**Smart Blind Stick**



**CIS 4923 Capstone Project 2**

**Capstone Assignment: 4  
Supervisor:**

**Team Members:**

|  |  |
| --- | --- |
| Name | ID |
| Mariam Hasan Alqahtani | H00442470 |
| Yasmeen Ali Almessabi | H00413684 |
| Kana Saleh | H00418715 |
| Khulood Qasem | H00442646 |

**Table of Contents**

[Abstract: 1](#_Toc197447773)

[Chapter # 1: Introduction: 1](#_Toc197447774)

[1.1 Context and motivation 1](#_Toc197447775)

[1.2 Problem Statement 2](#_Toc197447776)

[1.3 Project Objectives: 3](#_Toc197447777)

[1.4 Project Scope and Stockholders 3](#_Toc197447778)

[1.4.1 Project Scope 3](#_Toc197447779)

[1.4.2 Stakeholders 4](#_Toc197447780)

[1.5 Report Organization 5](#_Toc197447781)

[2 Chapter # 2: Domain related Concepts and Systems 5](#_Toc197447782)

[2.1 Introduction 5](#_Toc197447783)

[2.2 Methodology 6](#_Toc197447784)

[2.2.1 Adopted Methodology: 6](#_Toc197447785)

[2.3 Feasibility Study 7](#_Toc197447786)

[2.3.1 Technical Feasibility 7](#_Toc197447787)

[2.3.2 Operational Feasibility 9](#_Toc197447788)

[2.4 Time Management: 9](#_Toc197447789)

[2.4.1 Work Breakdown Structure 10](#_Toc197447790)

[2.4.2 Project milestones 11](#_Toc197447791)

[2.4.3 Gantt Chart 13](#_Toc197447792)

[2.4.4 Project Overview 14](#_Toc197447793)

[2.4.5 Pert Calculation 14](#_Toc197447794)

[2.5 Risk Management: 16](#_Toc197447795)

[2.5.1 DATA PRIVACY AND SECURITY RISKS 16](#_Toc197447796)

[2.5.2 SYSTEM INTEGRITY AND CYBER ATTACKS 16](#_Toc197447797)

[2.5.3 EMERGENCY RESPONSE FAILURES 16](#_Toc197447798)

[2.5.4 RISK REGISTER 16](#_Toc197447799)

[2.6 Cost Management: 20](#_Toc197447800)

[2.7 Requirements Gathering 21](#_Toc197447801)

[2.8 Conclusion: 26](#_Toc197447802)

[3 Chapter 3: Computing-based Solution/System Analysis: 26](#_Toc197447803)

[3.1 Introduction: 26](#_Toc197447804)

[3.2 Domain Requirements 27](#_Toc197447805)

[3.3 Analysis: 28](#_Toc197447806)

[3.3.1 Functional Requirements: 29](#_Toc197447807)

[3.3.2 Non-Functional Requirements: 30](#_Toc197447808)

[3.4 Conclusion 31](#_Toc197447809)

[4 Chapter 4: Computing-based Solution / System Design, Implementation, Test and validation: 31](#_Toc197447810)

[4.1 Introduction: 31](#_Toc197447811)

[4.2 Programming Environment 31](#_Toc197447812)

[4.3 Testing: 33](#_Toc197447813)

[4.3.1 Unit Testing 33](#_Toc197447814)

[4.3.2 System Testing 38](#_Toc197447815)

[4.4 Validation 38](#_Toc197447816)

[4.4.1 Survey Results 39](#_Toc197447817)

[4.4.2 Experimental Process 39](#_Toc197447818)

[4.5 Computing-based Solution/System Design: 44](#_Toc197447819)

[4.5.1 Design: 44](#_Toc197447820)

[4.6 Graphical User Interface: 45](#_Toc197447821)

[4.6.1 Architectural Pattern: 47](#_Toc197447822)

[4.6.2 Application Interface 56](#_Toc197447823)

[4.7 Implementation: 59](#_Toc197447824)

[4.7.1 Application Implementation: 59](#_Toc197447825)

[4.7.2 Hardware’s Coding Implementation: 64](#_Toc197447826)

[4.7.3 Database Implementation 65](#_Toc197447827)

[4.8 Deployment 67](#_Toc197447828)

[4.9 Conclusion 68](#_Toc197447829)

[5 Chapter 5: Conclusion: 69](#_Toc197447830)

[Reference: 71](#_Toc197447831)

**List of Figures**

[Figure 1 Agile Methodology 11](#_Toc197447832)

[Figure 2 WBS 15](#_Toc197447833)

[Figure 3 WBS 15](#_Toc197447834)

[Figure 4 Project Milestone Report 17](#_Toc197447835)

[Figure 5 Task List and Predecessors 18](#_Toc197447836)

[Figure 6 Gantt Chart 19](#_Toc197447837)

[Figure 7 Project Overview 19](#_Toc197447838)

[Figure 8 Q1 of the survey 27](#_Toc197447839)

[Figure 9 Q2 of the survey 27](#_Toc197447840)

[Figure 10 Q3 of the survey 28](#_Toc197447841)

[Figure 11 Q4 of the survey 28](#_Toc197447842)

[Figure 12 Q5 of the survey 29](#_Toc197447843)

[Figure 13 Q6 of the survey 29](#_Toc197447844)

[Figure 14 Q7 of the survey 30](#_Toc197447845)

[Figure 15 Q8 of the survey 30](#_Toc197447846)

[Figure 16 Q9 of the survey 31](#_Toc197447847)

[Figure 17: Dashboard Screen Test 39](#_Toc197447848)

[Figure 18: Sensor Monitoring Test 39](#_Toc197447849)

[Figure 19: Profile Screen Test 40](#_Toc197447850)

[Figure 20: Emergency Alerts Test 40](#_Toc197447851)

[Figure 21: Add Emergency Contact Test 40](#_Toc197447852)

[Figure 22: Emergency Contacts Test 41](#_Toc197447853)

[Figure 23: Create Account Test 41](#_Toc197447854)

[Figure 24: Login Test 41](#_Toc197447855)

[Figure 25: Survey Q1 45](#_Toc197447856)

[Figure 26: Survey Q2 46](#_Toc197447857)

[Figure 27: Survey Q3 46](#_Toc197447858)

[Figure 28: Survey Q4 46](#_Toc197447859)

[Figure 29: Survey Q5 47](#_Toc197447860)

[Figure 30: Survey Q6 47](#_Toc197447861)

[Figure 31: Survey Q7 47](#_Toc197447862)

[Figure 32: Survey Q8 48](#_Toc197447863)

[Figure 33: Survey Q9 48](#_Toc197447864)

[Figure 34: Survey Q10 48](#_Toc197447865)

[Figure 35 Class diagrams 52](#_Toc197447866)

[Figure 36 Module 1: User Authentication 53](#_Toc197447867)

[Figure 37 Module 2: Sensor and Feedback Management 54](#_Toc197447868)

[Figure 38 Module 3: Emergency Alert System 55](#_Toc197447869)

[Figure 39 Module 4: Notification Management 56](#_Toc197447870)

[Figure 40 Module 5: User Profile and Account Management 57](#_Toc197447871)

[Figure 41 Sequence Diagram 1: User Login 58](#_Toc197447872)

[Figure 42 Sequence Diagram 2: Obstacle Detection 59](#_Toc197447873)

[Figure 43 Sequence Diagram 3: Panic Button Activation 60](#_Toc197447874)

[Figure 44 Sequence Diagram 4: Notification Management 60](#_Toc197447875)

[Figure 45 Sequence Diagram 5: Account Management 61](#_Toc197447876)

[Figure 46: Dashboard Screen 61](#_Toc197447877)

[Figure 47: Sensor Monitor Screen 62](#_Toc197447878)

[Figure 48: Profile Screen 62](#_Toc197447879)

[Figure 49: Emergency Alert Screen 62](#_Toc197447880)

[Figure 50: Add Emergency Contact Screen 63](#_Toc197447881)

[Figure 51: Emergency Contacts Screen 63](#_Toc197447882)

[Figure 52: Create Account Screen 63](#_Toc197447883)

[Figure 53: Login Screen 64](#_Toc197447884)

[Figure 54: SignupView.swift 64](#_Toc197447885)

[Figure 55: AuthView.swift 65](#_Toc197447886)

[Figure 56: ThresholdSettingsView.swift 65](#_Toc197447887)

[Figure 57: EmergencyContactsView.swift 66](#_Toc197447888)

[Figure 58: HomeView.swift 66](#_Toc197447889)

[Figure 59: LoginView.swift 67](#_Toc197447890)

[Figure 60: ProfileView.swift 67](#_Toc197447891)

[Figure 61: SensorSummaryView.swift 68](#_Toc197447892)

[Figure 62: SensorsView.swift 68](#_Toc197447893)

[Figure 63: Firebase Authentication 71](#_Toc197447894)

[Figure 64: Realtime Database 71](#_Toc197447895)

[Figure 65: Cloud Firestore 71](#_Toc197447896)

[Figure 66: System's Deployment 73](#_Toc197447897)

**List of Table**

[Table 1 Similar Project 13](#_Toc197447898)

[Table 2 Key Milestones 17](#_Toc197447899)

[Table 3 PERT for Development Phase Sprints 20](#_Toc197447900)

[Table 4 Risk Register 21](#_Toc197447901)

[Table 5 Project Cost 26](#_Toc197447902)

[Table 6 Main Unit Testing 39](#_Toc197447903)

[Table 7 Unit Testing Table 42](#_Toc197447904)

[Table 8 Validation Survey 45](#_Toc197447905)

[Table 9 Hardware’s Coding Implementation 69](#_Toc197447906)

**List of Acronyms:**

|  |  |
| --- | --- |
| Acronym | Full Form |
| GUI | Graphical User Interface |
| UX | User Experience |
| UI | User Interface |
| IDE | Integrated Development Environment |
| iOS | iPhone Operating System |
| Wi-Fi | Wireless Fidelity |
| FCM | Firebase Cloud Messaging |
| ESP32 | Espressif Systems Processor 32-bit |
| MPU6050 | Microprocessor Unit 6050 (Accelerometer and Gyroscope Sensor) |
| SOS | Save Our Souls (Emergency Signal) |
| API | Application Programming Interface |

# Abstract:

In this project report, we present the Smart Blind Stick System, a technology-based solution aimed at improving the mobility and safety of visually impaired individuals. The system combines smart hardware with a user-friendly iOS mobile application to detect obstacles, water, and falls in real time. It also features an emergency alert function that notifies caregivers instantly by sharing the user’s location and condition through the app. The smart stick is built using an ESP32 microcontroller and equipped with sensors such as an ultrasonic sensor for obstacle detection, a water sensor, and an MPU6050 accelerometer for fall detection. These components work together to collect data and send it to the mobile app using Firebase. The mobile app, developed in Swift using Xcode, allows users to register, monitor live sensor data, and manage emergency contacts in a clear and accessible interface designed especially for visually impaired users. The development followed the Agile Scrum methodology, enabling flexible progress, frequent testing, and continuous user feedback. This helped us overcome key challenges, including real-time data syncing, sensor accuracy, and building an accessible UI. This project provides a smart, reliable, and accessible solution that empowers visually impaired people with greater independence and safety in their daily lives.

# Chapter # 1: Introduction:

## Context and motivation

Visually impaired people face many problems while moving around safely and on their own. Simple tools like white canes and guide dogs are helpful, but they cannot detect all types of obstacles or dangerous situations. These traditional tools also do not give enough information about the surroundings, which can lead to accidents or confusion. In the United Arab Emirates (UAE), there are around 100,000 to 120,000 people with some form of visual disability. This large number shows a strong need for better and smarter solutions that can make daily life easier and safer for them.

The purpose of this project is to design and build a Smart Blind Stick that uses modern technology to help visually impaired users. The stick will include sensors to detect objects, GPS to give location help, and AI to understand the surroundings better. It will also use computer vision to help the user know what is in front of them. These technologies will work together to provide real-time voice or vibration feedback so the person can take action quickly. One important feature of the smart stick is emergency assistance. If the user is in danger or lost, the device can send an alert to family members or emergency services with their location.

This feature adds a layer of safety and peace of mind for both the user and their loved ones. By making use of new and smart technologies, this stick will improve the way visually impaired people move in both known and unknown places. It will help reduce their need to depend on others and give them more control over their lives. In short, this project aims to bring more confidence, freedom, and safety to the lives of people who are blind or visually impaired.

## Problem Statement

According to the World Health Organization (2019), about 285 million people in the world have some form of visual impairment, and out of them, 39 million are completely blind. In the United Arab Emirates (UAE), nearly 21,000 people live with visual impairments and face daily struggles in moving around safely and independently. Even though tools like white canes and guide dogs are used, they still have many limits and do not fully solve the problem (Anwar & Aljahdali, 2017).

White canes are basic tools that only detect objects when the user touches them. This means that the person doesn’t get early warning about obstacles and might trip or fall. On the other hand, guide dogs can help a lot, but they are very expensive, need proper training, and require constant care and attention. Because of these issues, many visually impaired people cannot rely completely on these options.

As a result, blind or visually impaired individuals often find it hard to avoid obstacles, follow directions, or move confidently in complex areas such as busy roads or crowded places. These difficulties lower their independence and affect their mental and physical health (Elsonbaty & Technology, 2021).

There is a strong need for a more advanced and complete solution that does more than just detect obstacles. The ideal system should offer early alerts, improve personal safety, and provide emergency help when needed. This is where the Smart Blind Stick comes in. It combines modern technologies like sensors, GPS, and real-time alerts to help blind users move safely. With this tool, visually impaired individuals can experience better mobility, more freedom, and increased confidence in their daily lives (Agrawal & Gupta, 2018).

## Project Objectives:

The main goal of the Smart Blind Stick project is to help visually impaired people move around more safely and independently using smart technology. This device is designed to solve many daily problems faced by blind individuals and improve their quality of life. Below are the key objectives of this project:

* **Improve Mobility and Freedom:** The stick will have real-time sensors to detect nearby obstacles like walls, poles, or people. It will alert the user using vibration or sound so they can avoid the danger and walk confidently without depending on others.
* **Enhance Safety:** Safety features are very important. The Smart Blind Stick will include emergency alerts, water detection, and hazard identification to warn the user about risky situations. This will help reduce accidents and keep the user safe in places like roads, stairs, or wet areas.
* **Use AI and GPS Technology:** Artificial Intelligence (AI) will help guide users with voice instructions, and GPS will track their location and provide directions. These features will make navigation easier in both familiar and unfamiliar areas.
* **Emergency Response System:** In case of an emergency, the user can press a panic button on the stick. This will send a message to family members or emergency contacts along with the user’s location and a live photo, helping them get quick help.
* **User-Friendly Mobile App:** An iOS app will be developed that connects to the smart stick. The app will allow users or caretakers to adjust settings, track the stick’s location, and control it remotely. The interface will be simple and easy to understand, even for people who are not tech-savvy.

## Project Scope and Stockholders

### Project Scope

This project focuses on designing and developing a Smart Blind Stick equipped with modern technology such as sensors, GPS, and Artificial Intelligence (AI). The smart stick is built to help visually impaired people move safely and independently in their daily lives. It can detect nearby obstacles using ultrasonic and water sensors and provide real-time feedback through vibrations or voice alerts. This helps the user avoid dangerous objects or slippery surfaces.

The system also includes a mobile application for iOS devices, which allows users or their caretakers to control and monitor the smart stick easily. Through the app, users can customize alert settings, check the location of the stick using GPS, and receive emergency notifications. The app is designed with a simple and friendly user interface, making it easy to use even for people who are not good with technology.

The main goal of this system is to give visually impaired individuals more independence, confidence, and safety while moving around, especially in busy or unknown areas. Although the project focuses on helping people in the UAE, it can be adapted and used in other countries as well. This solution combines smart hardware and software to offer complete support for blind users.

### Stakeholders

The success of the Smart Blind Stick project depends on understanding and meeting the needs of various stakeholders. These are the individuals and groups who are directly or indirectly connected to the system and will be affected by its development and use.

**Primary Stakeholders:**

* **Visually Impaired Individuals:** These are the main users of the smart stick. They will benefit from features like obstacle detection, voice assistance, and emergency alerts. The device will help them move around more freely, with less dependence on others, improving their confidence and independence in daily life.
* **Caregivers and Families:** Family members and caregivers play a supportive role. The smart stick offers peace of mind to them, knowing their loved ones are safer and can navigate their environment with fewer risks. Features like location tracking and emergency alerts also help them respond quickly in case of danger.

