**Design Structure Paper: Mobile Application for Visually Impaired Indoor Navigation at LMJ Hospital**

**Title Page**

Project Title: Mobile Application for Visually Impaired Indoor Navigation at LMJ Hospital

Name of Student: Chrispine Sibale

Student ID Number: 00186137

Email Address: raf33arn@gmail.com

Phone Number: 0981167823 / 0887833995

Name of Project Coordinator: Mrs Mwausegha

Name of Project Supervisor: Mrs Singini

**Abstract Page**

This project proposes the development of a mobile application designed to assist visually impaired individuals in navigating LMJ Hospital independently. The nature of the project involves integrating smartphone-based technologies—such as computer vision, natural language processing, and machine learning—to support real-time obstacle detection, voice-activated interaction, landmark recognition, and audio-based indoor navigation. The system will use on-device processing, operate offline using SQLite-stored maps, and offer a user-friendly voice interface. The intended outcome is to empower visually impaired patients by enhancing mobility, privacy, and access to healthcare services without needing physical changes to hospital infrastructure. The project is expected to contribute a scalable, affordable, and practical solution to digital accessibility within local medical environments.1

**Contents Page**

1. Introduction  
   1.1 Project Overview: Problem Statement and Aims  
   1.2 Literature Review: A Comparative Analysis of Similar Systems  
   1.3 Technological Review and Justification
2. System Requirements  
   2.1 Requirements Analysis and Information Gathering  
   2.2 Functional Requirements (MoSCoW Prioritization)  
   2.3 Non-Functional Requirements
3. Architectural Design  
   3.1 System Architecture Overview  
   3.2 Structural Design: Detailed Class Diagram
4. Behavioral Design  
   4.1 Use Case Analysis and Modeling  
   4.2 Illustrative Sequence Diagram for a Core Use Case
5. Project Management and Planning
6. Conclusion
7. Referencing
8. Appendix

**1. Introduction**

**1.1 Project Overview: Problem Statement and Aims**

Navigating the complex indoor environment of LMJ Hospital in Blantyre presents significant challenges for visually impaired individuals seeking medical care. The hospital's layout, with its multiple wings, specialized departments, and busy corridors, creates a difficult setting for autonomous movement. Traditional navigation aids like canes or guide dogs offer basic obstacle avoidance but cannot provide specific directional guidance or contextual information about the hospital's various departments, consultation rooms, or essential facilities. While hospital staff strive to assist, their urgent medical duties make consistent personal guidance impractical.2

This reliance on memory, verbal directions, or accompanying family members compromises the independence and dignity of visually impaired patients and can potentially delay their access to timely medical attention. Current accessibility features like handrails and screen readers provide some assistance but fail to offer comprehensive spatial awareness within the hospital. These limitations can generate anxiety, causing patients to postpone necessary care while also increasing the workload for hospital staff.2

The aim of this project is to address these challenges by developing a smartphone-based mobile application that provides a real-time, hands-free navigation solution tailored for the visually impaired. The application will leverage computer vision to detect obstacles, provide step-by-step audio guidance using offline hospital maps, and support hands-free interaction through voice commands.2 The system is designed to enhance accessibility, improve patient safety, and promote user autonomy without requiring modifications to the hospital's physical infrastructure.1

**1.2 Literature Review: A Comparative Analysis of Similar Systems**

A critical review of similar systems reveals a diverse landscape of technological approaches to indoor navigation. The analysis of these systems is essential for validating the core design philosophy of the proposed mobile application, which prioritizes on-device processing and independence from external infrastructure. The systems reviewed include Navigine, Seeing AI, and LowViz Guide.1

**Navigine** is a global platform provider that specializes in integrated positioning technologies. Its core functionality is built upon the deployment of external physical infrastructure, such as Bluetooth Low Energy (BLE) iBeacons and Wi-Fi signals, to achieve high-accuracy indoor positioning. While this approach can be highly effective, it necessitates substantial initial investment and ongoing calibration, as BLE signals are susceptible to interference from building materials.1 This technological philosophy fundamentally diverges from the MVI-LMJ project's explicit goal of "no physical changes to hospital infrastructure" and its emphasis on affordability. The project's design is therefore a deliberate response to the limitations of infrastructure-dependent models, seeking a more cost-effective and broadly scalable solution.

