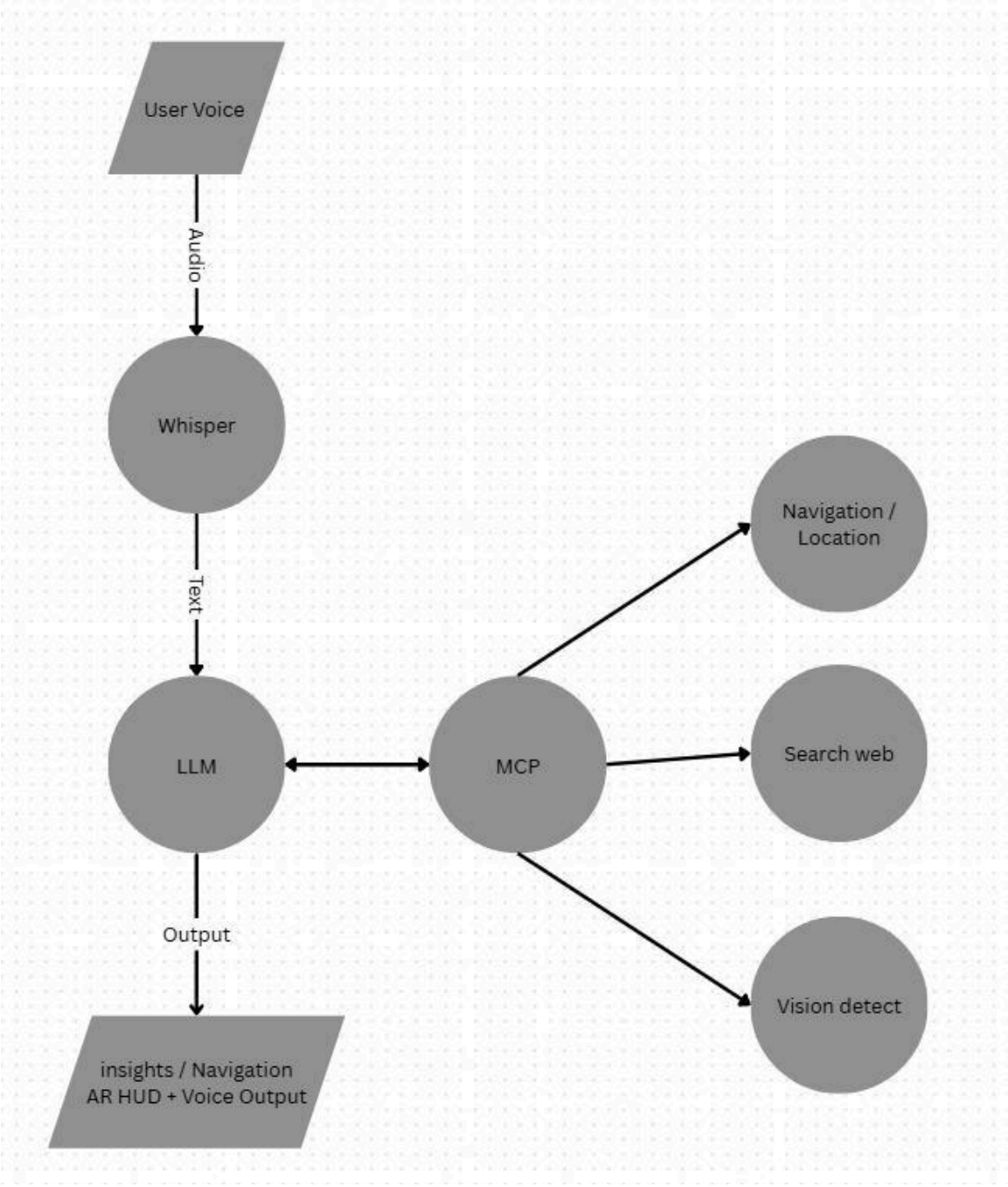
**UC2 – Indoor Navigation**

| Step | Description |
|---|---|
| 1 | User gives voice command: “Guide me to Room 215.” |
| 2 | Whisper → converts speech to text. |
| 3 | LLM → extracts destination + context. |
| 4 | Navigation engine computes shortest route. |
| 5 | AR overlay provides arrows + callouts. |

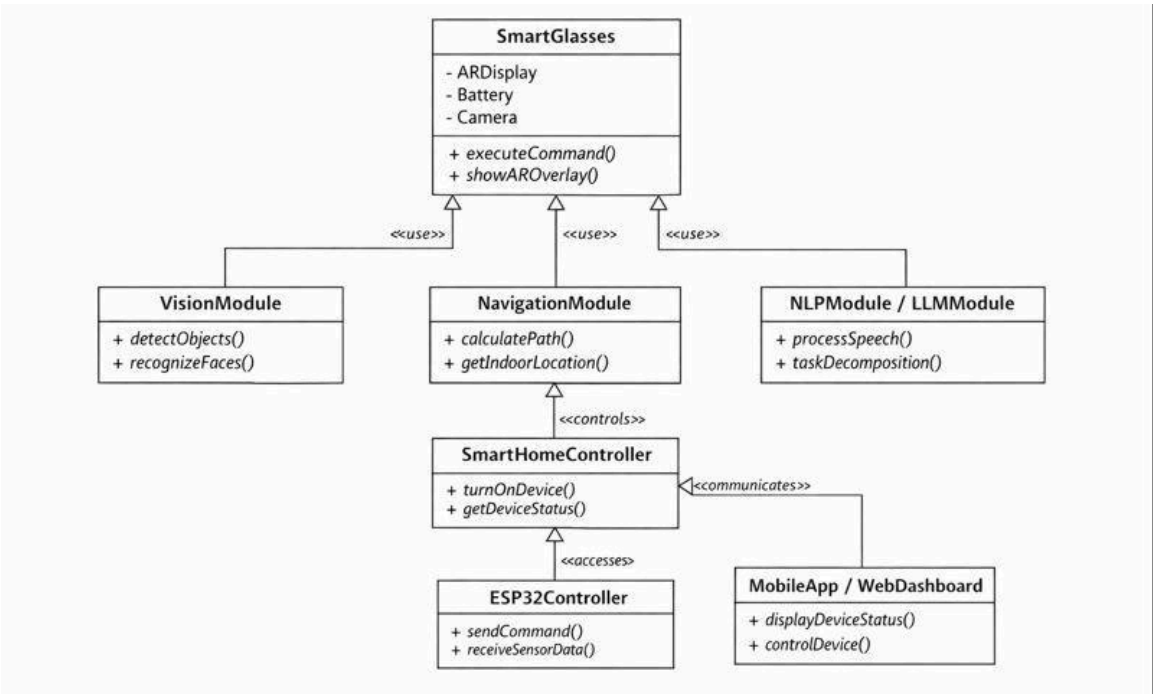
**UC3 – Smart Home Control**

| Step | Description |
|---|---|
| 1 | User says: “Turn on the lights.” |
| 2 | STT module transcribes the request. |
| 3 | LLM interprets command and creates a JSON instruction. |
| 4 | System sends command to ESP32. |
| 5 | Device activates the actuator. |