**Secondary Stakeholders:**

* **Technology Providers:** Developers and suppliers of sensors, GPS modules, and AI systems are essential for the project. They ensure the technology used is accurate, reliable, and up to date.
* **Healthcare Professionals:** Doctors and therapists support the use of assistive devices like the smart stick to improve patient safety and independence.
* **Organizations Supporting Disabilities:** These include centers like Cleveland Clinic Abu Dhabi and advocacy groups that promote better accessibility and equal opportunities for visually impaired people.

## Report Organization

This report is divided into five main chapters to explain the development and purpose of the Smart Blind Stick project. Chapter 1 introduces the motivation behind the project, explains the real-world problems faced by visually impaired individuals, outlines the project objectives, and identifies the primary and secondary stakeholders involved. Chapter 2 discusses domain-related concepts, adopted methodology, work breakdown structure (WBS), feasibility study (technical and operational), time and risk management strategies, cost estimation, and requirements gathering. Chapter 3 focuses on system analysis, covering both functional and non-functional requirements based on domain needs. Chapter 4 presents the system’s design, implementation, testing, and validation. It explains the programming environment, unit testing, survey validation, interface design, and deployment details. Lastly, Chapter 5 provides the conclusion of the project and is followed by a reference section listing all sources used in the report.

# Chapter # 2: Domain related Concepts and Systems

## Introduction

This assignment focuses on creating an improved Smart Blind Stick system to help visually impaired people move around more safely and independently. The project will use system analysis techniques to fully understand what users need and to carefully plan each step of development. By doing this, the team can make sure the final product is useful, reliable, and easy to use. To make development more flexible and responsive, the project will follow the Agile Scrum methodology.

This approach allows the team to break the work into smaller tasks, known as sprints, and improve the system step by step. It also allows for regular feedback from users and stakeholders, so any changes or improvements can be made early in the process. This reduces risks and saves time.

The smart stick will include modern technologies such as ultrasonic sensors, GPS tracking, and Artificial Intelligence (AI). These technologies will help detect obstacles, find the user's location, and provide alerts in real time. A water sensor will also be added to warn users about wet or slippery areas. In addition, the device will connect to a smartphone app, allowing users or their family members to set preferences and receive emergency alerts if needed.

During the analysis phase, tools like UML diagrams will be used to show how the system works, how users interact with it, and how data flows between components. These diagrams will help developers and stakeholders clearly understand the system’s design and functions. In the end, the goal is to create a user-friendly, safe, and efficient tool that meets the real-life needs of visually impaired individuals while giving peace of mind to their families and caregivers.

## Methodology

In any software or hardware development project, choosing the right methodology is crucial for success. A methodology provides a structured way to plan, design, develop, and test the system. It also helps in managing time, reducing risks, and ensuring that all stakeholders are kept in the loop throughout the process. For the Smart Blind Stick project, which combines hardware (stick and sensors) with software (iOS application), a flexible and collaborative development approach is necessary.

The project involves many tasks like system requirement gathering, user-centered design, sensor integration, mobile app development, and testing in real environments. It is also essential to handle continuous feedback from users such as visually impaired individuals and caregivers, to ensure the final product truly meets their needs.

Several traditional methodologies like the Waterfall model or V-model were reviewed, but they are often too rigid for projects that require ongoing feedback and changes. In contrast, iterative models offer more flexibility but lack the strong team coordination features required for this project. Therefore, a more dynamic and interactive methodology is better suited for a project like the Smart Blind Stick.

### Adopted Methodology:

This project, the Agile Scrum methodology has been adopted. Agile Scrum is an iterative and incremental approach that focuses on teamwork, communication, continuous feedback, and the ability to adapt to changes quickly. It is highly suitable for our Smart Blind Stick project because the product involves both software and hardware components, evolving requirements, and real-time testing.

The development process is divided into sprints, each lasting about 2–3 weeks. During each sprint, specific features are planned, developed, and tested. For example, one sprint might focus on building the obstacle detection feature using ultrasonic sensors, while another may develop the smartphone connectivity through Bluetooth and GPS.

Each sprint begins with a Sprint Planning Meeting, where the team decides which tasks to complete. Every day, Daily Scrum meetings are held to discuss progress and remove any blockers. After each sprint, the team holds a Sprint Review to demonstrate the completed work and gather feedback from stakeholders like users and caregivers. A Sprint Retrospective follows to reflect on what went well and what needs improvement.

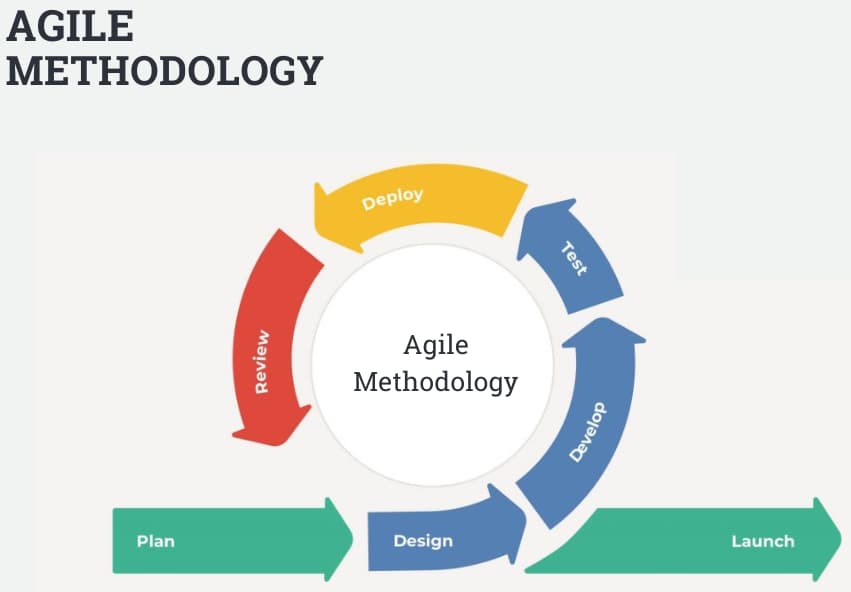


Figure 1 Agile Methodology

This approach allows the team to stay flexible, respond to user feedback, and continuously improve the product. It ensures that features like emergency alert systems, AI voice assistance, and user-friendly app functions are tested and refined in real-time. Agile Scrum helps the Smart Blind Stick team stay organized, deliver working features in short cycles, and build a high-quality, user-focused product that adapts to real-world needs effectively.

## Feasibility Study

### Technical Feasibility

There is also a branch of TRL named Technical Feasibility that determines if proper technology and expertise is available for a project together with considerations about integration, scalability, risks, and cost. The feasibility of the smart stick project in terms of the technical perspective involves seeking to identify whether the technology, resources, and knowledge are accessible depending on integration, scalability, risks, and expenses. The physical equipments used for the project are; Camera, Raspberry, GPS Navigation Device, Blind Stick, Laptop and LiDAR Range Finder Sensor. ThinkSpeak, Dweet. io, Thony, Xcode and StarUML are rather significant software tools. Due to features such as the use of GPS and AI to send real-time notifications and improve the kind of obstacles detected, this stick enhances safety. The objective of the project is to do better then the current available smart cane solutions that offer haptic feedback and help with AI features like Siri. When surveyed about sensor technologies, GPS and user interface design, the software and supports provide greater independence and less fear for the visually impaired. These courses include CIA 2513 Key Components of IoT Architecture, CIS 2303 Systems Analysis and Design and CIA 4003 Advanced Mobile Applications that has offered technical skills.

Similar Project

Table 1 Similar Project

|  |  |  |
| --- | --- | --- |
| Name | Description | Picture |
| Wewalk <https://wewalk.io/en/> | Meet your new “Intelligent Voice Assistant” with GPT integration. Your smart cane can answer anything and everything to make traveling simpler and more enjoyable! Get turn-by-turn navigation, find public transport options, and explore nearby points of interests with a simple push of a button on your smart cane. Need to know the menu prices at your local café? Ask your Voice Assistant without ever having to take your phone out of your pocket! |  |
| CAN Go  <https://can.co/> | With dependable LTE connectivity, a microphone, and excellent speakers, the integrated smartphone keeps you connected. There is an initial complimentary 12-month cellular plan, followed by a $20/month payment. The chic design provides a pleasant grasp, and the strong LED illumination guarantees security in low light. Track your exercise objectives and take advantage of fall detection, which can notify emergency contacts in case of a fall. In an emergency, GPS location tracking helps you locate your location and find your cane. The cane encourages self-reliance and wellness and is conveniently packaged with a wall mount and portable charger. |  |
| SMART CANE Blind Stick  <https://www.ebay.com/itm/174533219040> | Saarthi provides 98.2% angle precision and 99.7% obstacle detection accuracy. With simple range switching via Braille instructions, it has three navigation ranges: two feet for interior use, four feet for outdoor use, and up to eight feet for open regions. With input from more than 1,200 blind people, its ergonomic form was developed, and its rechargeable battery has a 30-day charge life. With an adjustable screw, Saarthi can be mounted on any white cane. It is intended to change as many lives as possible while remaining reasonably priced. Training for the device just takes ten minutes, making it easy to use. |  |

### Operational Feasibility

Operational feasibility is the measure of how well a proposed system solves problems and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

The smart stick for the blind will affect enhance independence and mobility by providing real-time navigation assistance and obstacle detection. This device utilizes sensors and AI to detect obstacles in the user’s path, alerting them through vibrations or audio cues, enabling and will fit perfectly with the everyday tasks of blind people, reducing their need for assistance from others, and can receive real-time position updates and information about nearby facilities, like shops, medical services, and hubs for public transportation, using the stick's GPS feature.

In addition, the system also ensures faster response in emergencies, allowing them to share their location with family members or emergency services. Should a blind person face any difficulty or danger push three times of a button, ensuring timely intervention.

## Time Management:

Any project needs time management to accomplish work on schedule, use resources efficiently, and meet goals without delay. Planning, scheduling, tracking, and adjusting to prevent bottlenecks are key to time management. The team must be organized, focused, and aware of deadlines and priorities. Time management helps sustain workflow and respond to project changes and obstacles. MS Project and Agile were used to carefully manage time for the Smart Blind Stick project. We started with a precise MS Project schedule that divided the project into commencement, planning, development, testing, and deployment. Our roadmap was obvious since each phase was broken into tasks with start and finish dates. Visualizing the chronology with Gantt charts helped the team track progress and meet deadlines.

We divided the project into four sprints to build sensor integration, emergency notifications, and mobile apps using Agile. MS Project tracked two-week sprints, illustrating work interdependence and potential delays. Regular sprint planning sessions defined goals and efficiently allocated work time. Daily standups assessed progress and resolved timeline-related concerns. After each sprint, retrospectives helped us assess our time management and improve for the next. MS Project and Agile helped us manage our time and create a working, well-integrated product on time.

### Work Breakdown Structure

A project's hierarchical Work Breakdown Structure (WBS) breaks it down into manageable parts. It defines the work needed to fulfill project goals, aiding planning, coordination, and progress monitoring. In a WBS, the project is split into stages and then into tasks or work packages. With this framework, the project team can see the whole project, understand dependencies, and manage resources efficiently. A good WBS guides the project and controls time, cost, and scope. It helps allocate tasks and create realistic deadlines for team members.

Time management relies on WBS to prioritize activities, assign resources, and predict durations. Breaking the project into smaller parts lets the team focus on each aspect without being overwhelmed by its complexity. The step-by-step breakdown helps detect hazards and delays early on by monitoring progress at each stage.

MS Project was used to create a detailed WBS for the Smart Blind Stick project to manage stages, tasks, and timeframes. MS Project lets us organize project objectives and tasks by component. Initiation, design, development, testing, and deployment were our project phases. Each work package has defined deliverables and tasks and subtasks for each phase.

Figure 2 WBS

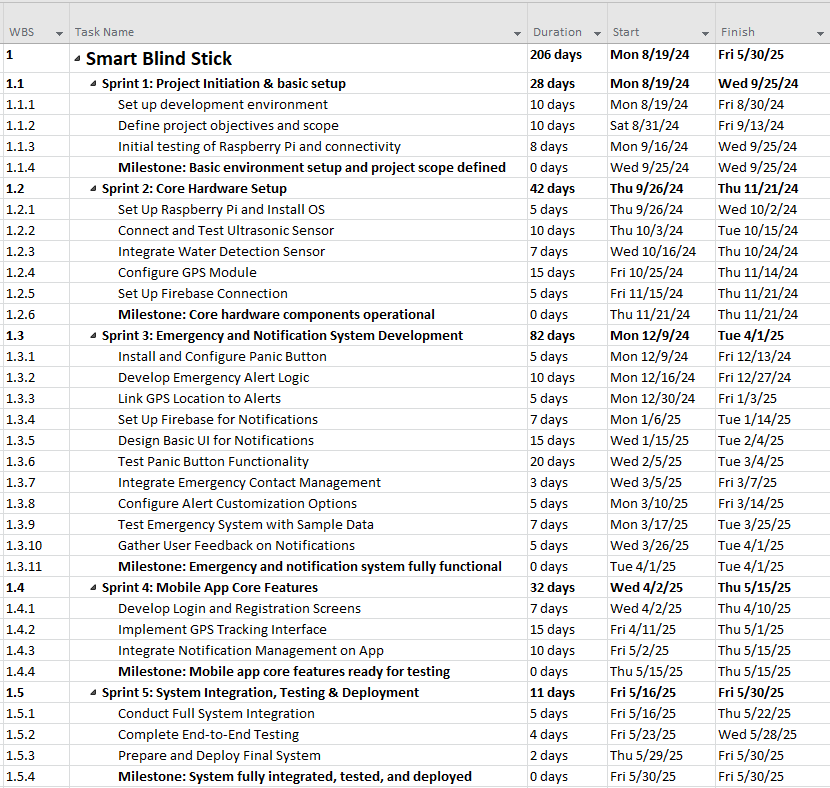


Figure 3 WBS

### Project milestones

Project milestones highlight the completion of major phases or deliverables. Milestones mark critical project choices or achievements, unlike tasks, which are continuing labor. Project team and stakeholders use milestones to measure progress, evaluate success, and determine project status. The team can also assess progress, hazards, and modifications at these moments. Milestones guide a well-structured project to completion, ensuring that all components meet project goals.

For the Smart Blind Stick project, we set milestones for critical phases and features. These sprint-aligned milestones helped us track project progress. MS Project helped us view the timetable and adapt it as we progressed by assigning milestones to phases or deliverables. Each milestone required sensor integration, GPS tracking, and the mobile app to work before going forward.

The table below summarizes the key milestones for this project:

Table 2 Key Milestones

|  |  |  |
| --- | --- | --- |
| **Sprint** | **Milestone** | **Explanation** |
| Sprint 1 | Basic environment setup and project scope defined | Initial setup of development environment and clarification of project goals. |
| Sprint 2 | Core hardware components operational | All primary hardware components are installed, configured, and functional. |
| Sprint 3 | Emergency and notification system functional | Emergency alert and notification systems are tested and ready for use. |
| Sprint 4 | Mobile app core features ready for testing | Key mobile app features, like login and GPS tracking, are implemented and tested. |
| Sprint 5 | System fully integrated, tested, and deployed | All system components are integrated, tested, and deployed for final use. |

#### Project Milestone Report

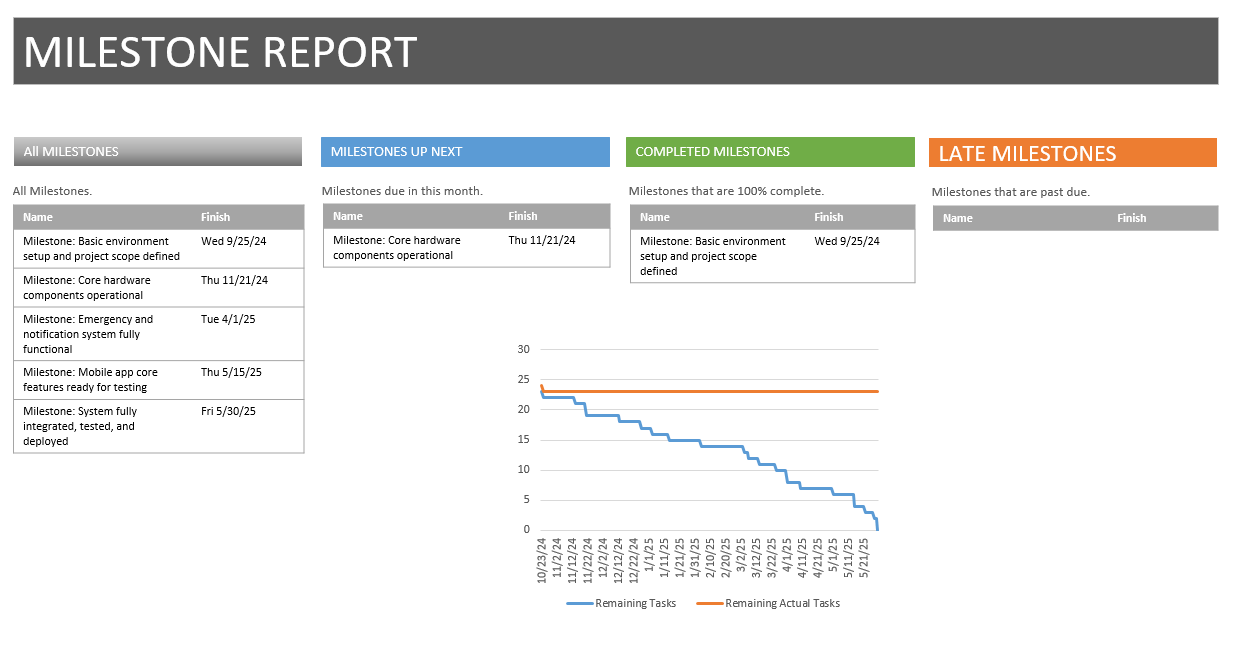


Figure 4 Project Milestone Report

#### Task List and Predecessors

From startup to deployment, the Smart Blind Stick project job list covers everything. Each activity has explicit durations and resources. A logical workflow is ensured by listing predecessors, which must be finished before others may begin. Define task dependencies, such as sprint planning before development, to arrange the project timetable and minimize delays. From sensor integration to system deployment, this framework assures seamless progress.