**Seeing AI**, developed by Microsoft Research, serves as a conceptual blueprint for the proposed system. It is a free mobile application that uses on-device Artificial Intelligence (AI) to "narrate the visual world" for individuals with low vision. Its innovative approach, particularly the "World" channel, provides an Audio Augmented Reality experience that announces objects in a user's surroundings using spatial audio. The system's ability to map indoor locations and create routes "with only a camera without GPS, Wi-Fi, or beacons" strongly validates the MVI-LMJ project's reliance on computer vision and AI for environmental understanding.1 This shows that a rich, context-aware form of navigation is possible without external hardware, moving beyond simple positional data to provide a dynamic "narrative" of the environment, including real-time obstacles and points of interest.

**LowViz Guide** is a specialized beacon-based app for indoor navigation, created for visually impaired users. It relies on small Bluetooth beacons to provide audio-guided navigation, particularly for temporary setups like conferences. The app’s use of pre-recorded audio messages and a changing pitch tone to indicate proximity highlights the critical importance of clear, structured, and intuitive audio feedback for visually impaired users.1 While its reliance on physical beacons aligns with the infrastructure-dependent models, its user-centric audio design offers a valuable lesson. The MVI-LMJ project must ensure that its dynamically generated audio instructions are as clear and concise as the pre-recorded messages of systems like LowViz Guide, especially when integrating real-time obstacle detection which introduces dynamic and unpredictable environmental information.

The following table summarizes the comparative analysis.

|  |  |  |  |
| --- | --- | --- | --- |
| System Name | Technological Approach | Core Functionality | Assessment against MVI-LMJ Goals |
| **Navigine** | Infrastructure-dependent (BLE beacons, Wi-Fi) | Wayfinding, foot traffic analytics, B2B platform | **Divergence:** Relies on costly physical infrastructure, conflicting with the project’s goal of "no physical changes." |
| **Seeing AI** | AI-driven camera-first (on-device processing) | Real-time text reading, scene description, object recognition | **Validation:** Confirms the feasibility of camera-based, on-device AI for rich environmental understanding and navigation without external hardware. |
| **LowViz Guide** | Beacon-based (iBeacons) | Audio-guided navigation with pre-recorded messages, proximity cues | **Lesson:** Underscores the importance of clear, structured, and intuitive audio feedback for user comprehension and trust, a key consideration for dynamic audio instructions. |

**1.3 Technological Review and Justification**

The selection of the technology stack for the MVI-LMJ project is a direct result of the design philosophy, which emphasizes affordability, offline functionality, and on-device processing. Each component has been chosen to fulfill a specific requirement of the system.1

* **Flutter & Dart:** This combination was chosen for the presentation layer due to its cross-platform compatibility, which enables the application to run seamlessly on both Android and iOS smartphones from a single codebase. This addresses the portability non-functional requirement by ensuring a wider user base can access the application without requiring separate development efforts for each platform.1
* **OpenCV & MediaPipe:** These libraries are the foundation of the processing layer, specifically for computer vision tasks. They are ideal for image analysis, real-time object detection, and obstacle recognition, which are critical for the system's ability to provide real-time, camera-based safety warnings to the user.1
* **TensorFlow Lite:** This lightweight machine learning framework is essential for enabling the efficient, on-device inference of machine learning models. The use of TensorFlow Lite ensures that the application can perform complex tasks such as object and landmark recognition without a network connection, directly supporting the non-functional requirement for offline operation and reinforcing the privacy and security principles of the project.1
* **Google TTS/STT APIs:** These APIs are used for converting text to speech and speech to text. They provide a robust and high-quality mechanism for generating the audio instructions that guide the user and for recognizing spoken commands. While these APIs can leverage cloud services, they also offer offline capabilities, which are crucial for maintaining the system's core functionality without an internet connection.1
* **SQLite:** This database engine is used for the data layer. Its serverless and self-contained nature makes it ideal for local storage of hospital maps, routes, and landmark data directly on the user's device. This design choice directly addresses the offline map access requirement, ensuring the system remains fully functional in areas of the hospital with poor or no network coverage.1
* **Rasa or similar NLP engine:** To enable hands-free, voice-activated control, an offline Natural Language Processing (NLP) engine is required. A solution like Rasa allows for the interpretation of natural language voice commands on-device, further supporting the offline and privacy requirements of the system.1