### 3.2. Design




3.2.1. Activity Diagram




3.2.2. Class Diagram

The diagram presents the main components of the Smart Glasses system and how they interact. The SmartGlasses class represents the device core, connecting with modules responsible for vision (VisionModule), navigation (NavigationModule), and language processing (NLPModule / LLMModule). The SmartHomeController oversees communication with IoT devices through the ESP32Controller, while the MobileApp / WebDashboard allows users to monitor and control the system. Each class is defined with its key attributes and methods, and the arrows indicate the relationships and interactions among the modules.

### 3.2.4. Deployment Diagram

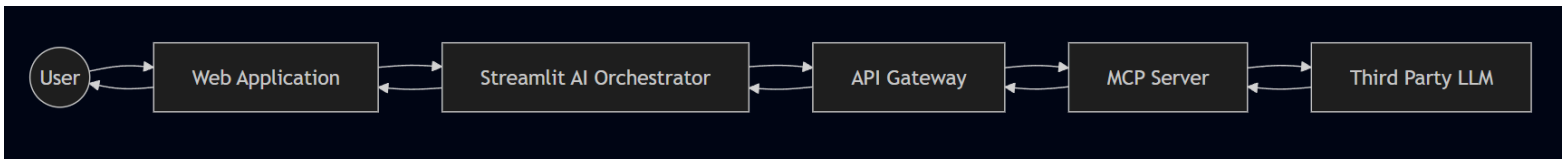



Figure 3.2.4. Deployment Diagram

### 3.3. Implementation

Main Technologies Used

- YOLOv11 for object detection
- OpenCV for camera streaming
- Whisper for speech recognition
- LLM API (OpenAI / local model)
- Unity / ARCore / ARKit for mixed reality rendering
- Python + Node.js backend
- ESP32 with MicroPython for IoT devices