Figure 5 Task List and Predecessors

### Gantt Chart

The Smart Blind Stick project's Gantt chart shows task start and finish dates and dependencies. It tracks progress, manages resources, and ensures each phase follows the project timeline. This graphic shows deadlines, milestones, and important tasks, keeping the project on track from start to finish.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 6 Gantt Chart

### Project Overview

The Project Overview offers a snapshot of the current progress of the Smart Blind Stick project, showing that 28% of the tasks have been completed. It highlights upcoming key milestones, such as system integration and final deployment, with clearly defined deadlines to ensure the project remains on schedule.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 7 Project Overview

### Pert Calculation

To estimate the duration of each task, three different scenarios are considered:

**Optimistic Scenario (TO):** Assumes that everything proceeds smoothly without any delays or complications, representing the best-case outcome for the task.

**Most Likely Scenario (TM):** Accounts for typical uncertainties and minor delays, representing the most probable duration for the task.

**Pessimistic Scenario (TP):** Anticipates significant delays or major issues, including external dependencies, resource constraints, or unforeseen events, representing the worst-case outcome for the task.

The estimated duration for each task is expressed in days and calculated using the following formula based on the beta distribution:

A black numbers and a line

AI-generated content may be incorrect.

**PERT for Development Phase Sprints:**

Table 3 PERT for Development Phase Sprints

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| WBS | Task | Optimistic (O) | Most Likely (M) | Pessimistic (P) | PERT Duration |
| 1.1 | **Sprint 1: Project Initiation & basic setup** | 24 | 28 | 32 | 28 |
| 1.2 | **Sprint 2: Core Hardware Setup** | 38 | 42 | 46 | 42 |
| 1.3 | **Sprint 3: Emergency and Notification System Development** | 78 | 82 | 86 | 82 |
| 1.4 | **Sprint 4: Mobile App Core Features** | 28 | 32 | 36 | 32 |
| 1.5 | **Sprint 5: System Integration, Testing & Deployment** | 7 | 11 | 15 | 11 |

## Risk Management:

For the Smart Blind Stick project, security and forensics-related risks are critical, especially concerning data privacy, system integrity, and emergency response reliability. Below are the categorized risks and the mitigation strategies.

### Data Privacy And Security Risks

* Risk: The smart stick system collects sensitive data such as GPS location and images, which could be intercepted or misused.
* Mitigation: Implement end-to-end encryption for data transmission between the smart stick and the mobile app. Secure all stored data using strong encryption methods and enforce multi-factor authentication (MFA) for accessing sensitive information.

### System Integrity And Cyber Attacks

* Risk: The system could be vulnerable to cyberattacks like hacking or malware that could compromise its functionality.
* Mitigation: Regular security audits and penetration testing will be conducted to identify vulnerabilities. Anti-malware software will be installed, and security patches will be updated regularly.

### Emergency Response Failures

* Risk: Failure to send emergency notifications due to network issues could delay assistance.
* Mitigation: Implement redundant communication channels (Wi-Fi, mobile data) to ensure notifications are sent through alternative means if one channel fails. Real-time system monitoring will alert administrators to any disruptions.

### RISK REGISTER

Here is the comprehensive Risk Register table for the Smart Blind Stick project, focusing on security and forensics risks:

Table 4 Risk Register

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Risk ID | Risk Description | Category | Impact | Likelihood | Severity | Mitigation Strategy | Fallback Plan | Responsible Party | Review Date |
| R1 | Data privacy breach due to intercepted GPS or images | Data Security | High | Medium | Critical | Encrypt data during transmission and storage. Implement multi-factor authentication (MFA). | Activate data encryption alerts. Temporarily suspend the system to investigate and patch. | Data Security Team | Bi-monthly |
| R2 | Cyberattacks such as hacking or malware infiltration | System Integrity | High | Medium | Critical | Regular security audits, penetration testing, and install anti-malware software. | Initiate system lockdown, disable external access, and restore from secure backups. | IT Security Team | Quarterly |
| R3 | Failure of emergency notification system due to network outages | Operational Failure | High | Low | Critical | Use redundant communication channels like Wi-Fi and mobile data for backup. | Notify caregivers manually. Activate SMS alert as a secondary communication method. | Operations Team | Weekly |
| R4 | Unauthorized access to system controls or user accounts | Access Control | Medium | Medium | Major | Implement role-based access control (RBAC) and MFA for system users. | Disable compromised accounts and restore access controls. Re-issue new credentials. | Data Management Team | Monthly |
| R5 | Data loss due to system crash or hardware malfunction | Data Availability | Medium | Low | Major | Schedule regular system backups and ensure cloud redundancy for critical data. | Restore from the latest backup. Notify users of any downtime and perform recovery actions. | IT and Backup Team | Quarterly |
| R6 | Delays in emergency response due to failure of panic button system | Hardware/System Failure | High | Low | Critical | Regular testing of the panic button system and real-time monitoring of hardware status. | Use manual notifications through the app and enable GPS tracking in the interim. | Hardware & App Dev Team | Weekly |
| R7 | Unauthorized data access from weak third-party integration | Third-Party Risk | High | Medium | Major | Conduct regular third-party security assessments and monitor access logs. | Immediately revoke third-party access. Perform thorough system audit to identify any breaches. | Vendor Management Team | Quarterly |
| R8 | GPS inaccuracy affecting real-time location updates | Technical/Operational | Medium | Medium | Major | Use high-precision GPS modules and periodically recalibrate location services. | Switch to manual location entry if available. Use additional location-based services as backup | Operations Team | Bi-monthly |

## Cost Management:

Establishing the estimated cost" refers to calculating the resources needed to complete project obligations. This method's ability to depict the complete amount of cash required for project completion is one of its main advantages. As a result, this procedure is followed regularly during the project.

The team has determined that several resources are required for our endeavor. This includes human resources, such as software developers, who are crucial to the success of our project.

Furthermore, certain tools will be needed, including actual hardware for code testing, The process of strategic planning and managing a business's financial resources is known as cost management.

We used a parametric cost-estimating method in our project. Using historical data, this method divides the project materials into discrete portions and calculates the cost of each section. As Kelechi Udoagwu mentioned in 2022, we then add together these prices from each separate part to get an estimate for the whole project cost.

Table 5 Project Cost

|  |  |  |
| --- | --- | --- |
| Item | Cost | Total |
| Normal blind Stick | 50 AED |  |
| Ultrasonic sensor | 10 AED |  |
| MPU 6050 | 170 AED |  |
| Buzzer | 40 AED |  |
| ESP 32 | 300 AED |  |
| Button | 100 AED |  |
| Water Sensor | 80 AED |  |
| MacBook Air 13 | 4700 AED x 4 | 18,800 AED |
| Xcode | Free of cost |  |
| Developers | 50 per day x 50 days | 2,500 AED |

## Requirements Gathering

Requirements gathering is an important process in which we identify the project's exact requirements. System requirements can be gathered in several methods, and a survey is one of them. Because they enable the simultaneous collection of requirements from many different stakeholders, surveys, and questionnaires are effective substitutes for interviews. It is important to develop and construct a good questionnaire that will ask insightful questions and help with identifying core requirements that might not be immediately apparent when using the survey as a method for system requirements.

We have created an online survey and distributed it via digital communication channels. We created the survey on Microsoft Forms. The questionnaire consisted of nine multiple-choice and short-answer questions. Data were collected in October 2024. The sample population was family and friends. A total of 46 responses were collected in the survey.

The results of the survey are the following:

A close-up of a chart

Description automatically generated**Q1. How helpful would a smart stick that detects obstacles be for a blind person in new places?**

Figure 8 Q1 of the survey

Based on the survey results with a total sample size of 46 respondents, approximately 65% of individuals (30 out of 46) That majority view obstacle detection as highly beneficial, especially for navigating unfamiliar areas. This indicates that people believe this feature will significantly improve safety and independence.

**A graph with numbers and a circle

Description automatically generated with medium confidenceQ2. Which feature would help a blind person the most?**

Figure 9 Q2 of the survey

According to the survey responses, when asked to choose the most features that help the blind person, around 23% of respondents reported detecting obstacles the most, vibration alerts and Emergency buttons 22% of the respondents, and Phone notifications and Detecting water 18%. The results show that obstacle detection, emergency support, water detection, vibration, and Phone notification alerts are all valued features. People prioritize features that improve safety and offer immediate assistance in case of danger.

**Q3. How important are alerts to keep a blind person safe while using the stick?A pink circle with black text

Description automatically generated**

Figure 10 Q3 of the survey

Based on the survey responses, it is clear that many people encounter challenges when trying to keep a blind person safe. A significant 63% response is "very important" to feel alerts are essential for safety, reflecting strong support for features that provide real-time feedback to avoid hazards and emergencies.

**A screenshot of a computer

AI-generated content may be incorrect.Q4. Do you think this smart stick could help a blind person who needs less help from others? Give reasons.**

Figure 11 Q4 of the survey

Question Four in the survey was open-ended. In their responses, People highlighted the importance of feedback, emphasizing a desire for smart blind sticks to ensure the safety of the blind. In the survey, respondents believe the smart stick will enhance independence by increasing confidence and allowing safer, self-directed navigation. Common responses include that it "warns about obstacles" and "keeps them safe," emphasizing the value of hazard alerts.

**A close-up of a graph

Description automatically generatedQ5. How much do you think an emergency button could help a blind person be more independent?**

Figure 12 Q5 of the survey

Based on the survey results, around 67% Of respondents believe the emergency button is highly helpful for independence, suggesting that quick access to help is seen as a critical safety net that enables solo mobility.

**A screenshot of a computer

Description automatically generatedQ6. How confident would you be to suggest this stick to a blind person?**

Figure 13 Q6 of the survey

Based on the feedback from the survey respondents, it looks like opinions on the current confidence of stick about a blind person and requirements vary. About 67% of people chose "Very confident," 33% chose "Somewhat confident," and none chose "Not confident." Most respondents feel positive about recommending the stick, indicating high confidence in its utility and safety features.

**Q7. Do you think phone notifications would make family or friends feel better about a blind person’s safety?**

**A screenshot of a computer

Description automatically generated**

Figure 14 Q7 of the survey

Survey results show that a significant 76% believe that receiving notifications would positively impact a blind person’s safety. People strongly agree that location notifications would reassure family or friends, indicating the importance of shared safety information for peace of mind.

**A close-up of a graph

Description automatically generatedQ8. What is the biggest worry a blind person might have about using a smart stick?**

Figure 15 Q8 of the survey

About 46 of people about blind people worry, and there are some responses: 23% for "Hard to understand," 43% for "Battery and tech problems," 19% for "Doesn’t last long," and 16% for "Not accessible enough." The primary concern is technology reliability, especially regarding battery life and potential technical difficulties. Usability and accessibility also matter as respondents see these as potential barriers.

**Q9. How likely is it that a smart stick would help a blind person feel more independent and safe?**

**A graph with numbers and a pink circle

Description automatically generated with medium confidence**Survey results show responses eight 52% chose "Very likely," 28% chose "Somewhat likely," 20% chose "Neutral," and none chose "Unlikely." Most respondents feel that the smart stick would positively impact independence and safety, though a few remain neutral. This suggests that the stick's design and functionality must directly support these needs to meet expectations fully.

Figure 16 Q9 of the survey

## Conclusion:

In this section has provided a comprehensive overview of the foundational planning and development strategies for the Smart Blind Stick project. We discussed the chosen Agile Scrum methodology, which ensures flexibility, teamwork, and continuous feedback throughout the development cycle. The feasibility study evaluated the technical and operational practicality of the system, confirming its potential to be effectively implemented. Cost management was outlined to ensure the project remains within budget while maintaining quality, and time management strategies, including a network diagram, were designed to track progress and meet deadlines efficiently. A detailed Work Breakdown Structure (WBS) was developed to organize tasks and responsibilities, ensuring smooth workflow and accountability. Lastly, requirement gathering helped identify and define the functional and non-functional needs of users and stakeholders. Together, these components form the core of our planning process, setting a strong foundation for the successful design and implementation of the Smart Blind Stick system.

# Chapter 3: Computing-based Solution/System Analysis:

## Introduction:

This Chapter analyzes and designs the Smart Blind Stick system to help visually impaired people with movement and safety. This chapter uses organized approaches to determine system requirements, analyze functions, and define architecture to create a solid framework. The system promises to improve user independence, safety, and confidence with cutting-edge obstacle recognition, water hazard alarms, GPS tracking, and emergency response functions.

This chapter begins with a domain requirements analysis to ensure the system meets end-user expectations. To create a complete solution, these needs address accessibility, hazard detection, emergency features, scalability, and privacy. System behavior and performance are also specified by functional and non-functional criteria.

We made case diagrams, sequence diagrams, activity diagrams, and class diagrams to turn requirements into actionable designs. These visual tools help stakeholders and developers understand the workflow and relationships in the solution by capturing the dynamic interactions between system components. The use case diagrams show user interactions, while the class diagrams clarify the system's structural relationships to meet user expectations.  
The system is developed iteratively using Agile methods to allow for flexibility and real-time feedback and testing. Agile ensures that each sprint generates a working system increment, with user feedback driving improvements.

The analysis comprises a product backlog that prioritizes user-important items to keep the development process focused and efficient. This chapter establishes a computer solution with advanced technology and user-centric design. This chapter prepares the Smart Blind Stick system to provide a novel and impactful solution for visually impaired people by systematic analysis, extensive requirement collection, and structured modeling.

## Domain Requirements

* **Accessibility for Users:**
  + Provide feedback through vibrations, sounds, or a simple mobile app interface.
  + Support screen readers and voice commands for easy use.
* **Hazard Detection:**
  + Detect obstacles, water, and drops within a 2-meter range.
  + Alert users with vibrations or sounds within 2 seconds of detection.
* **Emergency Features:**
  + Activate emergency mode by pressing the panic button three times.
  + Send GPS location and a photo to emergency contacts within 5 seconds.
* **GPS Tracking:**
  + Track the user’s location in real-time via the mobile app.
  + Show location history and current location on a map.
* **Data Security:**
  + Secure all data, like GPS and photos, with encryption.
  + Use role-based access to restrict sensitive features.
* **Power Efficiency:**
  + Run for at least 8 hours on a single charge.
  + Alert users when the battery is low.
* **Durability:**
  + Work well indoors, outdoors, on wet surfaces, and in low light.
  + Be sturdy enough to handle drops and minor impacts.
* **Scalability:**
  + Support more users and updates without replacing hardware.
  + Allow app and system updates to be done easily.
* **Stakeholder Features:**
  + Let caregivers manage emergency contacts and track multiple sticks.
  + Allow admins to manage user accounts and monitor the system.
* **Privacy:**
  + Protect user data and comply with privacy laws.
  + Ensure only authorized people can access sensitive features.

These requirements focus on making the stick user-friendly, secure, and reliable for visually impaired users.