**2. System Requirements**

**2.1 Requirements Analysis and Information Gathering**

Requirement analysis is a foundational phase of the project, ensuring that the system's design is accurately aligned with the needs of its end-users. The project utilized a multi-faceted approach to gather and analyze information, which is critical for developing a truly user-centric solution.1 The information gathering methods used were:

* **Interviews:** Interviews were conducted with visually impaired individuals and LMJ Hospital staff to gain a first-hand understanding of the daily challenges, common navigation paths, and specific assistance needs within the hospital environment.
* **Observation:** On-site analysis of common navigation paths and departmental layouts was performed to identify key landmarks, potential obstacles, and the typical flow of foot traffic.
* **Document Review:** Existing hospital maps and related materials were examined to inform the initial modeling of the environment, providing a blueprint for the application’s offline map database.1

**2.2 Functional Requirements (MoSCoW Prioritization)**

The following functional requirements define the core actions the system must perform. The MoSCoW prioritization technique has been applied to categorize each requirement based on its importance to the project's success, with each requirement presented in the prescribed format of a heading followed by a detailed explanation.1

Obstacle Detection

The system must detect physical obstacles such as walls, furniture, and moving people in real time using the phone’s camera. This is a critical safety feature and is essential for the application to serve its primary purpose.

Audio Navigation

The system must guide users using voice prompts based on predefined map paths. This is the central function of the application, providing the user with turn-by-turn directions to their chosen destination.

Voice Commands

The system must allow users to request destinations or actions using natural voice commands. This enables hands-free operation and is fundamental to the usability of the application for visually impaired individuals.

Landmark Recognition

The system must identify and announce key landmarks like departments, restrooms, and exits. This provides the user with vital contextual awareness and helps them to orient themselves within the hospital.

Offline Map Access

The system must operate without the internet using preloaded hospital maps in an SQLite database. This is a non-negotiable requirement that ensures the system’s reliability in environments with poor network connectivity.

Audio Settings

The system should allow users to adjust language, voice speed, and volume settings. While not critical to core functionality, this feature significantly enhances the user experience and accessibility.

Route Recalculation

The system could adapt and suggest new routes when a user deviates from the original path. This feature would improve the system’s robustness and user trust, but it is not essential for the initial release.

Usage Feedback

The system could track anonymized usage patterns for future improvement but must not store any health-related data. This feature would be valuable for continuous improvement but is not required for the system’s core purpose.

**2.3 Non-Functional Requirements**

The following non-functional requirements define the quality attributes and constraints of the system. These requirements are essential for ensuring the system is effective, reliable, and secure in its target environment.1

Performance

The system must provide obstacle alerts and audio navigation within two seconds of detection. This low latency is crucial for user safety, particularly in a dynamic environment like a hospital.

Usability

The system must support complete voice-based interaction and accessible audio output. The entire user experience must be designed for non-visual interaction, with clear, intuitive audio cues and controls.

Portability

The app must work on Android and iOS smartphones. This requirement ensures the system is accessible to a broad user base and is not limited by a specific device ecosystem.

Security & Privacy

The system must not transmit or save personal health information; all processing remains on-device. This is a critical requirement in a healthcare context and is achieved by the project’s architectural choice to use on-device processing for all sensitive data.

Maintainability

The system should have modular components to allow updates without reworking the entire system. This design principle will simplify future enhancements and bug fixes, ensuring the system’s long-term viability.

Reliability

The system must function in varying light and noise environments typical of hospital settings. This requires robust computer vision and audio processing algorithms that can handle real-world conditions.

**3. Architectural Design**

**3.1 System Architecture Overview**

The system architecture of the MVI-LMJ application is a layered design that promotes modularity, scalability, and adherence to key non-functional requirements such as offline functionality and on-device processing.1 The architecture is composed of three primary layers: the Presentation Layer, the Processing Layer, and the Data Layer.