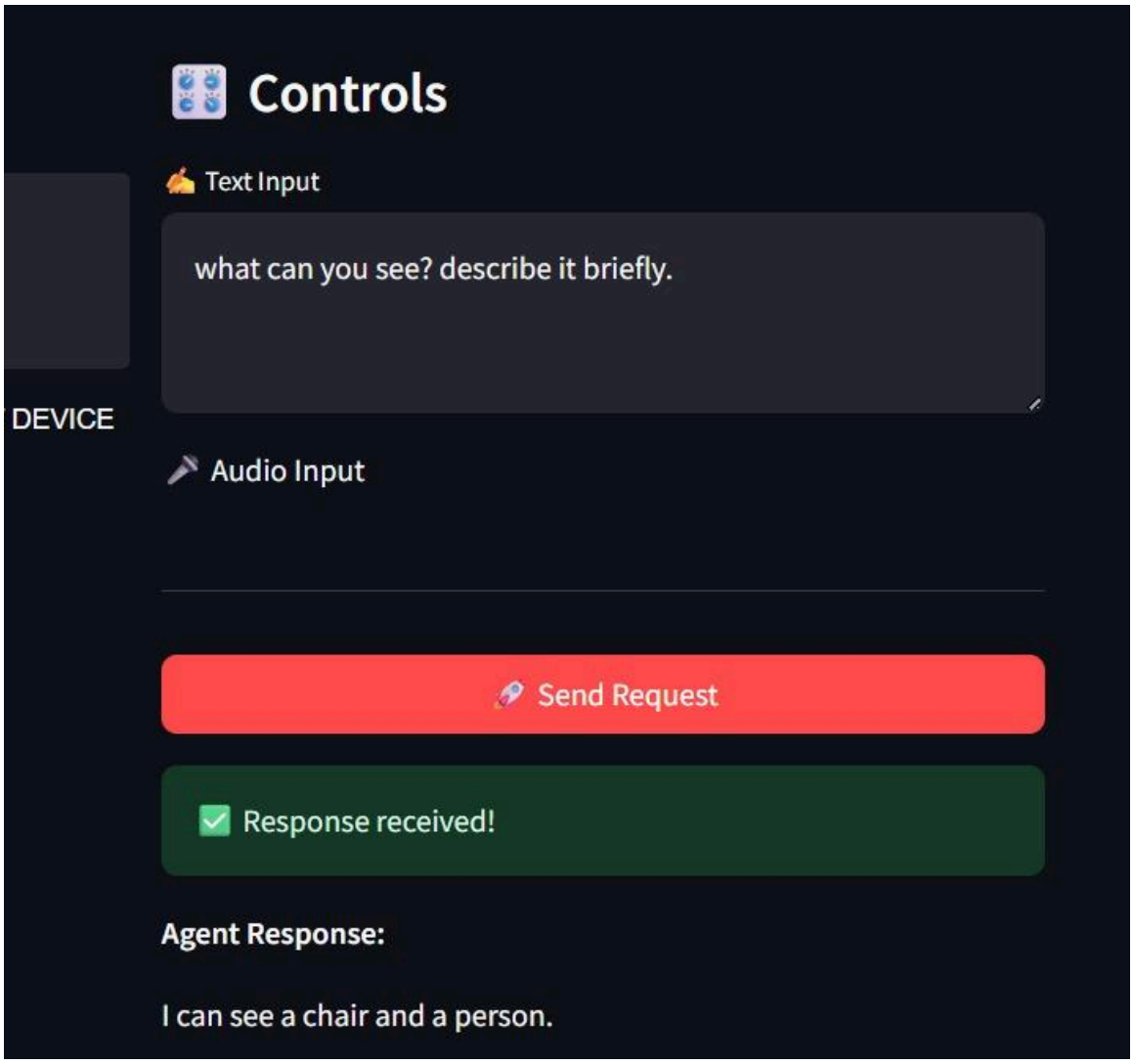



Figure 3.4.1: MCP Testing Web interface

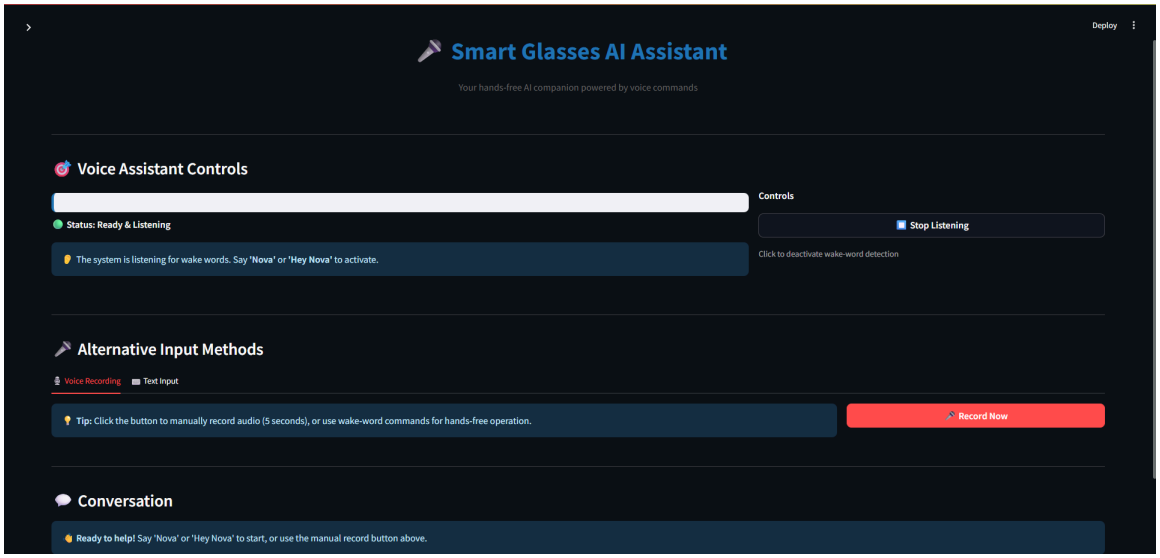



Figure 3.4.2: MCP Testing Web interface

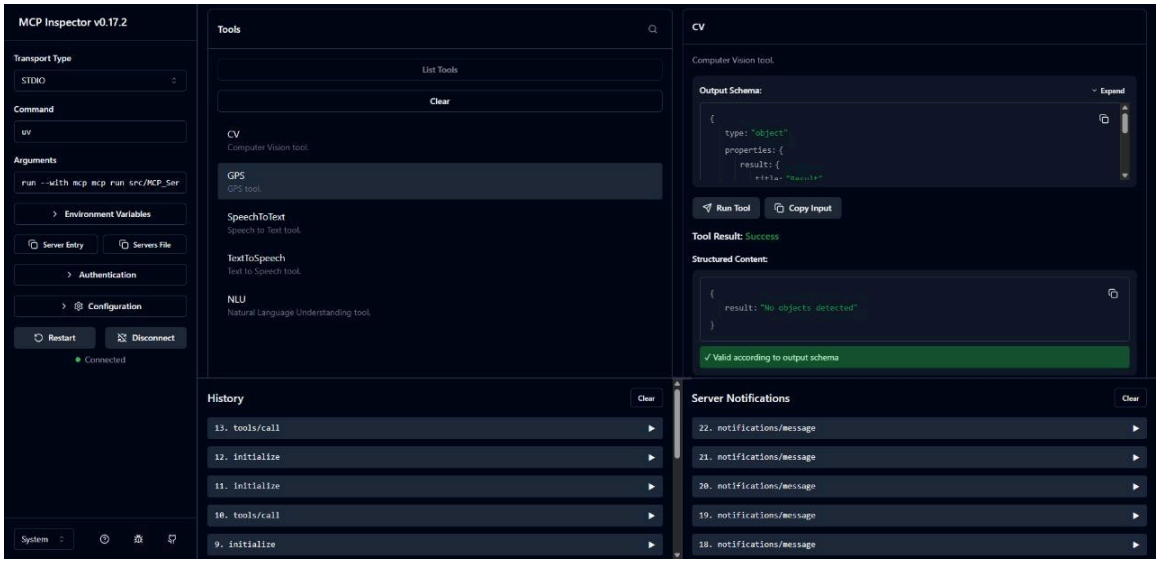



Figure 3.4.3: MCP Testing Web interface

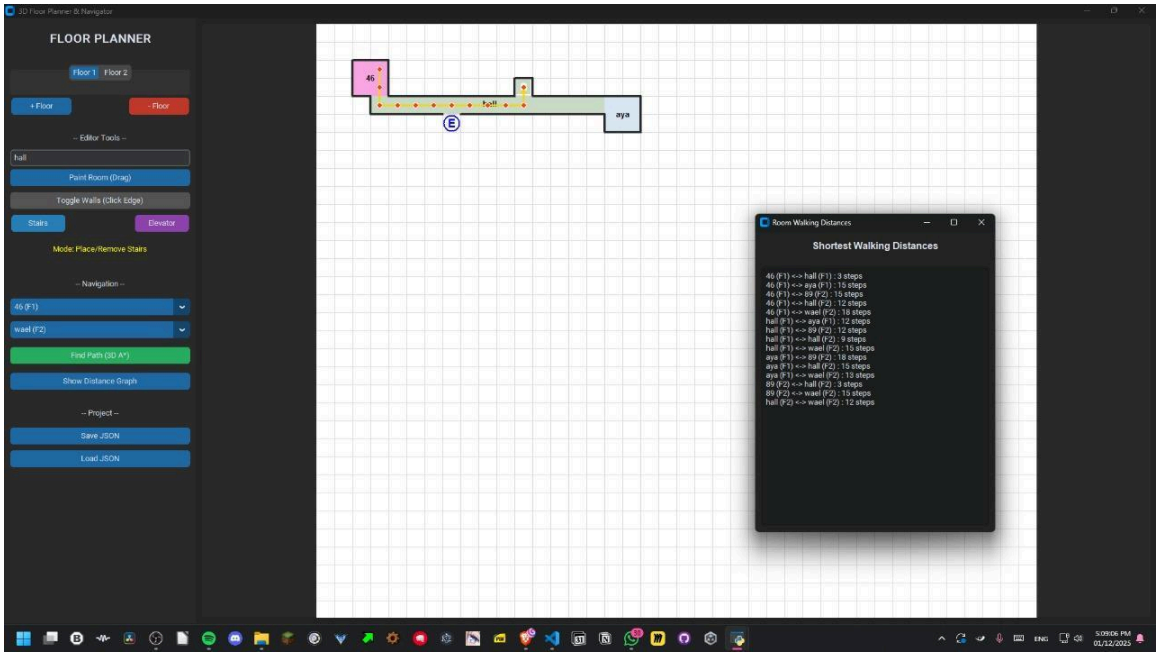



Figure 3.4.4: MCP Testing Web interface

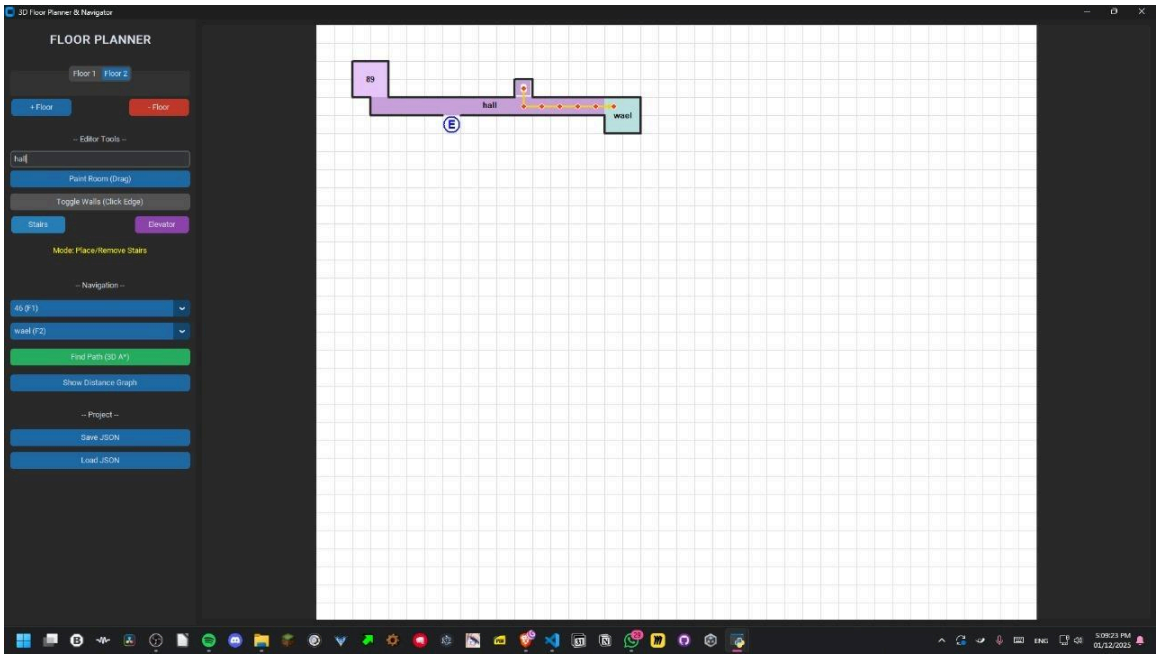


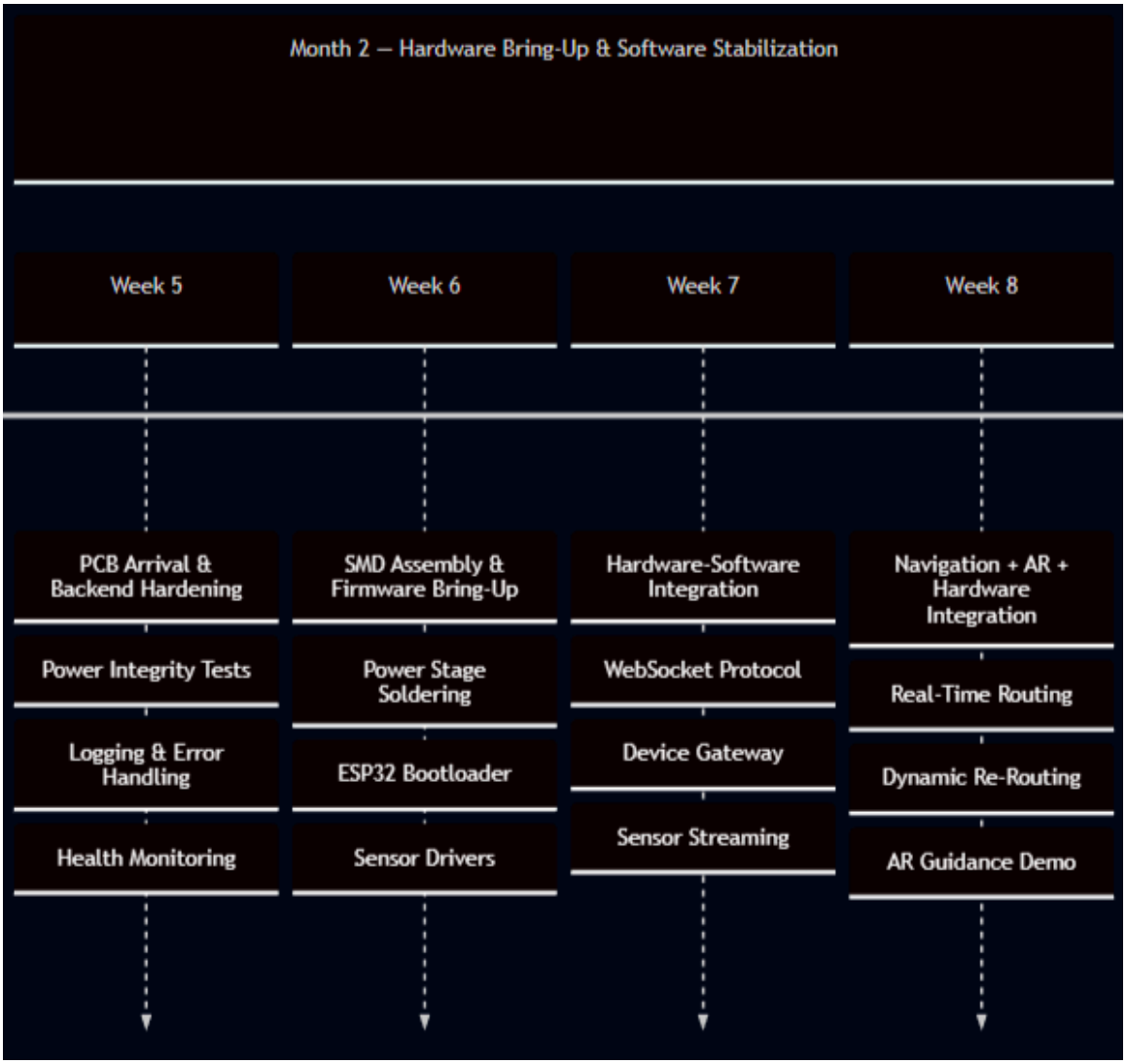

Figure 3.4.5: MCP Testing Web interface