## Analysis:

The Smart Blind Stick System addresses the mobility challenges faced by visually impaired individuals, as traditional walking aids often do not effectively help in detecting obstacles and hazards, which can lead to accidents and limit independence. This system enhances safety and confidence by incorporating advanced technologies like obstacle detection, water detection, and GPS navigation. It allows users to adjust settings, manage emergency contacts, and receive notifications through a mobile app. We are using an Agile methodology for development, enabling us to continuously implement new requirements as they arise. Additionally, we will utilize UML diagrams, such as use case diagrams and activity diagrams, to visually represent system requirements and illustrate the interactions between various components.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Product Backlog** | | | | | |
| ID | As a | I want to be able to | so that | Priority | status |
| 1 | Blind person | easily turn the stick on or off using a button | I can control when the stick is in use. | High | Not started yet |
| 2 | Blind person | press the panic button more than 3 times to notify my emergency contact in case of danger | I can receive assistance quickly. | High | Not started yet |
| 3 | Blind person | detect obstacles in my path | I can avoid collisions and move safely. | High | Not started yet |
| 4 | Blind person | detect water on the ground | I can avoid wet or slippery areas. | High | Not started yet |
| 5 | User | Register in the app | I can manage the blind person’s stick and receive alerts. | High | Not started yet |
|  |
| 6 | User | log into the app | I can access features such as GPS tracking and notification management. | High | Not started yet |  |
| 7 | User | track the blind person’s real-time GPS location | I can ensure their safety and respond quickly in emergencies. | High | Not started yet |  |
| 8 | User | manage my notification settings | I can choose how and when I am notified of emergencies or other events. | Medium | Not started yet |  |
| 9 | Caregiver | manage the list of emergency contacts | the right people are notified during an emergency. | High | Not started yet |  |
|  |
| 10 | Admin | manage user accounts | can activate, deactivate, or reset accounts when necessary. | Medium | Not started yet |  |

### Functional Requirements:

Functional requirements describe the specific functions and behaviors the Smart Blind Stick System must support to ensure safety, usability, and real-time responsiveness for visually impaired users.

**Input:**

* Users can create an account by entering their personal information through the mobile app.
* Users can log in securely using their registered credentials.
* Users can add, edit, or delete emergency contacts within the app.
* Users can configure threshold settings for obstacle detection, fall detection, and water detection.
* Users can press the physical SOS button on the smart stick to trigger an emergency alert.
* The system receives live input from sensors including ultrasonic distance, water presence, and motion/orientation data from the MPU6050.

**Output:**

* The mobile app displays real-time sensor values including distance, motion, and water detection.
* Users receive immediate alerts on the app when an obstacle, fall, or water hazard is detected.
* Emergency contacts receive instant notifications via Firebase when an emergency is triggered.
* The system activates a buzzer sound when the SOS button is pressed or a fall is detected.
* Sensor data is updated in real time on the app without manual refresh.

**Process:**

* The system verifies user credentials through Firebase Authentication before granting access.
* The ESP32 processes sensor input and sends data continuously to Firebase Realtime Database.
* The mobile app retrieves and displays updated sensor data as soon as it's received from Firebase.
* When an emergency condition is detected or the SOS button is pressed, alerts are sent automatically.
* All user data, contact information, and sensor thresholds are stored securely and synced through Firebase.

### Non-Functional Requirements:

**Performance:**

* The system will be user-friendly, ensuring ease of use for visually impaired users and caregivers.
* The system will be available 24/7 to ensure continuous support.
* The system will respond quickly to sensor detections, with a response time not exceeding 2 seconds for obstacle alerts.
* The system will be scalable to support an increasing number of users and connected devices.

**Control:**

* Users must authenticate through the app to access personal settings and alerts.
* Users and caregivers have access to only the features they are authorized for (updating emergency contacts).

## Conclusion

In this section, we focused on identifying and analyzing the functional and non-functional requirements, along with the domain-specific needs of the Smart Blind Stick system. The functional requirements define the key operations the system must perform, such as real-time obstacle detection, GPS-based navigation, water detection, emergency alerts, and voice-guided support. These functions are essential for assisting visually impaired users in moving safely and independently. The non-functional requirements address important system qualities like reliability, performance, usability, security, and maintainability, ensuring that the device not only works efficiently but also remains user-friendly and secure. Additionally, we examined the domain requirements to better understand the expectations and unique challenges faced by visually impaired individuals. This included studying existing assistive technologies, user behavior, and environmental conditions to design a solution that truly fits the needs of the target users. By analyzing these requirements in depth, we laid a solid foundation for the development of a smart and dependable system that enhances the daily lives of its users.

# Chapter 4: Computing-based Solution / System Design, Implementation, Test and validation:

## Introduction:

This chapter presents the comprehensive design, development, implementation, testing, and validation of the Smart Blind Stick System, an assistive technology solution aimed at enhancing the safety and independence of visually impaired individuals. The chapter outlines the step-by-step approach taken to engineer a fully functional and responsive system that integrates smart hardware components with a real-time mobile application.

The Smart Blind Stick was conceptualized to bridge the limitations of traditional mobility aids by incorporating advanced sensors, wireless communication, and emergency alert capabilities. The system design focuses on real-time obstacle detection, fall recognition, and water hazard alerts, all of which are processed and transmitted to a user-friendly iOS mobile app built using Swift and SwiftUI. The app serves as the user’s main interface, providing real-time feedback, emergency contact management, and secure authentication via Firebase.

Throughout the chapter, we detail the selected programming environments, hardware-software integration, and cloud-based database setup that underpin the project. Rigorous unit and system testing were conducted to ensure each feature met its intended function reliably, while a structured validation process involving real users confirmed the system’s effectiveness and accessibility.

This chapter also includes the graphical user interface design, interaction diagrams, architectural patterns, and application interface components that support the project. Additionally, we cover the coding implementation for both hardware and software modules, database configuration, and the final deployment process. Together, these sections demonstrate the complete engineering lifecycle of the Smart Blind Stick System from initial planning to functional delivery.

## Programming Environment

The programming environment for the Smart Blind Stick was carefully selected to ensure best performance, functionality, and seamless hardware-mobile application interface contact. The development tools and technologies applied in this project are broken out here in great depth.

**Integrated Development Environment (IDE):**

* **Xcode:** We decided to create the iOS application mostly using Xcode as the programming environment. Xcode ensures a flawless development process for an accessible and responsive mobile application by means of strong debugging tools, a real-time interface builder, and flawless connection with Swift.

**Programming Language:**

* **Swift:** Designed with speed, security, and Apple ecosystem compatibility in mind, Swift is the mobile app tool for creating a program that gives user experience top priority, its simple syntax and accessibility tools are quite good.
* **C Language:** Using the Arduino IDE, the ESP32 microcontroller and linked sensors are programmed in C ensuring exact sensor data handling and real-time communication.

**Database Management:**

* **Firebase:** Firebase ensures fast and safe processing of alarms and sensor updates, so the Smart Blind Stick is a dependable helpful tool.

**Sensor Data Integration:**

* **Arduino IDE with C Language:** Using Arduino IDE running C Language, the ESP32 microcontroller was programmed and several sensors—including the ultrasonic sensor, MPU6050 accelerometer, water sensor, emergency button, and buzzer—were combined. This setup ensures perfect identification of environmental hazards, falls, and obstacles even when real-time alarms are being enabled.

**Microcontroller Programming:**

* **ESP32:** Handling sensor inputs, data processing, wirelessly relaying information to the mobile application, the ESP32 is the essential processing unit. The ESP32 was chosen for its low power consumption, Wi-Fi features, numerous sensor compatibility, so ensuring seamless real-time communication between hardware and software components.

**Real-Time Alert and Monitoring Integration:**

* **Firebase Cloud Messaging (FCM):** When the system senses anything such as a fall, an obstacle, or the emergency button pressed, Firebase Cloud Messaging (FCM) sends immediate messages to the caregivers or the emergency contacts. This ensures immediate response and help as required to the visually impaired individuals.

By carefully selecting these technologies, the Smart Blind Stick programming environment provides a stable, responsive, and secure system that enhances the mobility and safety of visually impaired individuals.

## Testing:

In the testing phase of the Smart Blind Stick project, we conducted thorough system test scenarios to validate the platform’s accuracy, usability, and reliability. Key areas tested included user authentication, live sensor data monitoring, obstacle and fall detection alerting, emergency contact management, and cloud database integration. These tests ensured the system delivers secure, correct, and seamless support for visually impaired users and their caregivers.

### Unit Testing

We performed detailed unit tests on each module and screen to ensure all features functioned exactly as intended. Below is an overview of the tested modules:

* **Real-Time Sensor Monitoring:** Confirmed accurate, live updates of sensor data on the mobile app.
* **SOS Emergency Activation:** Verified emergency alert triggering from the stick and app side.
* **Obstacle Detection:** Tested reliable detection of nearby obstacles using the ultrasonic sensor.
* **Fall Detection:** Validated correct detection of sudden falls using the MPU6050 sensor.
* **Water Hazard Detection:** Ensured timely alerts when water hazards were sensed.
* **User Registration & Login:** Tested secure account creation and session management via Firebase Authentication.
* **Emergency Contact Management:** Verified addition, deletion, and storage of emergency contacts.
* **Data Synchronization:** Confirmed real-time updates and synchronization between ESP32 sensors and the mobile app via Firebase.
* **Mobile App Responsiveness:** Tested app performance across different iOS devices and screen sizes.

All unit tests were successfully completed, and the results confirmed that individual modules operated reliably, and no major functional bugs were detected.

Table 6 Main Unit Testing

|  |  |
| --- | --- |
| Descriptions | Screen |
| Test: Display real-time sensor data and trigger emergency alert when thresholds are crossed. Result: Real-time updates worked correctly, and SOS button successfully triggered emergency notification. | Figure 17: Dashboard Screen Test |
| Test: Display acceleration, gyroscope, distance, and water level sensor data dynamically. Result: All sensor data displayed accurately and updated without needing a manual refresh. | Figure 18: Sensor Monitoring Test |
| Test: Allow users to view, edit, and update personal and medical information; verify logout functionality. Result: Profile information edited and saved successfully; logout worked securely. | Figure 19: Profile Screen Test |
| Test: Show critical alert screen when dangerous sensor readings are detected and allow emergency calling. Result: Critical alerts displayed correctly, and emergency call feature worked as intended. | Figure 20: Emergency Alerts Test |
| Test: Allow users to input name, phone number, and relationship for adding emergency contacts. Result: Emergency contacts added successfully and validated against empty or incorrect entries. | Figure 21: Add Emergency Contact Test |
| Test: Allow users to view, edit, and delete emergency contacts dynamically. Result: Emergency contacts were correctly added and deleted; changes reflected immediately. | Figure 22: Emergency Contacts Test |
| Test: Allow new users to sign up by entering full name, email, password, and confirm password with validation. Result: Registration completed successfully with correct inputs; errors displayed for invalid data. | C:\Users\Unknown\AppData\Local\Packages\5319275A.WhatsAppDesktop_cv1g1gvanyjgm\TempState\85AAD03A5D5F909A016FDF0D47568E49\WhatsApp Image 2025-03-21 at 14.57.18_d661221a.jpg  Figure 23: Create Account Test |
| Test: Authenticate users with email and password, and handle incorrect login attempts. Result: Valid users logged in successfully; invalid attempts showed proper error messages. | C:\Users\Unknown\AppData\Local\Packages\5319275A.WhatsAppDesktop_cv1g1gvanyjgm\TempState\6A639C52538FD7BB3E1186724AFBE9B2\WhatsApp Image 2025-03-21 at 14.57.28_b763f2cc.jpg  Figure 24: Login Test |

**Unit Testing Table:**

Table 7 Unit Testing Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ****Test Case**** | ****Description**** | ****Input Data**** | ****Expected Result**** | ****Actual Result**** | ****Pass/Fail**** | ****Remarks**** |
| **User Registration** | Register a new user with email and password | User details entered | User successfully registered | Registration completed | Pass | Positive test case |
| **User Login** | Log in with registered credentials | Email and password entered | Login successful | User logged in successfully | Pass | Positive test case |
| **Sensor Data Display** | Display real-time sensor readings | Open the app home screen | Live sensor data shown | Data displayed correctly | Pass | Functional test |
| **Obstacle Detection** | Detect objects in front of the user | Ultrasonic sensor input | App notifies user of obstacles | Alert triggered correctly | Pass | Threshold tested |
| **Fall Detection** | Detect if the user falls | MPU6050 sensor values change | App sends fall alert | Notification sent successfully | Pass | Sensor accuracy verified |
| **Water Detection** | Detect wet or slippery surfaces | Water sensor detects water | App warns the user | Alert triggered correctly | Pass | Functional test |
| **Emergency Button** | Manually activate emergency alert | Press emergency button | Buzzer sounds and alert sent | Alert and sound activated | Pass | Verified response |
| **Buzzer Response** | Ensure buzzer activates when needed | Emergency button pressed | Buzzer sounds | Audible alarm triggered | Pass | Functional test |
| **Emergency Contact** | Add and save emergency contacts | Contact details entered | Contact saved | Contact added successfully | Pass | Positive test case |
| **Real-Time Sync** | Ensure real-time updates between hardware and app | Sensor values change | App updates data instantly | Data synchronized without delay | Pass | Functional test |
| **UI Usability Test** | Test UI navigation and responsiveness | User interacts with app UI | UI elements function correctly | Interface worked smoothly | Pass | Verified accessibility |

### System Testing

In system testing, we evaluated the entire Smart Blind Stick System from hardware setup to mobile application interaction and real-time emergency response. These tests ensured that all modules sensors, microcontroller, cloud database, and mobile application — work together as a complete, reliable, and accessible system.

The following workflows were tested successfully:

* User registration and login on the mobile app, ensuring secure authentication and smooth redirection to the dashboard.
* Smart Stick sensor data transmission to the mobile app via Firebase Real-time Database.
* Real-time alerts for obstacle detection, fall detection, and water hazard detection.
* Emergency SOS button triggering immediate alerts to caregivers and activating the buzzer.
* Dynamic rendering of live sensor values, emergency contacts, and alert notifications on the app interface.

## Validation

To ensure that the Smart Blind Stick System was ready for real-world deployment, we conducted a structured user validation survey focusing on key aspects such as functionality, ease of use, system responsiveness, and overall satisfaction. Our goal was to assess how effectively the system improved user safety and independence.

* We created a focused survey with 10 multiple-choice questions and distributed it among selected users, including visually impaired individuals and caregivers. The survey covered areas such as registration and login experience, real-time sensor monitoring, emergency contact management, alert response times, and system usability. Participants rated their experience and provided feedback based on hands-on interaction with both the stick and the mobile app.
* The collected survey responses helped us identify strengths and improvement areas. Users appreciated the system’s ease of use, reliable real-time alerts, and clear emergency notifications. Some suggestions included enhancing the visual contrast for better readability and fine-tuning the sensitivity of fall detection.

Based on the feedback, we implemented several updates, such as improving the app's UI accessibility, refining sensor thresholds for fall detection, and optimizing the emergency alert delivery process. These enhancements ensured that the Smart Blind Stick System is not only functional but also user-centered, secure, and fully aligned with the real needs of visually impaired individuals and their caregivers.

### Survey Results

The survey conducted was designed to assess the non-functional requirements of the **Smart Blind Stick System**. These questions focused on key aspects such as:

* **Ease of use**: How intuitive and user-friendly is the mobile application and smart stick?
* **Reliability**: How dependable is the system in terms of real-time sensor data updates and emergency alert notifications?
* **Usability**: How well does the system work across different devices and environments, ensuring it remains accessible and functional?
* **Performance**: How efficiently does the system operate, especially under different user interactions or when the system is under heavy load?

The goal of this survey was to gather insights into the system’s overall effectiveness and usability, ensuring that it meets the accessibility and reliability needs of its target audience — visually impaired individuals.

### Experimental Process

To ensure the validity of the survey responses, we allowed both visually impaired individuals and non-visually impaired individuals (including family and friends) to test the system. The non-visually impaired participants were asked to use both the **Smart Blind Stick** and the **mobile application**, following the same steps as the target users. They were instructed to explore the functionalities of the system, such as obstacle detection, fall detection, water hazard alerts, and emergency SOS activation, before answering the survey questions.

This approach helped us collect feedback not only from those who would directly benefit from the system but also from individuals without visual impairments, offering a more comprehensive view of the system's usability. We encouraged the participants to provide honest feedback on their experience using the system and the app, helping us identify potential usability issues, performance concerns, and areas for improvement.

The feedback from the experiment was invaluable in refining the **Smart Blind Stick System**, leading to improvements in user interface design, sensor sensitivity, and real-time responsiveness. This iterative process of testing and validation has helped ensure that the system is both functional and accessible to its intended users.