The **Presentation Layer** consists of the Flutter-based mobile front end. It is responsible for all user interactions, capturing visual data from the smartphone's camera and voice commands via the microphone. This layer provides all user feedback through audio from the text-to-speech engine and optional haptic vibrations, ensuring a fully accessible interface.1

The **Processing Layer** serves as the application's core logic. It contains the computer vision module, which uses OpenCV and MediaPipe for real-time image processing. It also houses the navigation logic, which determines optimal paths based on user input and map data, and the natural language processing (NLP) engine for interpreting voice commands. The use of TensorFlow Lite for on-device machine learning inference is a key component of this layer, enabling all critical processing to occur without an internet connection.1

The **Data Layer** utilizes an SQLite database for offline storage of all necessary data, including hospital maps, routes, and landmark information. This design choice is fundamental to the system's ability to operate without network connectivity, a critical requirement for a hospital environment where Wi-Fi and mobile reception can be inconsistent.1 The system interfaces with the human user through voice and physical interaction, and with automated systems via the smartphone’s camera, microphone, and sensor APIs.1

**3.2 Structural Design: Detailed Class Diagram**

The system employs a coordinator pattern with the Navigator class serving as the central orchestrator that manages navigation logic and coordinates interactions between specialized subsystems. The structural design, as modelled in the UML class diagram (Figure 6 in the original document), demonstrates a clear separation of concerns, which is a core principle of Object-Oriented Methodology. This approach ensures that each class has a single responsibility and is minimally coupled, facilitating maintainability and extensibility.1

The Navigator class maintains the system's navigation state through private attributes such as currentLocation, destination, and activeRoute. It exposes public methods for calculateRoute() and updatePosition() to manage the navigation flow. The Navigator establishes usage relationships with three key components: ObstacleDetector, LandmarkRecognizer, and MapManager. This design allows the Navigator to delegate complex tasks to dedicated subsystems without being responsible for their internal implementation.1

The ObstacleDetector is responsible for real-time hazard detection through camera processing, while the LandmarkRecognizer identifies and announces predefined locations. The MapManager serves as a shared data repository, providing access to map data from the SQLite database. The VoiceInterface class is integrated through an aggregation relationship, indicating that the Navigator contains a voice interface component for processing speech commands and providing audio feedback. This architecture successfully translates the abstract principles of OOM—modularity, reusability, and encapsulation—into a concrete, operational design.

**4. Behavioral Design**

**4.1 Use Case Analysis and Modelling**

The behavioural design of the MVI-LMJ system is defined by its use cases, which describe the interactions between the actors and the system's functionalities. The primary actor is the **Patient**, a visually impaired individual who uses the system to navigate the hospital independently. The secondary actor is the **Administrator**, IT personnel responsible for system maintenance and data updates.1

The use case diagram (Figure 4 in the original document) presents a comprehensive model of the system's capabilities. It includes critical system-level use cases such as Recognize Landmarks and Detect Obstacles & Objects, which are fundamental to the system's core purpose of providing real-time spatial awareness and safety. The diagram also illustrates key patient-centric use cases, including Navigate to Department, Process Voice Commands, and Access Emergency Help, which directly address the user's need for autonomy and safety. The inclusion of administrator use cases, such as Load Offline Maps and Monitor System Performance, demonstrates a mature understanding of the system’s complete lifecycle, from end-user functionality to back-end maintenance and data management.

**4.2 Illustrative Sequence Diagram for a Core Use Case**

The sequence diagram for the use case Navigate to Department illustrates the dynamic interaction between the user and the system's core components. This diagram provides a dynamic view of how the static class structure collaborates to fulfill a specific user request. The process begins with the **Patient** actor initiating a request via a voice command, for example, "Navigate to Radiology."

1. The **Patient** speaks the command.
2. The **VoiceInterface** receives and processes the speech command.
3. The **VoiceInterface** sends the parsed request to the **Navigator**.
4. The **Navigator** receives the request, identifies the destination, and calls the calculateRoute() method within the **MapManager** to determine the optimal path.
5. The **MapManager** accesses the SQLite database, retrieves the necessary map data, and returns the calculated route to the **Navigator**.
6. The **Navigator** then begins the navigation loop, which involves:

* Sending a request to the **ObstacleDetector** to analyze the real-time camera feed.
* The **ObstacleDetector** processes the feed and returns any detected obstacles or objects to the **Navigator**.
* The **Navigator** integrates this real-time information with the pre-calculated route and sends a structured audio instruction (e.g., "Obstacle ahead, turn right in five meters") to the **VoiceInterface**.