### 3.1. Overview of the Dataset/Model

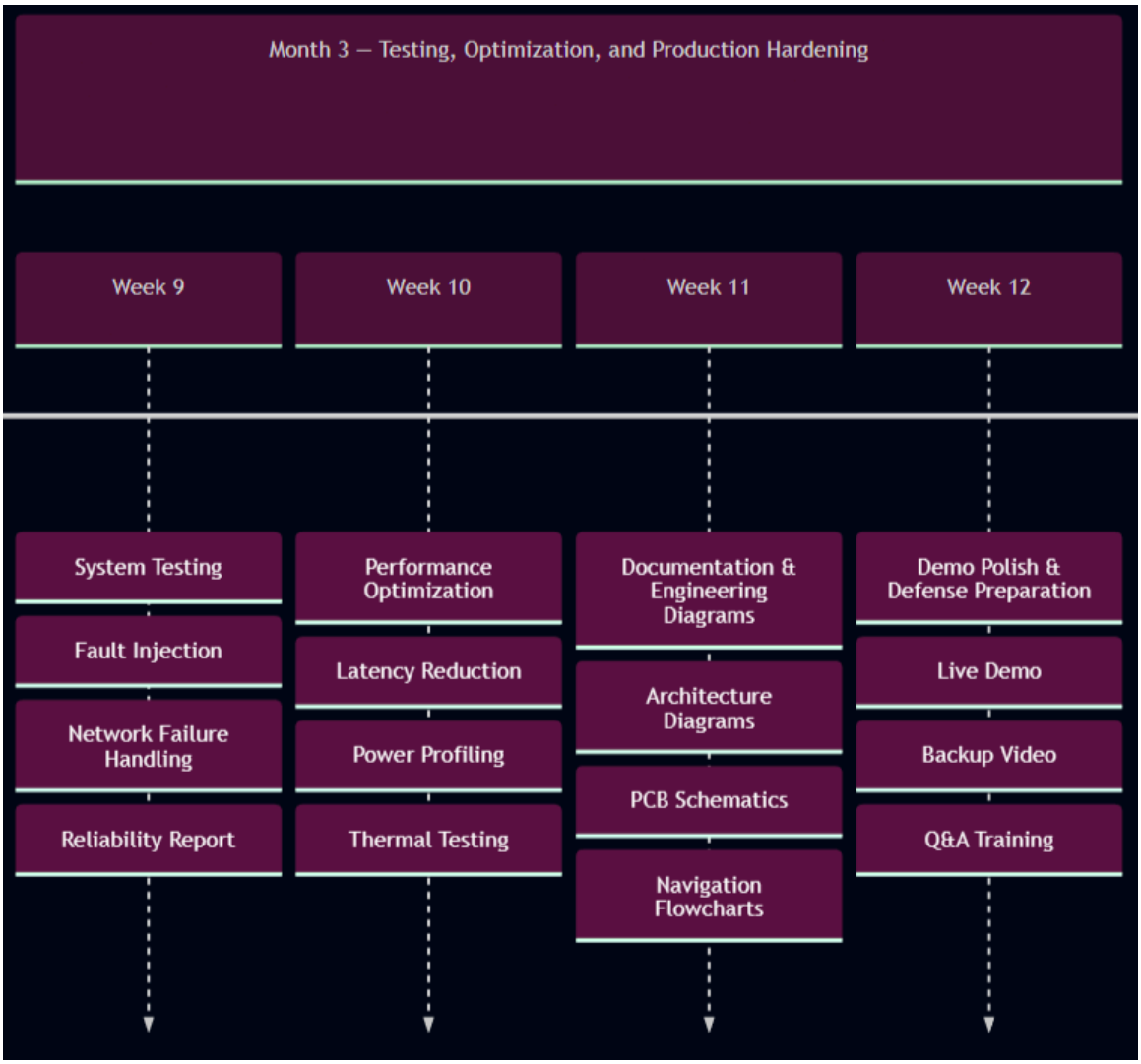
### 3.2. Tools and Technology

### 3.3. Proposed Approach


Cerebro Smart Glasses - 3 Month Engineering Timeline




Cerebro Smart Glasses - 3 Month Engineering Timeline




Cerebro Smart Glasses - 3 Month Engineering Timeline

- For System Design Projects;

### 3.1. Design Overview

### 3.2. System Architecture

The system follows a **modular architecture** consisting of six main modules:

1. **Vision Module (YOLO11 [7] + Custom Dataset)**
2. **Indoor Navigation Module (GPS + JSON-based Floor Map)**
3. **NLP Pipeline (Whisper + LLM + Task Breakdown)**
4. **Mixed Reality Interface (AR Overlay + VR Simulation)**
5. **Smart Home Control Module**

6. **Hardware Integration Layer (ESP32 + Sensors + Battery Unit)**

Each module is loosely coupled and communicates through a central processing unit running the LLM and control logic.