Table 8 Validation Survey

|  |
| --- |
| Question and Responses |
| A pie chart with text below  AI-generated content may be incorrect.  Figure 25: Survey Q1 |
| Most users (65%) found the registration and login process very easy, while 35% rated it as easy, showing the app's accessibility for new users. |
| Figure 26: Survey Q2 |
| A majority (55%) said the sensor data (distance, fall, water detection) was always correct, and 40% said it was mostly correct, confirming high sensor reliability. |
| Figure 27: Survey Q3 |
| 45% of users found the app very fast in updating sensor alerts, 20% reported instant updates, while a smaller group noted it was acceptable or slow. |
| Figure 28: Survey Q4 |
| 60% of users confirmed the emergency button and buzzer always worked, while 35% said they mostly worked, indicating strong emergency system performance. |
| Figure 29: Survey Q5 |
| 70% found it very easy to add or manage emergency contacts in the app, highlighting user-friendliness in emergency management. |
| A pie chart with different colored circles  AI-generated content may be incorrect.  Figure 30: Survey Q6 |
| 55% reported that emergency alerts were always sent to contacts, and 40% said they were mostly sent, confirming dependable alert delivery. |
| A pie chart with text on it  AI-generated content may be incorrect.  Figure 31: Survey Q7 |
| 70% rated the app’s usability (clarity, navigation, accessibility) as excellent, while 30% rated it as good, reflecting a highly accessible design. |
| A pie chart with text  AI-generated content may be incorrect.  Figure 32: Survey Q8 |
| 45% found obstacle detection always accurate, while 50% said it was mostly accurate, confirming effective obstacle sensing. |
| A pie chart with text overlay  AI-generated content may be incorrect.  Figure 33: Survey Q9 |
| Fall detection accuracy was evenly split: 50% said it was very accurate, and 50% said it was mostly accurate, showing strong fall detection performance. |
| A pie chart with text and numbers  AI-generated content may be incorrect.  Figure 34: Survey Q10 |
| 70% of users were very satisfied with the system, and 25% were satisfied, indicating a high level of overall user approval. |

## Computing-based Solution/System Design:

### Design:

Designing the Smart Blind Stick was a crucial step in ensuring the device fulfills daily mobility and safety needs of visually challenged people. Combining advanced sensors, a mobile app, and real-time notifications, we carefully crafted an efficient, easily accessible, responsive assistive instrument.

* **Conceptualization and Requirements Gathering:** We began by determining the system's most basic requirements. This covered real-time identification of nearby hazards, sense water or slick surfaces, identify falls, and warn emergency contacts or caregivers. Knowing the daily difficulties experienced by visually impaired people helped us create well defined objectives. These specifications directed to the overall structure of the hardware as well as the mobile application.
* **User Interface (UI) and User Experience (UX) Design:** Accessibility was considered in both the UI and UX design of the iOS app. large letters and simple buttons help users and caregivers to easily grasp sensor values and get alarms from the clean and minimal interface. Simple navigation allowed emergency notifications and status updates to be available rapidly, even in terrible situations.
* **System Architecture and Database Design:** Firebase Real-time Database allows us to link the ESP32-based hardware to the iOS app with reliability. Fast and safe data transfer is ensured by the architecture allowing perfect communication between the sensors and the app. designed to effectively store sensor values, emergency circumstances, and user-related data, the Firebase structure.
* **Prototyping:** Before finalizing development, we created functional prototypes of both the stick and the app. These prototypes allowed us to test how the sensors interacted with Firebase and how the app responded to simulated events like a fall or water detection. Prototyping helped us fine-tune the hardware design and improve app functionality before deployment.
* **Development Environment Setup:** Hardware was built using the Arduino IDE and C language programming to interface the ultrasonic, water, and MPU6050 sensors. From the software standpoint, we created a native iOS app using Xcode with Swift and SwiftUI. Firebase was selected as the backend to ensure consistent, real-time hardware and software synchronizing capability.
* **Design Review and Feedback:** We conducted internal reviews and gathered feedback from potential users and team members. Suggestions regarding sensor accuracy, app layout, and notification timing were used to improve both the hardware response and app usability. These reviews ensured the system was functional, accessible, and aligned with user needs.
* **Documentation:** Throughout the design process, we maintained comprehensive documentation detailing the architecture, sensor configurations, Firebase structure, UI design, and code implementation. This documentation supports team collaboration and serves as a reference for future maintenance or upgrades.

The Smart Blind Stick's design focused on producing a consistent and easily available device combining smart communication with sensor-based awareness. The system was designed to help visually challenged people securely and autonomously by means of well-defined planning and user-oriented choices.

## Graphical User Interface:

The GUI of the Novel Incident Reporting System is meticulously crafted to ensure user-friendly navigation while maximizing the systems advanced safety features. The interface is straightforward, engaging, and supports the core functionality of incident reporting, augmented with real-time location sharing. Here are the specific components of the GUI tailored to user needs:

**Login and Registration Screen:**

* Provides a way via which people might access the system.
* Features: Simple account creation and sign-in choices; additional security measures including password recovery and verification ensure user trust and safety.

**Main Dashboard:**

* Acts as the central hub for users to navigate the app's functionalities.
* Features: Direct access to vehicle status, recent incident reports, and rapid emergency contact alerts so streamlining the user experience for drivers.

**Incident Reporting Interface:**

* Designed to facilitate quick and efficient reporting of road incidents.
* Features: Intuitive forms for incident details, automated sensor data integration, and a one-click submission process, ensuring rapid response and ease of use during urgent situations.

**Real-Time Monitoring Screen:**

* Real-time current location and sensor status of the vehicle are shown on this interactive map.
* Features: Integrates sensor alarms so customers may quickly check the safety of their vehicle and get instant updates during an emergency.

**Incident History and Analysis:**

* Presents a log of past incidents, with detailed breakdowns for review and analysis.
* Features: A clear and organized layout offering in-depth insights into each recorded event, enabling users to review past incidents and sensor performance.

**Emergency Contacts Management:**

* Allows users control their emergency contact list for quick alert in the case of the incident.
* Features: Easy addition and removal of contacts, with quick test alert functionality to ensure the system's reliability in real emergencies.

These elements are used in the Novel Incident Reporting System's GUI to provide a smooth and simple user interface. We want to improve road safety for all drivers by integrating real-time incident reporting with user-friendly design, hence ensuring rapid identification and reporting of any road events or disasters.

### Architectural Pattern:

#### Design Class Diagram

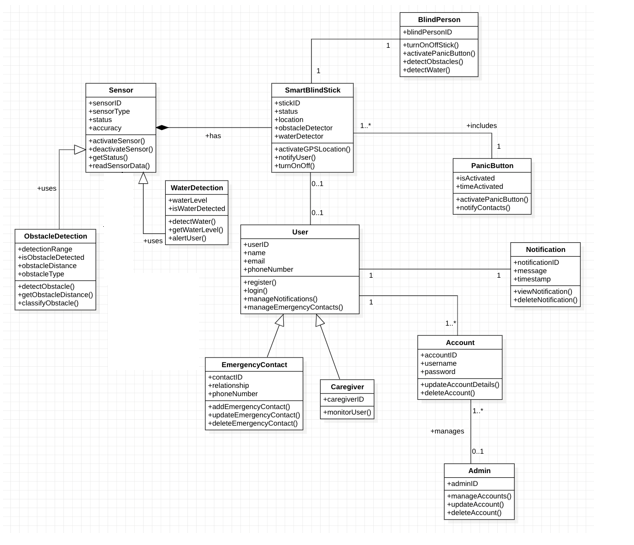
Class diagrams provide a visual representation of the organization and connections between the various parts of a system. There are thirteen classes in our system. Every one of these classes has unique characteristics and functions.

Figure 35 Class diagrams

**Package Diagram**

This module allows users to log in and register for the application. The **Register** feature enables account creation, while th**e Login** feature verifies credentials for secure access. The Controller processes user inputs, and the Model stores user credentials.

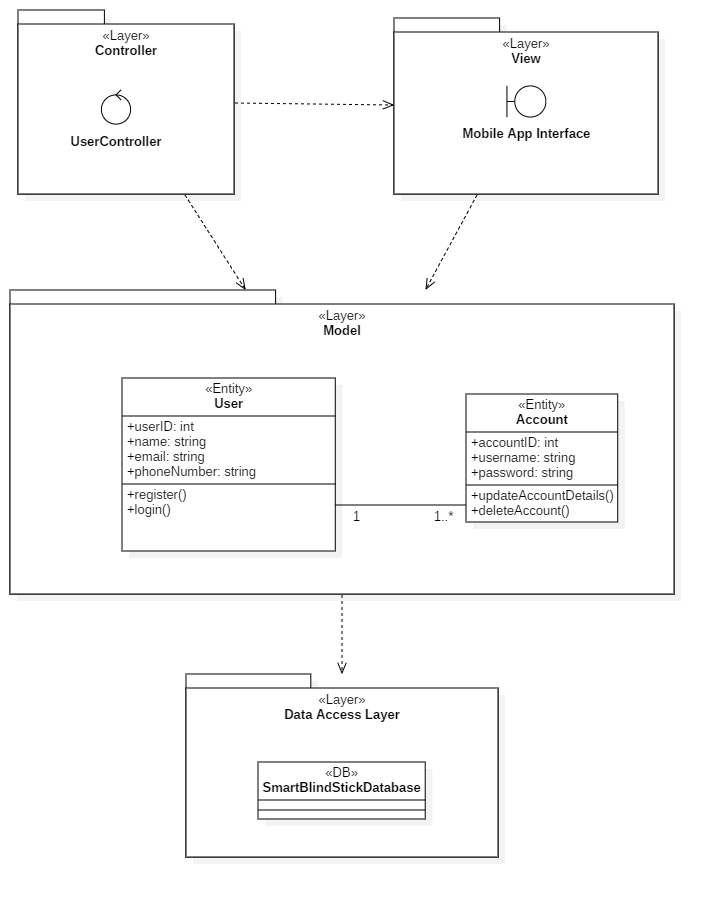


Figure 36 Module 1: User Authentication

This module manages the hardware sensors **(ObstacleDetection, WaterDetection, GPS)** and feedback mechanisms like vibrations and audio alerts. Users can configure sensor settings via the View, while the Controller ensures smooth communication with the Model.

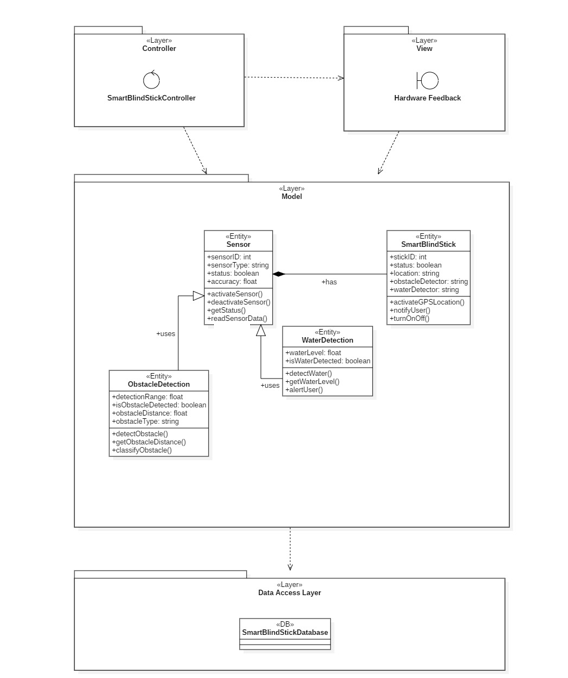


Figure 37 Module 2: Sensor and Feedback Management

This module handles the **PanicButton** feature, triggering alerts in emergencies. It captures the GPS location and sends notifications to emergency contacts. The View displays alert statuses, while the Controller processes panic button actions.

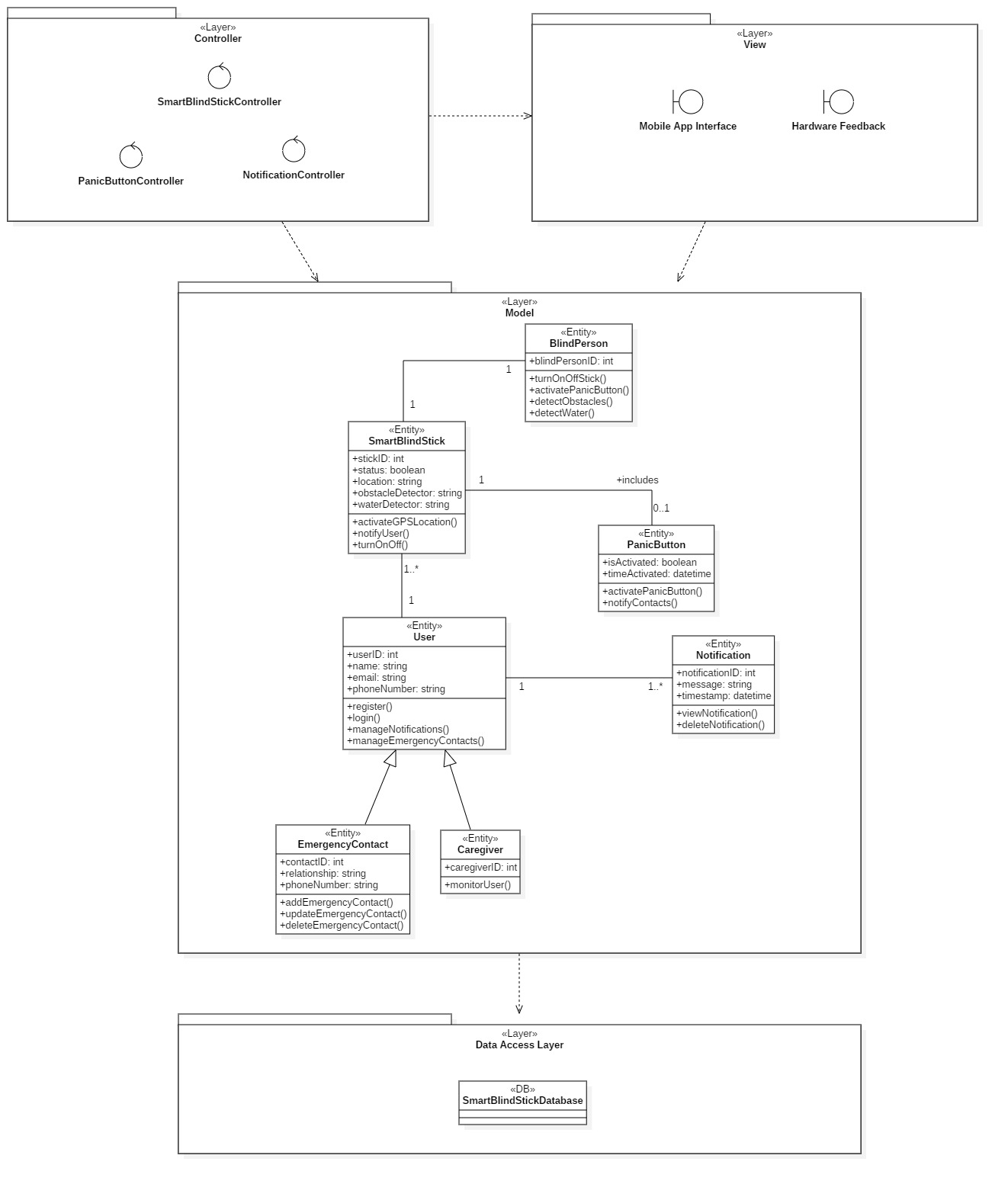


Figure 38 Module 3: Emergency Alert System

Users can customize notification preferences for obstacles, hazards, and emergencies. The Controller updates settings in the Model, and the View provides an interface to manage these options.