1. The **VoiceInterface** converts this instruction into speech and delivers it to the **Patient**.

This sequence of events continues until the LandmarkRecognizer identifies the final destination, at which point the **Navigator** provides a final voice instruction and terminates the navigation process. This flow demonstrates how the system’s modular components work in concert to provide a safe, responsive, and context-aware navigation experience for the user.

**5. Project Management and Planning**

The project is structured around a methodical and well-defined plan, as detailed in the Work Breakdown Structure (WBS) and Gantt Chart. The WBS breaks down the project into six distinct phases: Planning, Literature Review, Design, Implementation, Testing, and Final Documentation.1

The **Planning** phase, lasting four weeks in January, established the project’s foundation with the finalization of the problem statement, scope, and objectives. The **Literature Review** phase, conducted in February, involved a four-week deep dive into existing navigation aids, culminating in the finalization of the literature review document. The **Design** phase, in March, focused on designing the non-visual user interface, voice interaction flows, and the database schema, with a key milestone of completing the system and UI documentation. The **Implementation** phase represents the most intensive part of the project, spanning 16 weeks from April to August, with a focus on developing core modules such as real-time obstacle detection, voice control, and indoor navigation logic. The **Testing** phase, scheduled for September, will involve unit testing of individual modules and gathering feedback from test users. The final phase, **Final Documentation**, will occur in October, leading to the submission of the final project report and presentation. This structured plan demonstrates a clear, phased approach to development, adding credibility to the project’s feasibility and manageability.1

**6. Conclusion**

This design structure paper has provided a comprehensive overview of the MVI-LMJ project, detailing its purpose, design, and planned implementation. The analysis has confirmed that the proposed solution effectively addresses the problem of indoor navigation for visually impaired individuals at LMJ Hospital by focusing on a unique, infrastructure-independent approach. The literature review demonstrated that while existing systems like Navigine and LowViz Guide offer valuable insights, their reliance on physical infrastructure creates significant limitations in terms of cost and scalability. By contrast, the MVI-LMJ project’s reliance on on-device, camera-based AI, inspired by solutions like Seeing AI, provides a more practical and affordable solution. The architectural design, with its layered and modular structure, directly supports the project’s core non-functional requirements for offline access, security, and maintainability. Furthermore, the detailed use case and behavioral analysis, coupled with a robust project management plan, prove that the design is not only structurally sound but also functionally viable. The project is well-positioned to deliver a transformative solution that enhances the safety, independence, and dignity of visually impaired patients in the hospital environment.

**7. Referencing**

Apple, 2022. *App Store Preview: Navigine*. [Online] Available at: <https://apps.apple.com/jo/app/navigine/id972099798> [Accessed 1 July 2025].

Apple, 2025. *Image showing Seeing AI’s capabilities to Read Text, Describe Surroundings, Scan Products, Locate Items, and Recognize People*. [Online] Available at: <https://apps.apple.com/us/app/seeing-ai/id972099798> [Accessed 1 July 2025].

Corada, 2025. *LowViz Guide Navigation*. [Online] Available at: <https://www.corada.com/products/lowviz-guide-navigation> [Accessed 1 July 2025].

Guardian, S. t., 2021. *Blindnes and Visual Impairement*. [Online] Available at: <https://www.theguardian.com/society/2021/jul/22/new-device-could-help-visually-impaired-avoid-obstacles-research-suggests> [Accessed 19 April 2025].

Guild, L., 2018. *A Hybrid Indoor Positioning System for the Blind and Visually Impaired Using Bluetooth and Google Tango*. [Online] Available at: <https://lighthouseguild.org/a-hybrid-indoor-positioning-system-for-the-blind-and-visually-impaired-using-bluetooth-and-google-tango/> [Accessed 1 July 2025].

Kendrick, B., 2015. *LowViz Guide: Indoor Navigation for the Visually Impaired*. [Online] Available at: <https://www.corada.com/products/lowviz-guide-navigation> [Accessed 1 July 2025].