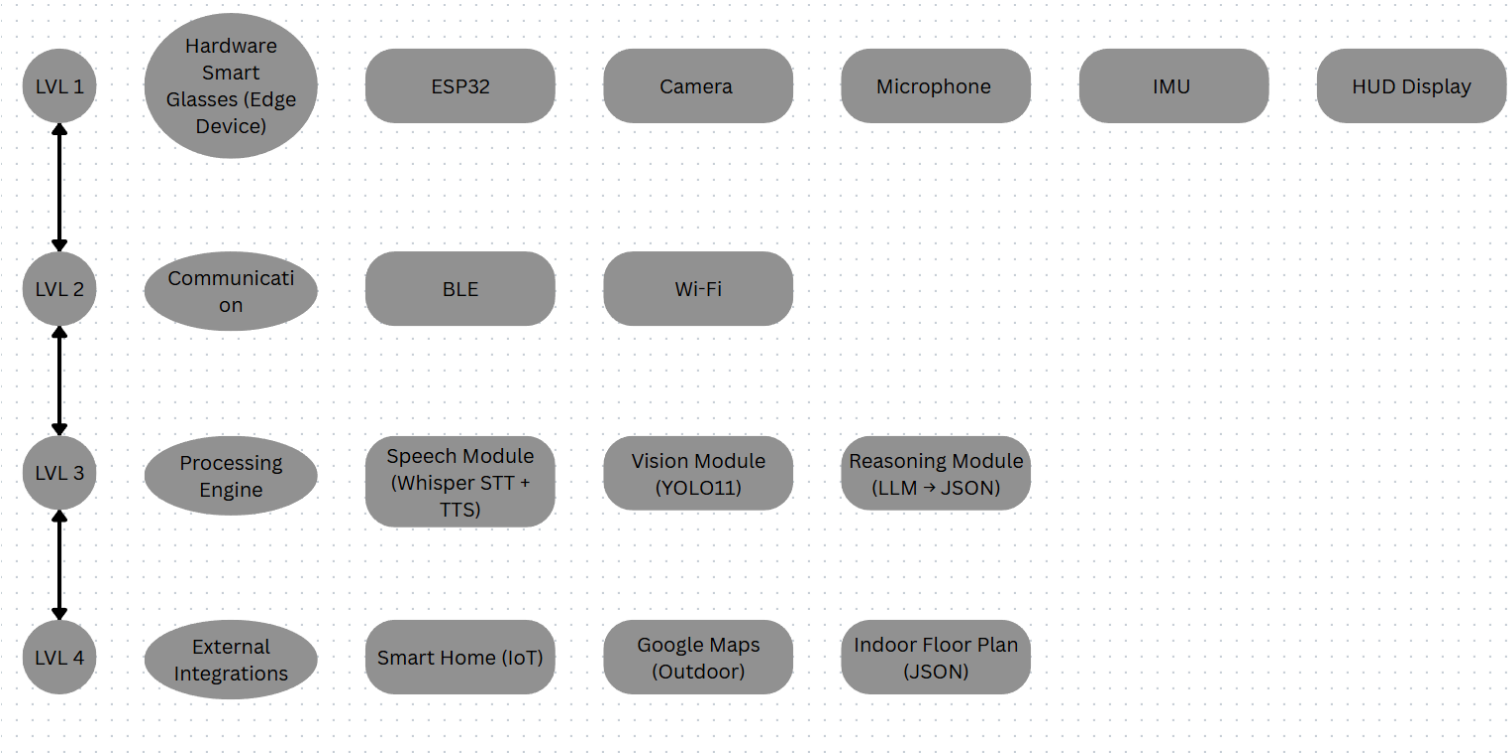



Figure 3.2.4: System Architecture

3.2.5. Hardware Architecture Diagram

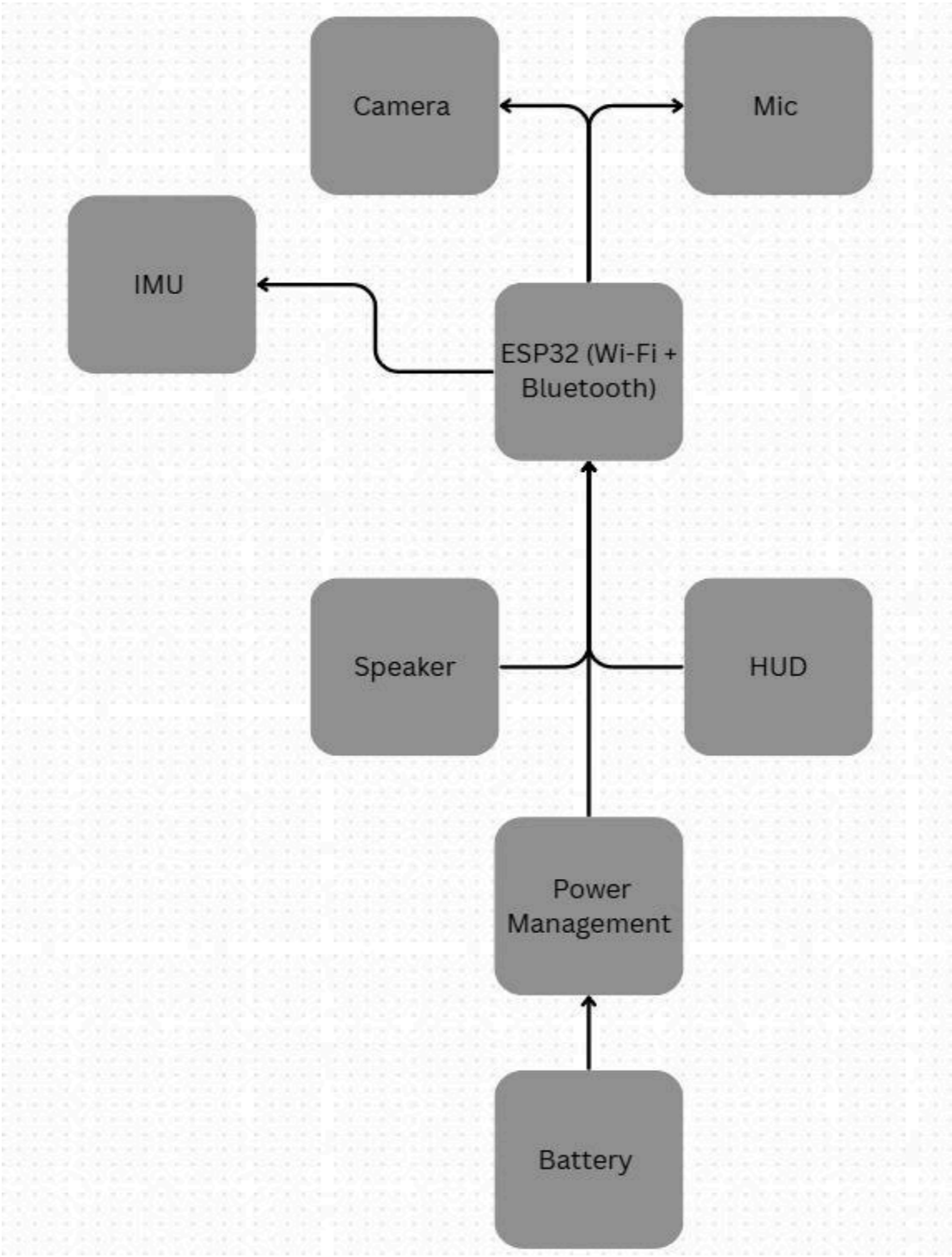



Figure 3.2.3. Hardware Architecture Diagram

### 3.3 Data Collection and Preparation

1. **Vision Dataset Collection**
   - Dataset: custom images captured inside the Computer Science building.
   - Classes: doors, stairs, elevators, signs, obstacles.
   - Annotation Tool: Roboflow.
   - Preprocessing: resizing, normalization, augmentation.
2. **Floor Map Data for Indoor Navigation**
   - The building map was manually drawn.
   - Converted into **JSON format** where each room, corridor, node, and path is represented by IDs.
   - Applied graph representation for navigation (nodes, edges, weights).
3. **Speech Dataset**
   - Whisper handles multilingual transcription.
   - No manual data collection required.