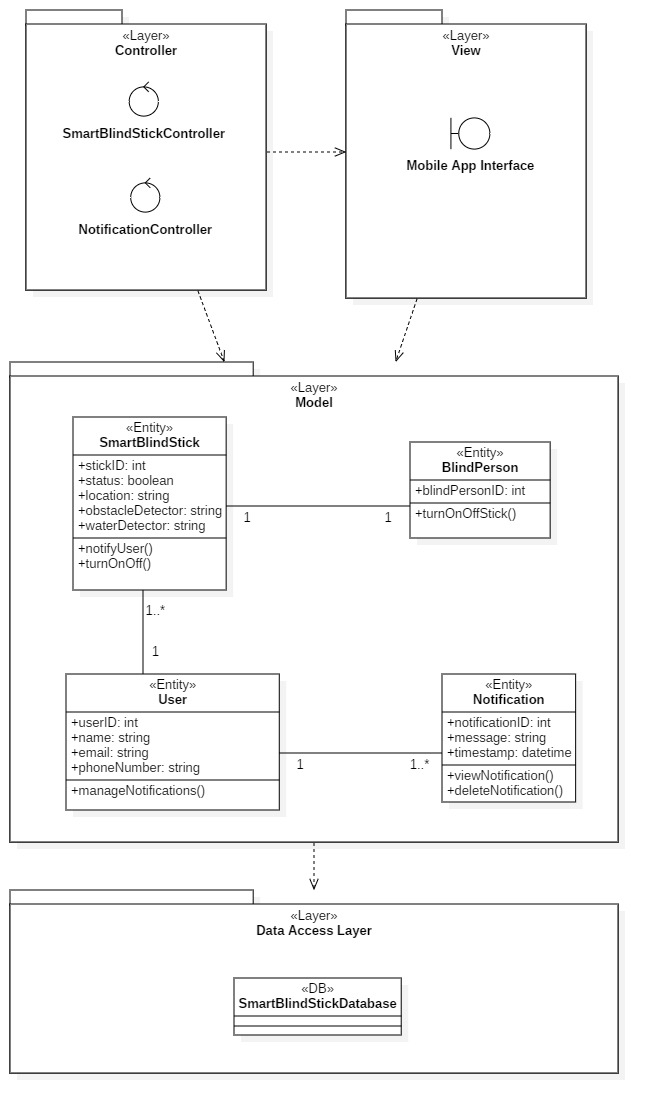


Figure 39 Module 4: Notification Management

This module allows users to update profiles, reset passwords, or delete accounts. The View displays account settings, the Controller handles user actions, and the Model securely manages account data.

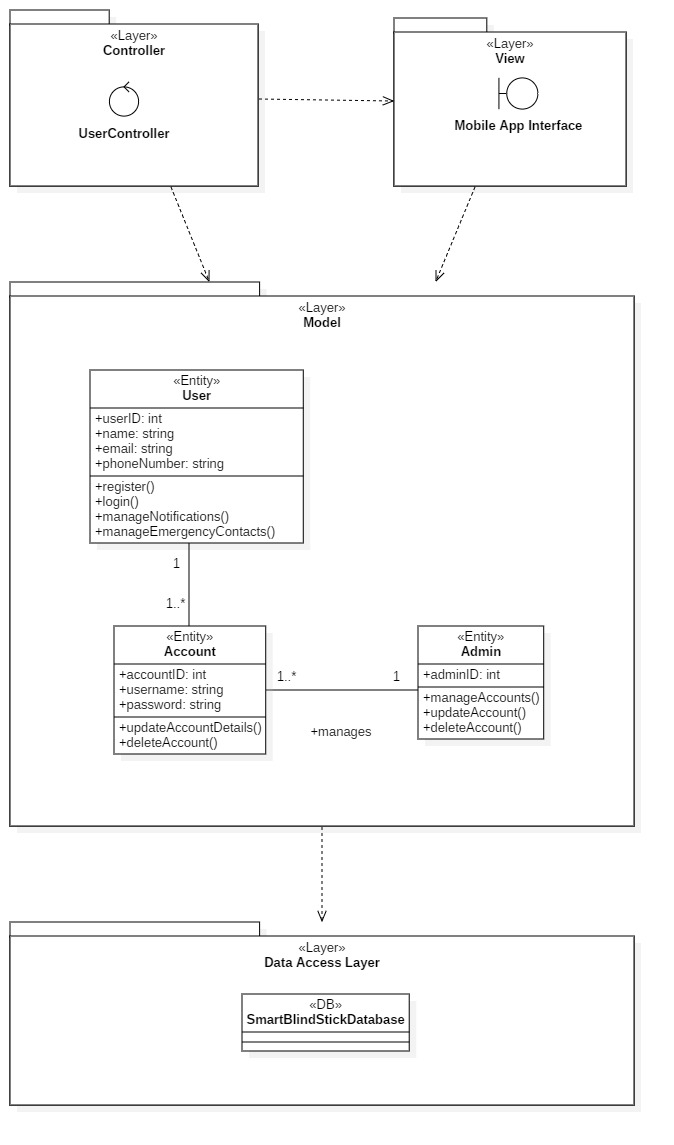


Figure 40 Module 5: User Profile and Account Management

This modular structure ensures parallel development, flexibility, and easy scalability for the **Smart Blind Stick project.**

#### Interaction Diagram

Software engineers and business experts use sequence diagrams, which are visual representations of how operations are carried out in a series of events or messages, to record the requirements for a new system. Because they illustrate how a system event occurs, sequence diagrams are also referred to as event diagrams or event scenarios.

The Actor (User) interacts with the Login Boundary to provide login credentials. These credentials are processed by the User Controller, which validates them against the Account Entity and Database (DB). The system grants or denies access based on validation.

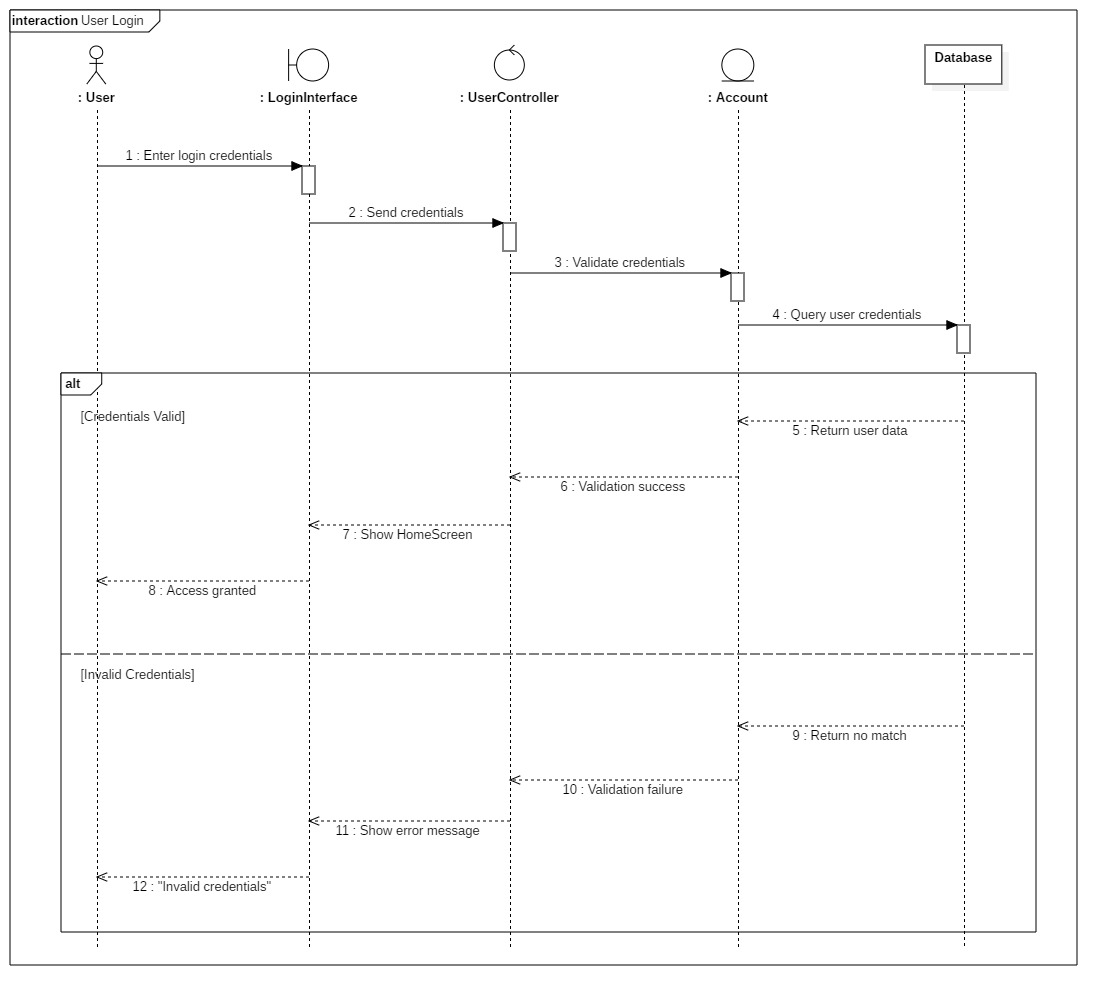


Figure 41 Sequence Diagram 1: User Login

The Actor (User) interacts with the StickFeedback Boundary. When a sensor detects an obstacle, the SmartBlindStickController processes the data from the ObstacleDetection Entity, logs it in the Database (DB), and triggers feedback (vibration or sound).

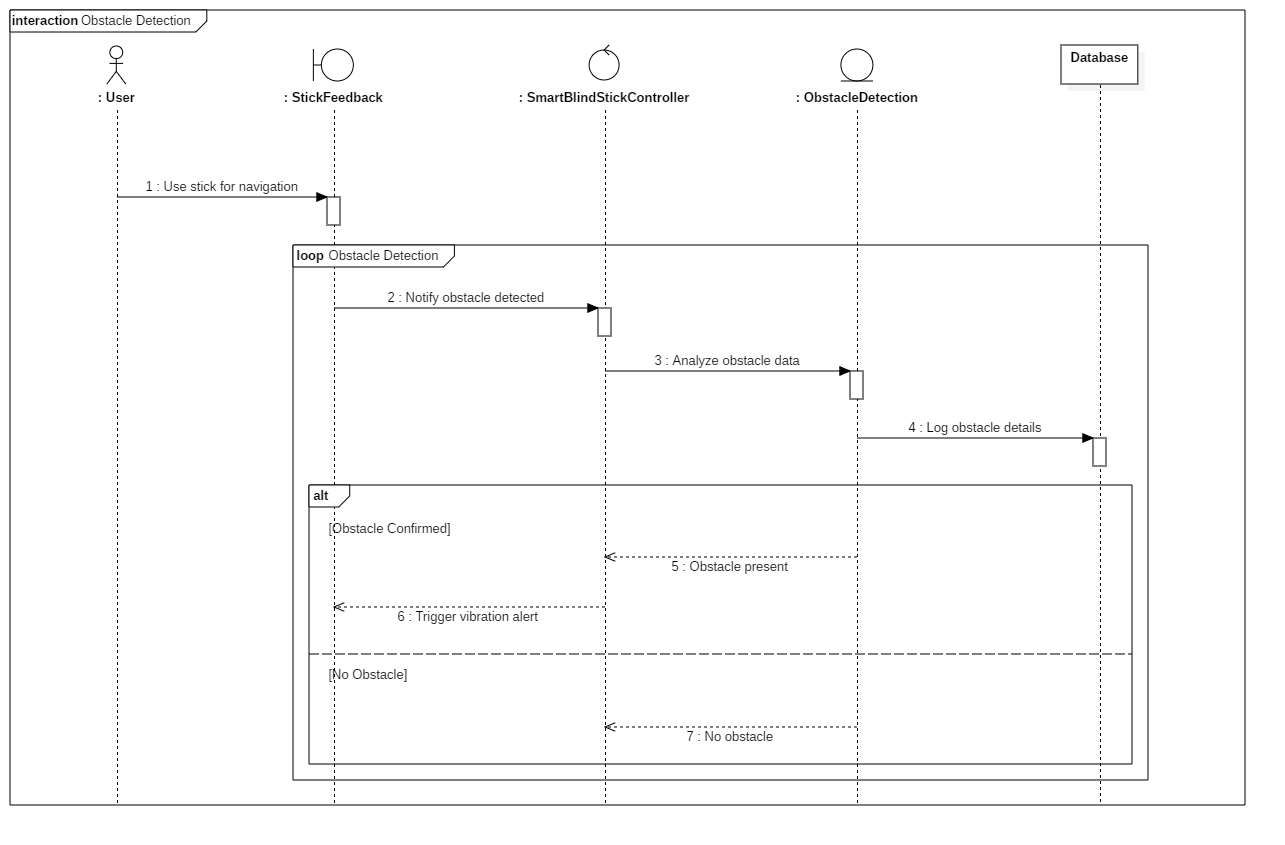


Figure 42 Sequence Diagram 2: Obstacle Detection

When the Actor (User) presses the panic button, the PanicButtonController retrieves GPS data from the GPS Entity, stores the emergency event in the Database (DB), and sends notifications to emergency contacts via the Notification Entity.

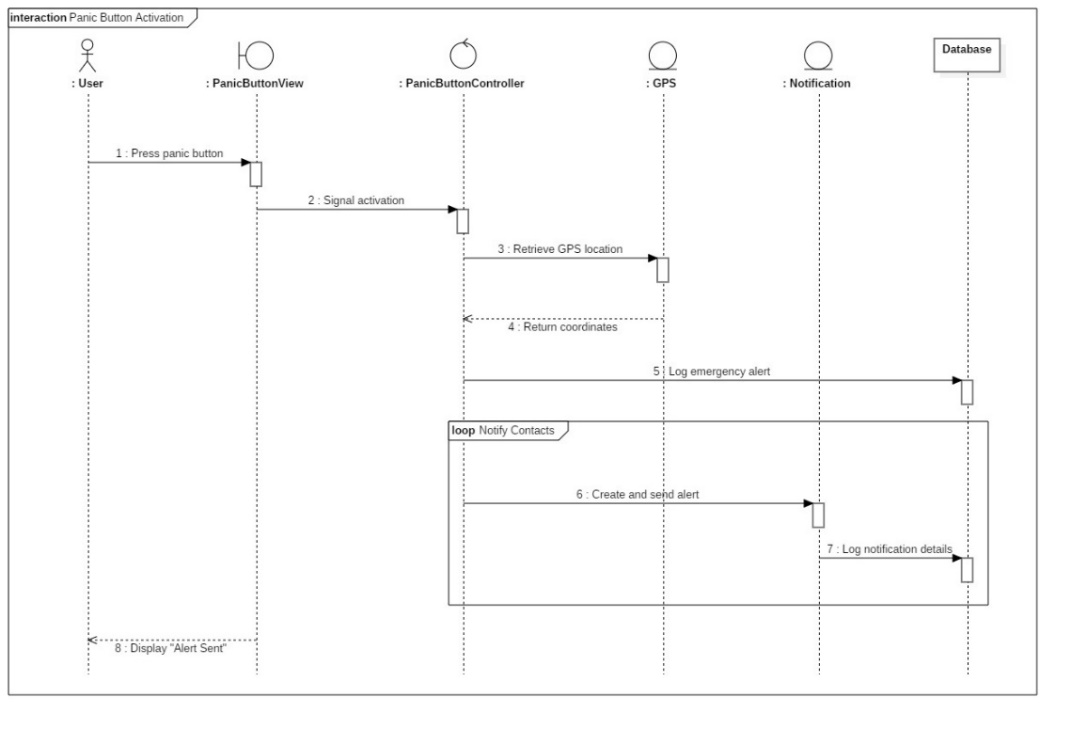


Figure 43 Sequence Diagram 3: Panic Button Activation

The Actor (User) interacts with the Notification Boundary to customize notification settings. The NotificationController processes the request, updates the Notification Entity, and stores the new settings in the Database (DB).

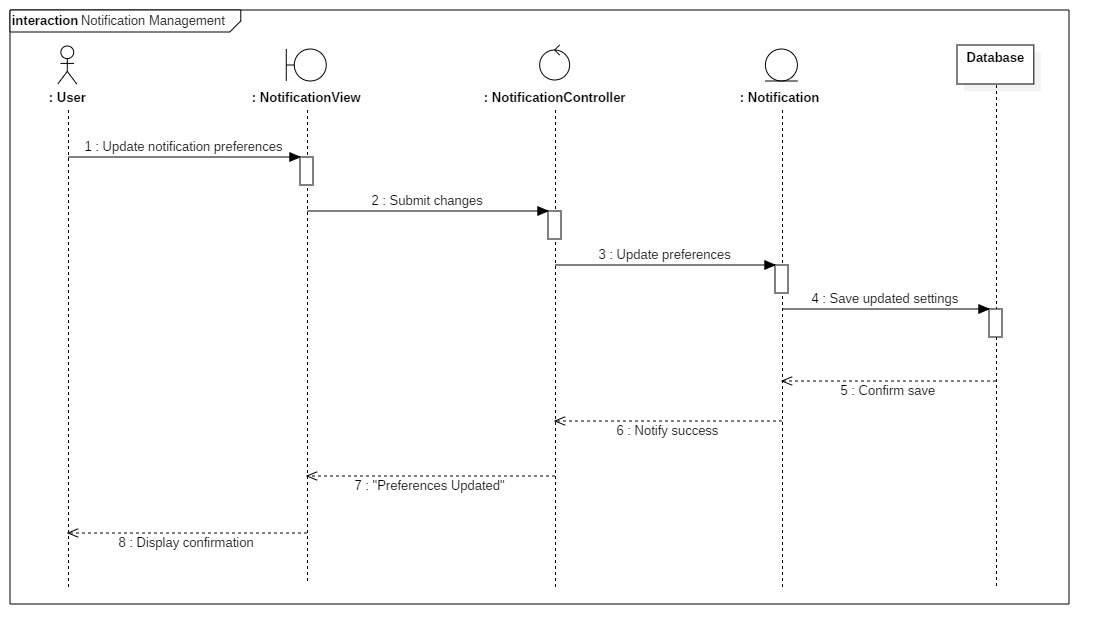


Figure 44 Sequence Diagram 4: Notification Management

The diagram shows how both the User and Admin can manage account details. The Admin can access and modify accounts through the Admin Interface, while the User uses the Account Management Boundary. Both interact with the UserController, which updates the account in the Account Entity and stores the changes in the Database (DB).