Liao, Y., 2024. *Seeing AI*. [Online] Available at: <https://www.microsoft.com/en-us/garage/wall-of-fame/seeing-ai/> [Accessed 1 July 2025].

Lukman, Angreani, D., Muh & Bakti, R. Y., 2025. *The Use of Artificial Intelligence in The Development of Vision Aids for The Visually Impaired*. [Online] Available at: <https://www.researchgate.net/publication/389070867_The_Use_of_Artificial_Intelligence_in_The_Development_of_Vision_Aids_for_The_Visually_Impaired> [Accessed 1 July 2025].

Medicine, N. L. o., 2020. *Indoor Navigation Systems for Visually Impaired Persons: Mapping the Features of Existing Technologies to User Needs*. [Online] Available at: <https://pubmed.ncbi.nlm.nih.gov/31979246/> [Accessed 1 July 2025].

Microsoft, 2016. *An app for visually impaired people that narrates the world around you*. [Online] Available at: <https://www.microsoft.com/en-us/garage/wall-of-fame/seeing-ai/> [Accessed 1 July 2025].

Microsoft, 2022. *Navigine Platform*. [Online] Available at: <https://appsource.microsoft.com/en-us/product/web-apps/navigine1585640136850.navigine_platform?tab=overview> [Accessed 1 July 2025].

Perkins, D., 2022. *How AI is helping the visually impaired navigate their world*. [Online] Available at: <https://www.microsoft.com/en-us/garage/wall-of-fame/seeing-ai/> [Accessed 1 July 2025].

PicoVoice, 2022. *NLP Applications in Voice Recognition*. [Online] Available at: <https://picovoice.ai/blog/voice-recognition-NLP/> [Accessed 18 April 2025].

Riehle, T., 2008. *An Indoor Navigation System to Support the Visually Impaired*. [Online] Available at: <https://umaine.edu/vemi/wp-content/uploads/sites/220/2016/08/Riehle-etalEMBC-2008-An-Indoor-Navigation-System-to-Support-the-Visually-Impaired.pdf> [Accessed 1 July 2025].

SeeingAI, 2025. *Google Play*. [Online] Available at: <https://play.google.com/store/apps/details?id=com.microsoft.seeingai> [Accessed 1 July 2025].

Tara, 2025. *LowViz Guide*. [Online] Available at: <https://www.microsoft.com/en-us/garage/wall-of-fame/seeing-ai/> [Accessed 1 July 2025].

Taylor&Francis, 2020. *Seeing AI*. [Online] Available at: <https://taylorandfrancis.com/knowledge/Engineering_and_technology/Computer_science/Seeing_AI/> [Accessed 1 July 2025].

University, C., 2022. *Computer Vision and Pattern Recognition*. [Online] Available at: <https://arxiv.org/abs/2212.04745> [Accessed 17 April 2025].

**8. Appendix**

**A.1 Use Case Descriptions**

|  |  |  |
| --- | --- | --- |
| Use Case Name | Primary Actor | Description |
| **Recognize Landmarks** | Patient | The system identifies and announces predefined landmarks such as hospital departments, waiting areas, elevators, and other significant locations to help patients orient themselves within the facility.1 |
| **Detect Obstacle & Objects** | Patient | The system uses camera-based detection to identify obstacles (wheelchairs, medical equipment, furniture) and objects in the patient's path, providing audio alerts to ensure safe navigation.1 |
| **Access Emergency Help** | Patient | Patients can quickly access emergency assistance through voice commands or gesture controls, immediately connecting them to hospital staff or security personnel when urgent help is needed.1 |
| **Navigate to Department** | Patient | The system provides step-by-step audio navigation instructions to guide patients from their current location to specific hospital departments, clinics, or service areas.1 |
| **Process Voice Commands** | Patient | The system interprets and responds to spoken instructions from patients, including navigation requests, information queries, and system control commands through natural language processing.1 |
| **Find Nearby Facilities** | Patient | Patients can locate nearby amenities such as restrooms, cafeterias, pharmacies, gift shops, ATMs, and other hospital facilities within their vicinity.1 |
| **Adjust Settings** | Patient | Users can customize system preferences including voice speed, volume levels, language selection, navigation preferences, and accessibility options to suit their individual needs.1 |
| **System Backup & Recovery** | Administrator | Administrators perform regular system backups, data recovery operations, and ensure system continuity through comprehensive backup and restoration procedures.1 |
| **Load Offline Maps** | Administrator | The system administrator uploads and maintains offline map data to ensure navigation functionality continues even during network connectivity issues or system maintenance periods.1 |
| **Update Landmark Database** | Administrator | Regular maintenance of the landmark database including adding new locations, updating existing landmark information, removing obsolete entries, and ensuring accuracy of location data.1 |
| **Monitor System Performance** | Administrator | Continuous monitoring of system performance metrics, user activity logs, error rates, response times, and overall system health to ensure optimal operation.1 |
| **Manage User Accounts** | Administrator | Creation, modification, and deletion of user accounts, assignment of access privileges, password resets, and management of user permissions within the system.1 |
| **Generate Usage Reports** | Administrator | Production of comprehensive reports detailing system usage statistics, user behaviour patterns, popular destinations, system performance metrics, and other analytical data for decision-making.1 |