### 3.4 Computer Vision Module

#### YOLO11 Model Training

We used YOLO11 [7] due to its high speed, light weight, and strong accuracy in real-time environments.

**Steps:**

1. Annotate images.
2. Split into train/val/test.
3. Train using transfer learning.
4. Validate accuracy (mAP, precision, recall).

#### Why We Chose YOLO11

- Outperforms YOLOv8 in speed.
- Lightweight enough to run on portable devices.
- Excellent for object detection in dynamic environments.

### 3.5 Indoor Navigation Module

#### Creating the Navigation Graph

- Each corridor, hallway, and room is converted into nodes.

- Connections between nodes form edges with distances as weights.
- The result: a navigable graph.

Pathfinding Algorithm

- We apply A* because:
- It is faster than Dijkstra.
- Provides optimal paths.
- Works well for indoor map constraints.

### 3.6 NLP + LLM Reasoning Module

The NLP pipeline handles user speech, understands intent, breaks tasks into steps, and triggers actions.

**Pipeline Flow**

1. Whisper converts speech → text.
2. LLM analyzes the text.
3. Task Breakdown Engine (MCP) converts the sentence into sub-steps.
4. Action Manager decides what to trigger (navigation, smart home, CV, AR overlay).
5. Output JSON defines system actions.

**Why LLM?**

- Understands natural language.
- Can perform reasoning.
- Handles complex commands like:

"Guide me to the AI lab and tell me if there are stairs on the way."

### 3.7 Mixed Reality (AR/VR Module)

**AR Overlay**

- Arrows are shown on the lenses.

- Based on navigation output.
- Highlights detected objects with bounding boxes.

VR Mode

Used for testing and for users with disabilities to simulate navigation.

### 3.8 Smart Home Automation Module

The smart glasses communicate with a Flask/FASTAPI backend.

**Supported Features:**

- Turn lights on/off
- Open/close smart door
- Adjust temperature
- Control appliances

### 3.9 Hardware Design and Integration

**Components Used:**

- ESP32 microcontroller
- speakers
- Dual microphones
- Ultra‑light camera
- Rechargeable battery pack
- Bluetooth/WiFi module

**Why This Hardware**

- Low power consumption
- Lightweight
- Supports real‑time streaming
- Affordable for a graduation project

### 3.10 System Workflow Summary

1. User speaks command.
2. Whisper transcribes it.
3. LLM interprets and classifies the task.
4. CV module or Navigation module activates.
5. Output is displayed through AR overlay.

Smart home actions executed if needed.

### 3.11 Implementation Steps (Detailed)

1. **Collect Datasets (Vision + Map)**

   Identify the types of data required for the project: images for computer vision and maps for indoor navigation.

   For vision datasets: collect images of objects, environments, or specific locations relevant to the use case. Include diverse conditions such as lighting variations, occlusions, and multiple angles to ensure model generalization.

   For navigation datasets: obtain floor plans of buildings, including corridors, stairs, elevators, entrances, and exit points. If indoor GPS is not available, use manual measurements or mapping tools.

   Ensure the dataset is comprehensive and representative to avoid bias in model performance.

2. **Prepare and Annotate Images**

   Clean and preprocess all collected images by resizing, normalizing, and correcting orientations.

   Annotate objects using professional labeling tools (e.g., LabelImg, CVAT). Create bounding boxes, masks, or keypoints depending on the YOLO model requirements.

   Maintain annotation consistency and define a clear labeling convention to reduce errors.

   Split the dataset into **training (70%)**, **validation (15%)**, and **test (15%)** sets to ensure accurate model evaluation.

3. **Train YOLOv11 Model**

   Configure YOLOv11 architecture: define number of classes, anchor boxes, input dimensions, and backbone network.

Prepare the data pipeline to feed annotated images into the model efficiently, including data augmentation (rotation, scaling, flipping) to improve robustness.
Train the model on a GPU-enabled environment to speed up learning. Monitor training loss, precision, recall, and mAP (mean Average Precision) metrics.
Perform hyperparameter tuning such as learning rate, batch size, and epochs for optimal performance.
Test the trained model on the test set to evaluate generalization and fine-tune if necessary.

4. **Build Navigation Graph + JSON Map**
   Convert building floor plans into a **graph structure** where nodes represent key locations (rooms, intersections) and edges represent paths or connections.
   Include details such as stairs, elevators, and corridors. Assign weights to edges to indicate distance or difficulty.
   Encode the graph in **JSON format**, storing all nodes, edges, and metadata to allow the system to parse and compute navigation routes programmatically.
   Implement pathfinding algorithms (e.g., Dijkstra or A*) to calculate the shortest and most efficient path to the target destination.
5. **Develop Whisper + LLM Pipeline**
   Integrate the **Whisper speech-to-text engine** to process audio inputs from the user.
   Detect the spoken language (Arabic or English) automatically and transcribe the speech accurately.
   Send the transcribed text to the **Large Language Model (LLM)** for semantic understanding and task planning.
   Translate user commands into structured steps in **JSON format** for downstream execution by different modules (navigation, AR, smart home control).
   Ensure error handling for misheard commands, ambiguous instructions, or unsupported requests.
6. **Implement AR Rendering**
   Develop the AR engine to overlay digital objects, navigation indicators, and notifications onto the real-world view captured by the glasses.
   Integrate outputs from YOLO, GPS/IPS navigation, and environmental sensors to render accurate, real-time information.
   Optimize frame rate and latency for smooth visualization in the Heads-Up Display (HUD).
   Test AR overlays in multiple lighting and environmental conditions to ensure stability and reliability.
7. **Integrate ESP32 Hardware**
   Connect microphone, speaker, sensors, and other peripherals to the **ESP32 microcontroller**.

Establish wireless communication via **Wi-Fi or Bluetooth** between ESP32 and Smart Glasses for real-time data transfer.
Implement firmware to handle audio capture, playback, sensor reading, and command execution.
Ensure the hardware responds instantly to commands from both the user and the software modules.