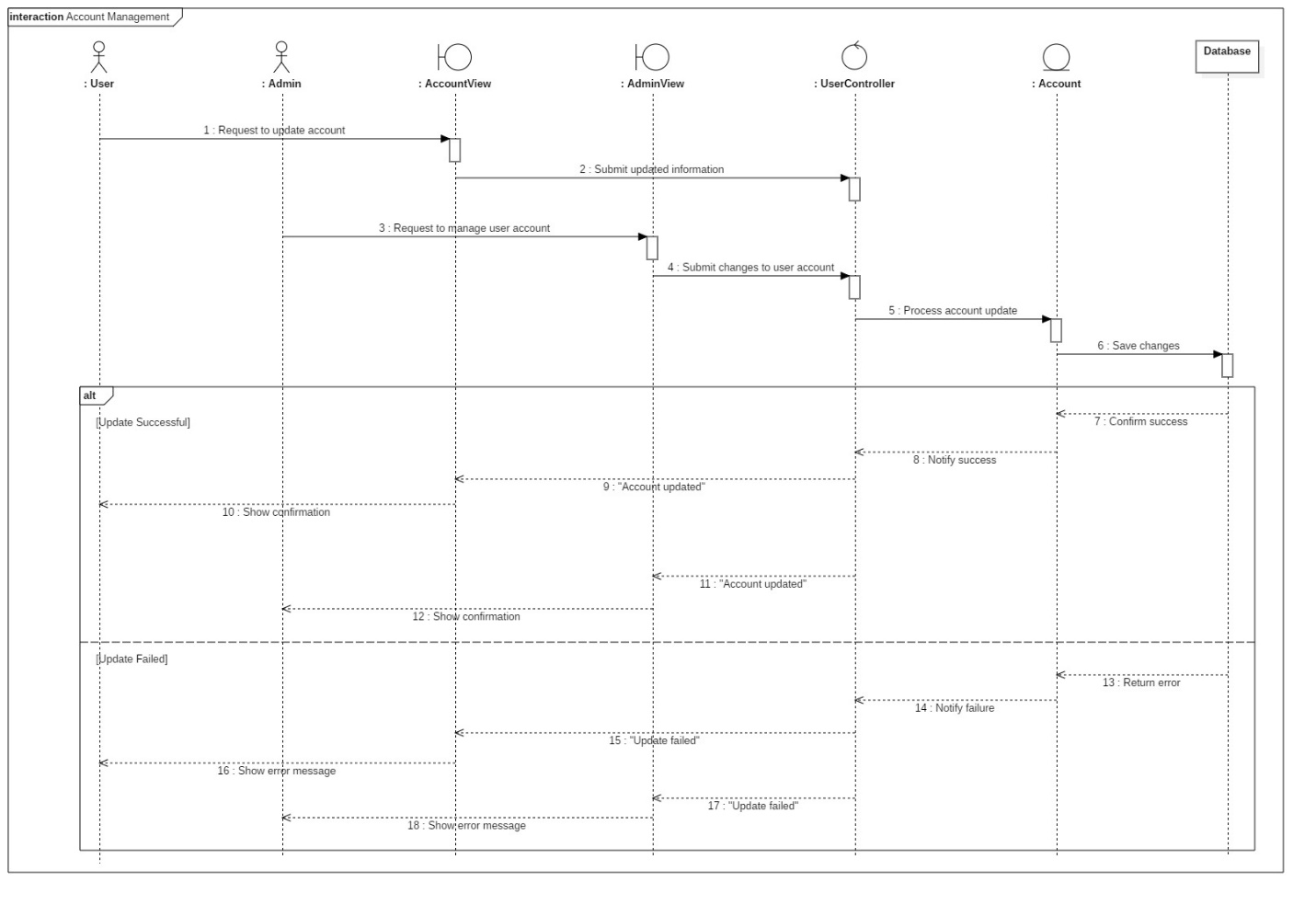


Figure 45 Sequence Diagram 5: Account Management

### Application Interface

|  |  |
| --- | --- |
| Descriptions | Screen |
| This screen displays real-time sensor readings and alerts users if sensor values exceed a safe threshold. It also provides access to the SOS Emergency button for immediate assistance. | Figure 46: Dashboard Screen |
| This screen shows detailed sensor data, including acceleration, gyroscope readings, distance measurements, and water level, allowing users to monitor environmental conditions. | Figure 47: Sensor Monitor Screen |
| This screen allows users to manage their personal and medical information, including name, email, phone number, and blood group, with options to edit details or log out. | Figure 48: Profile Screen |
| This screen appears when critical sensor values are detected, prompting the user to call emergency contacts for immediate assistance. | Figure 49: Emergency Alert Screen |
| Entering their full name, phone number, and relationship allows users of this screen to add emergency contacts, therefore ensuring rapid alerts in need to an emergency occur. | Figure 50: Add Emergency Contact Screen |
| This screen allows the user add or delete contacts, thus making sure that alarms get to the correct persons in case of an emergency. | Figure 51: Emergency Contacts Screen |
| This screen allows new users to register by entering their full name, email, password, and confirming their password before signing up. | C:\Users\Unknown\AppData\Local\Packages\5319275A.WhatsAppDesktop_cv1g1gvanyjgm\TempState\85AAD03A5D5F909A016FDF0D47568E49\WhatsApp Image 2025-03-21 at 14.57.18_d661221a.jpg  Figure 52: Create Account Screen |
| This screen enables registered users to log in by entering their email and password, with an option to sign up for new users who don’t have an account. | C:\Users\Unknown\AppData\Local\Packages\5319275A.WhatsAppDesktop_cv1g1gvanyjgm\TempState\6A639C52538FD7BB3E1186724AFBE9B2\WhatsApp Image 2025-03-21 at 14.57.28_b763f2cc.jpg  Figure 53: Login Screen |

## Implementation:

### Application Implementation:

|  |
| --- |
| Application Codes |
| C:\Users\Unknown\AppData\Local\Packages\5319275A.WhatsAppDesktop_cv1g1gvanyjgm\TempState\84E84640ED1B31F3046CA6B905542217\WhatsApp Image 2025-03-21 at 14.57.18_5baba06e.jpg  Figure 54: SignupView.swift |
| This screen lets new users create an account by entering their full name, email, and password. It also saves the user’s basic profile info to Firebase. |
| C:\Users\Unknown\AppData\Local\Packages\5319275A.WhatsAppDesktop_cv1g1gvanyjgm\TempState\A30B36CD9C2E0343AE527B263A7B0ABB\WhatsApp Image 2025-03-21 at 14.57.18_b9381c36.jpg  Figure 55: AuthView.swift |
| This view manages switching between the login and signup screens. It acts as the starting point when users open the app. |
| C:\Users\Unknown\AppData\Local\Packages\5319275A.WhatsAppDesktop_cv1g1gvanyjgm\TempState\8E3E36041282159455001DFA35E02AA8\WhatsApp Image 2025-03-21 at 14.57.19_8d6dfd45.jpg  Figure 56: ThresholdSettingsView.swift |
| This view lets users adjust and save threshold values for acceleration, gyroscope, distance, and water level. It updates settings in Firebase. |
| C:\Users\Unknown\AppData\Local\Packages\5319275A.WhatsAppDesktop_cv1g1gvanyjgm\TempState\B8FD187B3F6B41D95BBC6C831075F367\WhatsApp Image 2025-03-21 at 14.57.19_0032884f.jpg  Figure 57: EmergencyContactsView.swift |
| This file allows users to add, view, and delete emergency contacts. It also enables calling a contact directly from the app during an emergency. |
| C:\Users\Unknown\AppData\Local\Packages\5319275A.WhatsAppDesktop_cv1g1gvanyjgm\TempState\9DCC7F7C341F77A6D814E6F1E0FC6DF3\WhatsApp Image 2025-03-21 at 14.57.20_ff51e7c9.jpg  Figure 58: HomeView.swift |
| This screen shows a welcome message, sensor alerts, and a live summary of sensor data. It also has an SOS button to trigger emergency actions. |
| C:\Users\Unknown\AppData\Local\Packages\5319275A.WhatsAppDesktop_cv1g1gvanyjgm\TempState\E1F5B6486CFC532A00CBC1EF67B40358\WhatsApp Image 2025-03-21 at 14.57.20_1e661320.jpg  Figure 59: LoginView.swift |
| This screen allows users to log in using their email and password. If the login is successful, it stores the user ID and updates the login status. |
| C:\Users\Unknown\AppData\Local\Packages\5319275A.WhatsAppDesktop_cv1g1gvanyjgm\TempState\A6CFF0397725A493BA4047565693F84A\WhatsApp Image 2025-03-21 at 14.57.21_d32acf6c.jpg  Figure 60: ProfileView.swift |
| This screen shows user information like name, email, phone number, and blood group. Users can edit or update their profile and also log out. |
| C:\Users\Unknown\AppData\Local\Packages\5319275A.WhatsAppDesktop_cv1g1gvanyjgm\TempState\EBFEBC42D8F8DD67DA25A2DCAB4AC8FF\WhatsApp Image 2025-03-21 at 14.57.21_37fdc263.jpg  Figure 61: SensorSummaryView.swift |
| This file displays current sensor readings such as acceleration, gyroscope, distance, and water level. It highlights values that exceed safe thresholds. |
| C:\Users\Unknown\AppData\Local\Packages\5319275A.WhatsAppDesktop_cv1g1gvanyjgm\TempState\36F9C2AE6EC1D54417FA9C6BA46BA9BD\WhatsApp Image 2025-03-21 at 14.57.28_17b01c68.jpg  Figure 62: SensorsView.swift |
| This screen shows detailed sensor data and allows users to change alert thresholds. It also listens to live sensor updates from Firebase. |

### Hardware’s Coding Implementation:

In the development phase of the Smart Blind Stick with Real-time Sensor Data the Stick side includes several hardware side controller code:

Table 9 Hardware’s Coding Implementation

|  |  |
| --- | --- |
| Explanation | Hardware Codes |
| This part includes the necessary libraries and headers for Arduino, Wi-Fi (choosing the correct library for ESP32 or ESP8266), Firebase, I2C, and the MPU6050 sensor. It also defines constants for Wi-Fi credentials, Firebase API key and URL, as well as global objects and pin assignments for sensors and peripherals. |  |
| This section declares global Firebase objects for data, authentication, and configuration. It also initializes timing variables and a flag for Firebase signup status. Pin definitions for the ultrasonic sensor, buzzer, button, and water level sensor are set here, along with the MPU6050 sensor instance. |  |
| The setup() function initializes serial communication and I2C for the MPU6050. It configures the hardware pins, connects to Wi-Fi (printing the connection progress and IP address), sets up Firebase with the API key and URL, and attempts Firebase authentication. Finally, it initializes and tests the MPU6050 sensor connection. |  |
| Within the loop() function, the code first checks if the button is pressed to activate the buzzer briefly. It then triggers the ultrasonic sensor to measure distance by sending a pulse and calculating the distance from the echo duration. It also reads the water level and gets acceleration and gyroscope data from the MPU6050, printing all sensor values to the serial monitor. |  |
| This final segment of the loop checks if 15 seconds have passed, and if so, it sends the sensor data (distance, water level, acceleration, and gyroscope readings) to Firebase’s Real-time Database. It prints confirmation or error messages for each data upload and then waits briefly before repeating the cycle. |  |

### Database Implementation

Firebase is an essential component backend solution that ensures real-time data updates, safe authentication, and organized data storage in the evolution of the Smart Blind Stick. Firebase is absolutely crucial for managing user authentication, sensor data management, and emergency alarms, hence the system is quite responsive and efficient.

**Firebase Authentication Panel:** Verification of user identity and safe login handling falls to the Firebase Authentication Panel. It lets users log in with their email and password, thus stopping unwanted access. Maintaining a list of registered users together with their individual user IDs and login timestamps, the system ensures that only authorized users may access crucial characteristics.

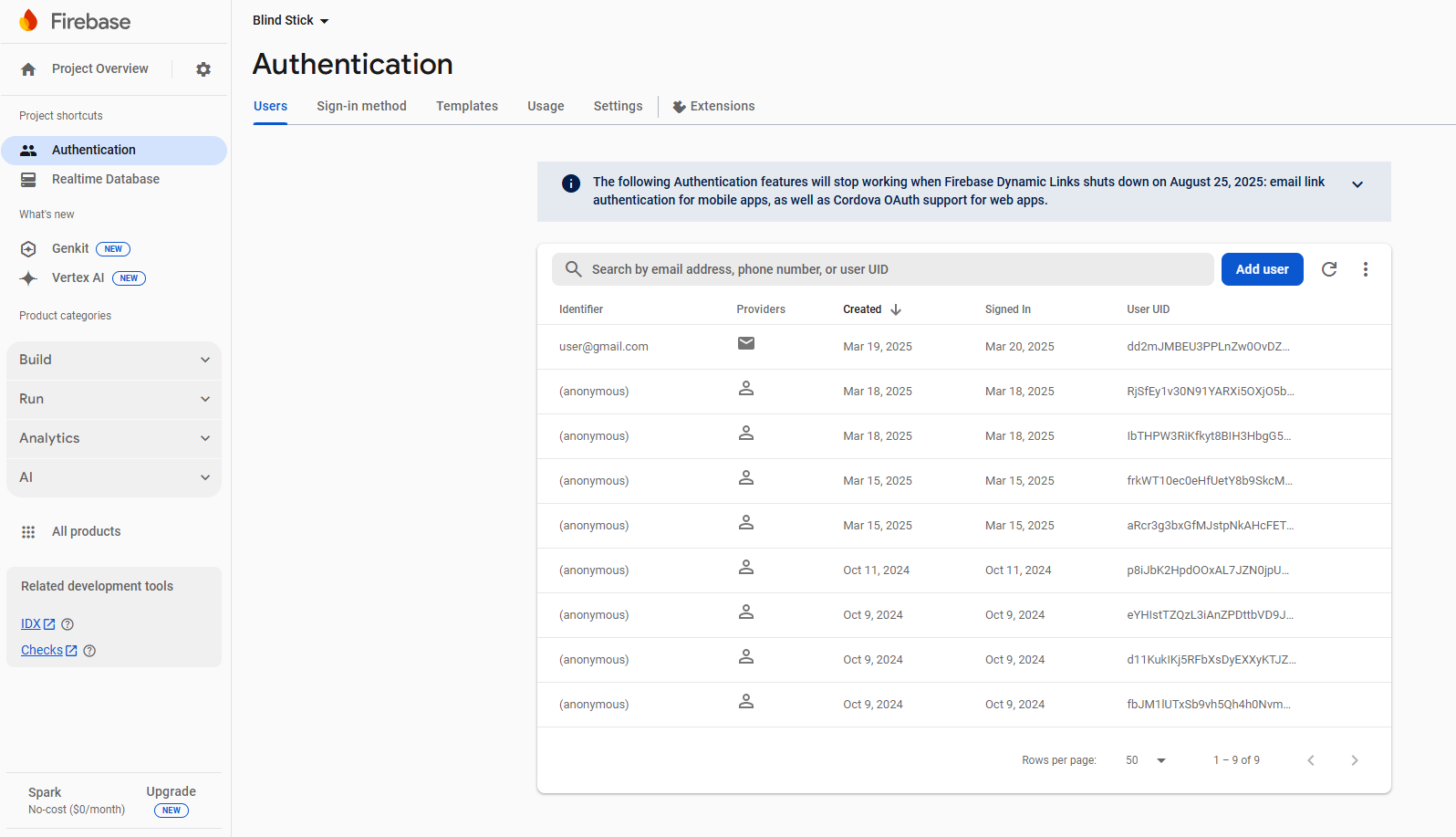


Figure 63: Firebase Authentication

**Firebase Real-time Database:** The Real-time Database handles live sensor data updates from the blind stick. It records and updates sensor readings constantly: acceleration, gyroscope measurements, distance detection, water level, and emergency warnings. This ensures rapid execution of all important data, allowing real-time monitoring and quick reaction to emergencies.