**A.2 Detailed Class Definitions**

|  |  |  |
| --- | --- | --- |
| Class Name | Attributes | Methods |
| **Navigator** | currentLocation (Position), destination (String), activeRoute (Route) | calculateRoute(destination), updatePosition(newPosition), provideAudioGuidance(instruction), startNavigation(), stopNavigation() |
| **ObstacleDetector** | cameraFeed (Stream), obstacleThreshold (float) | detectObstacles(imageFrame), getDetectedObstacles() |
| **MapManager** | dbPath (String), hospitalMap (Map), landmarkData (Map) | loadMap(mapID), getRoute(start, end), getLandmarks(), getMapTiles() |
| **VoiceInterface** | ttsEngine (Object), sttEngine (Object), volume (float), voiceSpeed (float) | listenForCommand(), processCommand(command), speak(text), adjustSettings() |
| **LandmarkRecognizer** | landmarkModels (Map), recognitionThreshold (float) | recognizeLandmark(imageFrame), getRecognizedLandmark() |

**A.3 Work Breakdown Structure**

|  |  |  |  |
| --- | --- | --- | --- |
| Phase | Description | Duration | Milestones |
| **Planning** | Creating and documenting the problem statement, system scope, and objectives, including resources required and a timeline.3 | 4 weeks (January) | Finalized documentation of problem statement, project scope, and objectives.1 |
| **Literature Review** | Researching other navigation aids and mobile applications for the visually impaired.3 | 4 weeks (February) | Literature review document finalization.1 |
| **Design** | Designing the user interface with no dependence on visual elements like buttons or icons, as well as the voice interaction and computer vision flow diagrams.3 | 4 weeks (March) | System, UI, and database schema completion and documentation.1 |
| **Implementation** | Installing development tools; coding the front-end and user interface using Flutter; developing the real-time obstacle detection module; developing the voice control and interaction module; and developing the indoor navigation logic (pathfinding).3 | 16 weeks (April - August) | Developing a functional prototype and implementation of all core modules.1 |
| **Testing** | Conducting unit testing for each individual module, such as obstacle detection and voice command recognition, and gathering feedback from test users.3 | 4 weeks (September) | Finalization of testing documentation and incorporation of feedback.1 |
| **Final Documentation** | Preparing the full project documentation, including screenshots, code explanations, and a user manual.3 | 4 weeks (October) | Submission of final documentation and presentation of the project.1 |

**A.4 Gantt Chart**

A Gantt chart, as presented in the original document (Figure 5), visually represents the project timeline, showing each of the six phases in chronological order. The chart includes the duration for each phase, clearly illustrating the parallel and sequential nature of the project's activities. It highlights the intensive 16-week implementation period and the subsequent dedicated phases for testing and documentation, providing a clear roadmap for project execution and completion.1

**Works cited**

1. Mobile Application for Visually Impaired Indoor Navigation at LMJ Hospital - Requirements Specification Document.docx
2. Mobile Application for Visually Impaired Indoor Navigation at LMJ Hospital.docx
3. Chrispine “Rafe” Sibale - PROPOSAL SUBMISSION
4. REQUIREMENTS SPECIFICATION