8. **Test Modules Individually**
   **Vision Module:** Evaluate object detection accuracy and robustness using test images.
   **Navigation Module:** Verify route calculation and indoor/outdoor navigation correctness.
   **Whisper + LLM Pipeline:** Check transcription and command interpretation accuracy.
   **AR Rendering Module:** Test overlay placement, HUD clarity, and AR responsiveness.
   **Hardware Module:** Confirm microphone input, speaker output, and sensor readings are accurate.
   Record results and debug any issues before integrating modules.
9. **Combine Modules for Full System**
   Integrate all modules into a single workflow to ensure seamless interaction between hardware and software.
   Test communication between modules: commands from LLM should trigger navigation, AR, smart home actions, or media capture correctly.
   Implement synchronization mechanisms and handle concurrency issues to prevent system crashes.
   Perform end-to-end testing to ensure system stability and reliability under different use scenarios.
10. **Evaluate Performance**
    Define performance metrics:
    - YOLO detection accuracy
    - Speech recognition accuracy
    - Navigation precision
    - AR overlay responsiveness
    - Smart home command execution latency

    Conduct real-world testing in multiple environments.
    Identify bottlenecks and optimize system components where necessary.
    Document results with screenshots, logs, and statistics to support evaluation.

### 3.3. System Software

## REFERENCES


[1] Luo, Z., Shi, X., Lin, X., & Gao, J. (2025, April 15). Evaluation Report on MCP Servers. arXiv.

[2] Huang, B.-C., Hsu, J., Chu, E. T.-H., & Wu, H.-M. (2020). ARBIN: Augmented reality based indoor navigation system. Sensors, 20(20), 5890

[3] Alkady, Y., Rizk, R., Alsekait, D. M., Alluhaidan, A. S., & Abdelminaam, D. S. (2024). SINS_AR: An efficient smart indoor navigation system based on augmented reality. IEEE Access, 12, 109171–109183.

[4] Tepper, O. M., Rudy, H. L., Lefkowitz, A. B., Weimer, K. A., Marks, S. M., Stern, C. S., & Garfein, E. S. (2017). Mixed reality with HoloLens: Where virtual reality meets augmented reality in the operating room. Plastic and Reconstructive Surgery, 140(5), 1066–1070.

[5] Alla, M. (2025). Scalable MCP server-client architecture with FastMCP in microservices. Sarcouncil Journal of Multidisciplinary

[6] Singh, A., Ehtesham, A., Kumar, S., & Talaei Khoei, T. (2025). A survey of the Model Context Protocol (MCP): Standardizing context to enhance large language models (LLMs). arXiv.

[7] He, L.-H., Zhou, Y.-Z., Liu, L., Zhang, Y.-Q., & Ma, J.-H. (2025). Research on the directional bounding box algorithm of YOLO11 in tailings pond identification. Measurement, 253(Part C), 117674.

[8] Mastouri, M., Ksontini, E., & Kessentini, W. (2025). Making REST APIs agent-ready: From OpenAPI to MCP servers for tool-augmented LLMs. arXiv.

[9] Kelly, C., Hu, L., Yang, B., Tian, Y., Yang, D., Yang, C., Huang, Z., Li, Z., Hu, J., & Zou, Y. (2024). VisionGPT: Vision-language understanding agent using generalized multimodal framework. arXiv.

[10] C. Wang, W. Luo, S. Dong, X. Xuan, Z. Li, and L. Ma, "MLLM-Tool: A Multimodal Large Language Model for Tool Agent Learning," in Proceedings of the 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Tucson, AZ, USA, Feb. 26–Mar. 6, 2025

[11] W. Berrios, G. Mittal, T. Thrush, D. Kiela, and A. Singh, "Towards Language Models That Can See: Computer Vision Through the LENS of Natural Language," arXiv preprint arXiv:2306.16410, 2023.

[12] S. P. Samarth and S. B. Mahalingam, "The Gemma Sutras: Fine-Tuning Gemma 3 for Sanskrit Sandhi Splitting," in Proceedings of the 9th Widening NLP Workshop (WiNLP), Association for Computational Linguistics, Suzhou, China, Nov. 2025, pp. 235–241

[13] Kopanov, K., & Atanasova, T. (2025). A comparative pattern analysis of Qwen 2.5 and Gemma 3 text generation. Institute of Information and Communication Technologies, Bulgarian Academy of Sciences.

[14] Wang, S., Yang, C.-H., Wu, J., & Zhang, C. (2024). Can Whisper perform speech-based in-context learning? In Proceedings of the 2024 IEEE International Conference on Acoustics, Speech and Signal Processing
[15] Yuan, J. (2024). Performance analysis of deep learning algorithms implemented using PyTorch in image recognition. Procedia Computer Science.
[16] Lei, X., Jiang, X., & Wang, C. (2013). Design and implementation of a real-time video stream analysis system based on FFMPEG. In 2013 Fourth World Congress on Software Engineering. IEEE.
[17] Hassler, D. M., Norbury, J. W., & Reitz, G. (2017). Mars science laboratory radiation assessment detector (MSL/RAD) modeling workshop proceedings. Life Sciences in Space Research, 14, 1–2.
[18] Amazon, "Echo Frames (3rd Gen) Smart Glasses," Available: https://www.amazon.com/Echo-Frames-3rd-Gen-Smart-audio-glasses-with-Alexa-Modern-Rectangle-frames-in-charcoal-gray/dp/B09SVG2M7R
[19] Microsoft, "Microsoft HoloLens 2 Documentation," Available: https://learn.microsoft.com/hololens/
[20] Meta, "Ray-Ban Meta Smart Glasses Official Announcement," Available: https://about.fb.com/news/2023/09/new-ray-ban-meta-smart-glasses/
[21] Google, "Google Maps Indoor (Indoor Navigation)," Available: https://www.google.com/maps/about/partners/indoormaps/
[22] Google, "Google Home," Available: https://home.google.com/welcome/
[23] Amazon, "Alexa Voice Assistant," Available: https://www.amazon.com/alexa-voice-assistant
[24] HERE, "Indoor Positioning Documentation," Available: https://developer.here.com/documentation/indoor-positioning/
[25]Apple, "Apple Home (HomeKit)," Available: https://www.apple.com/ios/home/

# APPENDIX

Include additional content (raw data, code listing, etc.) as necessary to provide a detailed explanation that is not essential in the body of the report but that would be of interest of readers. If this section is not used, remove it from the project template. In case of having multiple appendix sections, informatively title and label as Appendix A, Appendix B, etc., according to the order in which they are mentioned in the text.