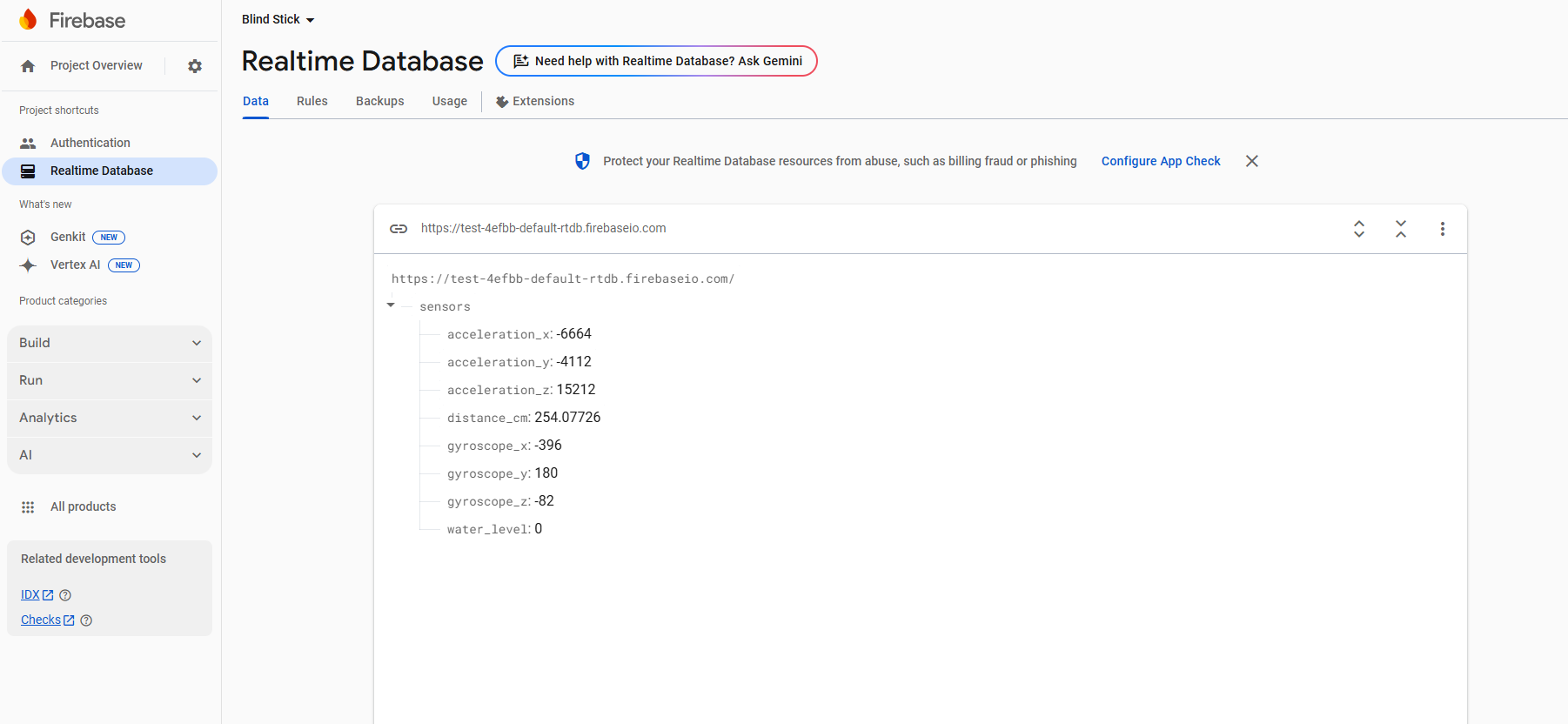


Figure 64: Realtime Database

**Firebase Cloud Firestore Database:** Structured user profile data—personal information, emergency contacts, sensor threshold settings—is stored in the Firebase Cloud Firestore Database. By way of safe data retrieval and querying made possible by this database, caretakers or managers may effectively evaluate past events and control user settings.

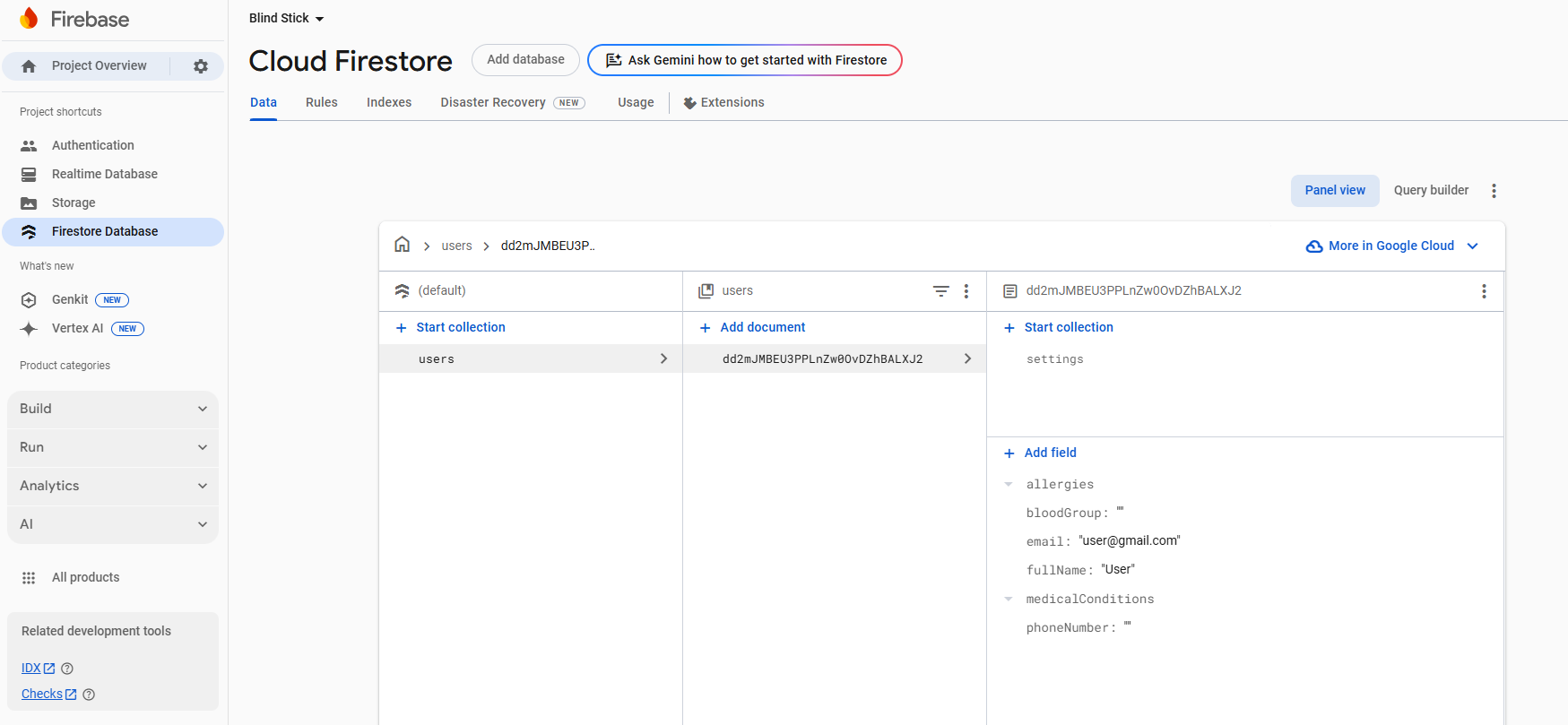


Figure 65: Cloud Firestore

The Smart Blind Stick ensures safe user access, real-time sensor updates, and well-organized data storage by leveraging Firebase Authentication, Real-time Database, and Firestore, so making the system quite responsive, effective, and user-friendly.

## Deployment

In the deployment phase of the Smart Blind Stick System, we integrated modern development tools and frameworks to ensure a responsive, secure, and real-time platform that enhances the mobility and safety of visually impaired individuals.

* **Arduino IDE** was used as the primary development environment for programming the ESP32 microcontroller. It provided efficient libraries and real-time debugging support for sensor integration and Wi-Fi connectivity.
* **ESP32 Microcontroller** was selected for its low power consumption, strong Wi-Fi capabilities, and compatibility with multiple sensor modules (Ultrasonic, MPU6050, Water sensor).
* **Xcode** served as the development environment for building and testing the iOS mobile application. It offered seamless Swift and SwiftUI integration, ensuring smooth frontend development and device simulation.
* **Swift and SwiftUI** were used for the mobile app's development, offering a clean, accessible, and responsive user interface tailored for visually impaired users.
* **Firebase Real-time Database and Authentication** handled backend services, including secure user account management, real-time sensor data updates, and emergency contact storage.
* **Firebase Cloud Messaging (FCM)** was integrated to ensure instant emergency notifications to caregivers when critical hazards or emergencies were detected.
* **Security measures** included session-based authentication via Firebase, encrypted data storage, and secure management of user and emergency contact information.

The deployment process involved connecting the ESP32 hardware to Wi-Fi, synchronizing real-time sensor data with Firebase, and launching the iOS mobile app via Xcode simulators and real devices for live testing. The system was tested under real-world conditions to verify full functionality from hardware detection to emergency alert generation.

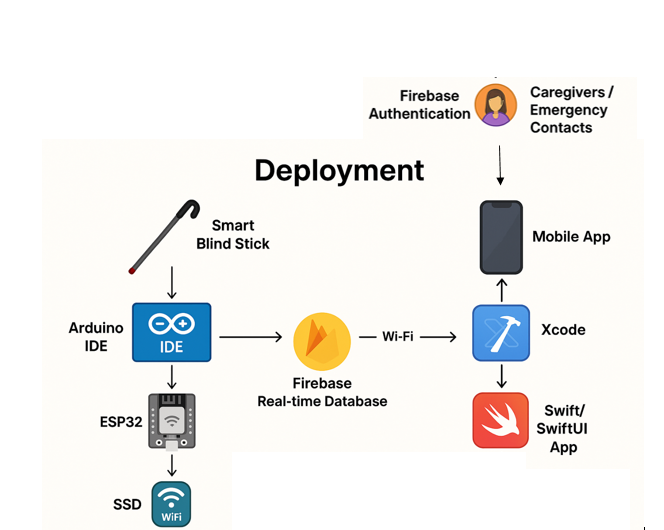


Figure 66: System's Deployment

**Final testing confirmed:**

* Accurate sensor data collection and real-time transmission to the mobile app.
* Functional emergency button activation triggering immediate alerts and buzzer alarms.
* Secure registration, login, and session management via Firebase Authentication.
* Real-time updates of sensor alerts on the mobile dashboard without page reloads.
* Proper storage and retrieval of emergency contact data within the Firebase structure.

With all tools and technologies successfully integrated, the Smart Blind Stick System operates as a stable, intelligent, and secure platform, delivering real-time hazard detection and emergency support for visually impaired individuals.

## Conclusion

he Smart Blind Stick System project marks a significant step forward in the integration of assistive technology with real-time mobility support for visually impaired individuals. Designed to address a clear societal need, this system combines hardware and software innovation to create a user-centric solution that promotes safety, independence, and confidence in navigation.

Throughout the development process, we leveraged a variety of advanced technologies such as the ESP32 microcontroller, ultrasonic and water sensors, MPU6050 accelerometer, and Firebase Real-time Database—to build a system capable of detecting and responding to environmental hazards. Simultaneously, the iOS mobile application, built with SwiftUI, offered a streamlined, high-contrast, and accessible interface designed specifically for ease of use by visually impaired users. This interface enabled real-time feedback, secure user authentication, and instant communication with emergency contacts.

The development journey was marked by notable technical challenges. Integrating multiple sensors into a cohesive, responsive unit required meticulous calibration and synchronization. Maintaining stable wireless connectivity across devices and handling real-time data transmission posed hurdles, especially under fluctuating network conditions. Yet, through testing, optimization, and design iteration, these issues were systematically resolved.

User-centered design remained at the core of our approach. Feedback gathered through surveys and live user trials informed several improvements, such as enhancing sensor alert timing and refining the emergency response mechanism. Functionalities like the SOS button, buzzer alarm, and fall detection were rigorously tested to ensure rapid and reliable operation under diverse scenarios.

Moreover, this project reinforced the value of Agile-based iterative development. We were able to adapt quickly to evolving requirements, continuously refine our features, and focus on quality assurance through repeated cycles of testing and validation. Ultimately, the Smart Blind Stick stands as more than a technical achievement it is a practical and impactful tool that offers greater autonomy and peace of mind to visually impaired individuals and their families. It underscores the power of thoughtful innovation, inclusive design, and persistent engineering to solve real-world challenges and improve lives.

# Chapter 5: Conclusion:

In this project, we addressed the problem of limited mobility and safety faced by visually impaired individuals. Our main objective was to design a Smart Blind Stick system that could detect obstacles, water, and falls in real-time, and also provide emergency alerts. To solve this, we created a system that connects smart hardware with a user-friendly iOS mobile app.

We used different tools and technologies to build this solution. The ESP32 microcontroller was used to connect all the sensors, including an ultrasonic sensor for obstacle detection, MPU6050 for fall detection, and a water sensor. These sensors collect real-time data from the environment. For software development, we used the Arduino IDE to program the ESP32. The mobile application was built using Xcode with Swift and SwiftUI, which made it easy to create a clean and accessible interface for visually impaired users.

We also used Firebase Realtime Database and Authentication to manage user data and share sensor readings between the hardware and the mobile app instantly. This made the system fast, secure, and reliable. With the help of these tools and technologies, we successfully developed a smart, affordable, and helpful solution that increases independence and safety for visually impaired people.

**Team Roles and Reflections**

* Mariam Hasan Alqahtani – Team Leader & System Designer

Mariam led the team by organizing meetings, assigning tasks, and ensuring deadlines were met. She also contributed significantly to the system design, especially in planning the integration between hardware and software. Reflecting on this project, Mariam learned how leadership and time management are crucial for keeping the project on track and guiding the team effectively through challenges.

* Yasmeen Ali Almessabi – Hardware Integration Specialist

Yasmeen was responsible for the hardware components, including the ESP32, sensors, and their configurations. She worked on connecting and testing the sensors to ensure accurate data collection. Through this experience, Yasmeen improved her practical skills in embedded systems and understood the importance of precision when working with real-time hardware data.

* Kana Saleh – Mobile Application Developer

Kana developed the iOS mobile application using Swift and SwiftUI. She ensured the app was user-friendly, accessible, and connected well with Firebase and the ESP32. This project helped Kana improve her mobile development skills, especially in building accessible interfaces and managing real-time data communication.

* Khulood Qasem – Documentation & Testing Lead

Khulood managed project documentation, including reports and user feedback surveys. She also led the testing phase, ensuring all features worked correctly and met user needs. Through her role, Khulood enhanced her attention to detail, learned the importance of thorough testing, and recognized how documentation supports smooth project development.

In the future, we plan to enhance the Smart Blind Stick System by adding more advanced features such as voice assistance for better user interaction, vibration alerts for quiet environments, and AI-based object recognition to identify specific obstacles like stairs or vehicles. We also aim to improve battery life, optimize the mobile app for Android users, and introduce offline functionality for areas with limited internet access. These improvements will make the system more reliable, accessible, and helpful for visually impaired individuals in diverse environments.

# Reference:

Mahdy, S. J. I. J. o. I. and S. Development (2025). "Smart Blind Stick using Mobile Application and Bluetooth Module." **6**(1): 43-53.

Prema, S., et al. (2022). Smart stick using ultrasonic sensors for visually impaired. Advances in Parallel Computing Algorithms, Tools and Paradigms, IOS Press**:** 436-442.

Rahman, M. W., et al. (2021). "The architectural design of smart blind assistant using IoT with deep learning paradigm." **13**: 100344.

Ramachandran, R., et al. (2018). "Iot Sensors and Mobile Applications for Life Style Management of Visually Challenged Persons." **20**(5): 77-84.

Ramesh, G., et al. (2022). "Development of e-stick for blind persons using IoT." **10**(2).

Sanjorjo, J., et al. (2024). Design and Development of a Smart Blind Stick for Enhanced Mobility and Independence of Visually Impaired Individuals. 2024 4th Asian Conference on Innovation in Technology (ASIANCON), IEEE.