



Senior Design Project: Cognitive Assistance with LiDAR Localization (C-ALL)

Project Concept Proposal

Ahmad Shah, Neeti Mistry, Sara Gaber &
Sohan Chatterjee

Team Members and Roles



Neeti

Documentation Manager + User
Advocate + Risk Management



Sara

Use Cases + Front-End Lead



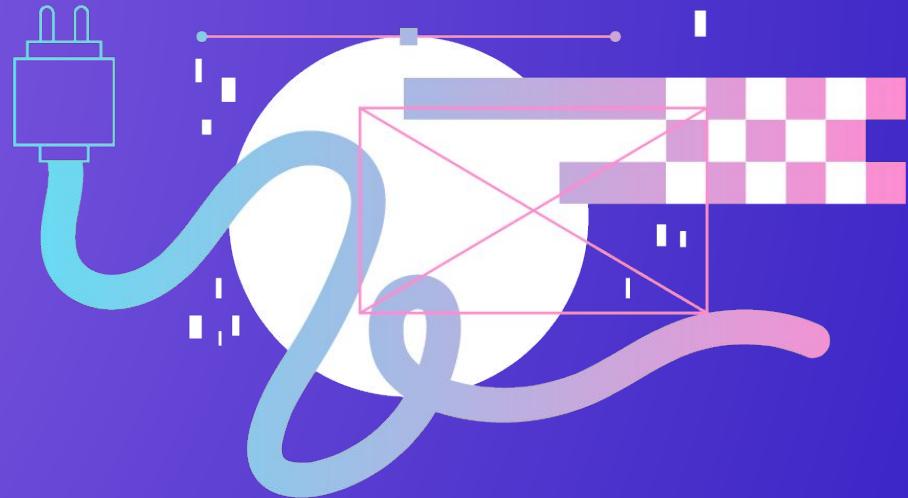
Ahmad

Development + Hardware Lead



Sohan

Test Lead + Buildmeister



Project Description

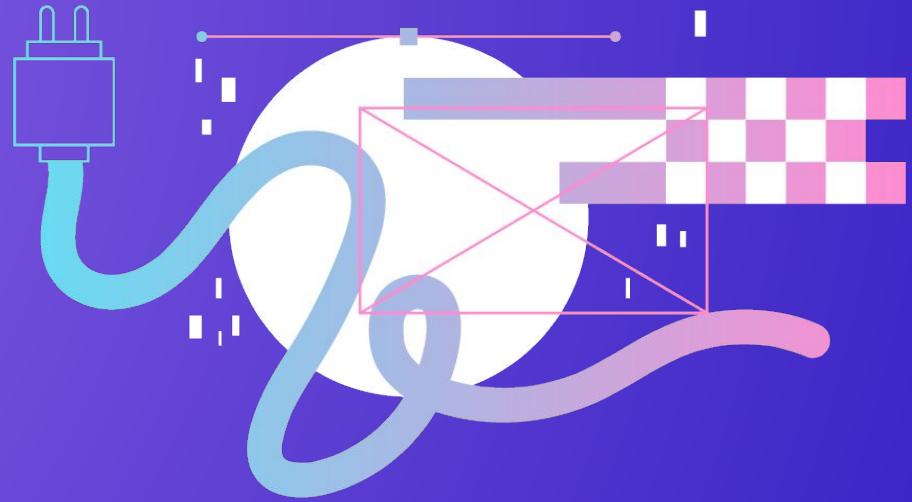
Project details summary and brief overview
from the last demo

Project Description

- **Objective:** Aid visually impaired individuals with an innovative navigation tool
 - C-ALL (Cognitive Assistance with LiDAR Localization)
 - Provide an alternative to current tools
- **Key Components:**
 - LiDAR Sensor: Integrated within a smartphone to map the environment in 3D
 - Physical Device: Provides navigation guidance through tactile feedback
 - Mobile Application: Processes data and facilitates user interaction

Project Description

- **Technology Utilized:**
 - 3D Point Cloud Mapping: Converts raw LiDAR data into simplified 3D objects
 - Simultaneous Localization and Mapping (SLAM): Determines the user's position in real-time
- **Final Product:**
 - Navigation Glove: Features a moving ball that indicates the direction to navigate, with adjustable sensitivity for obstacle detection



Key Stakeholders and Motivation

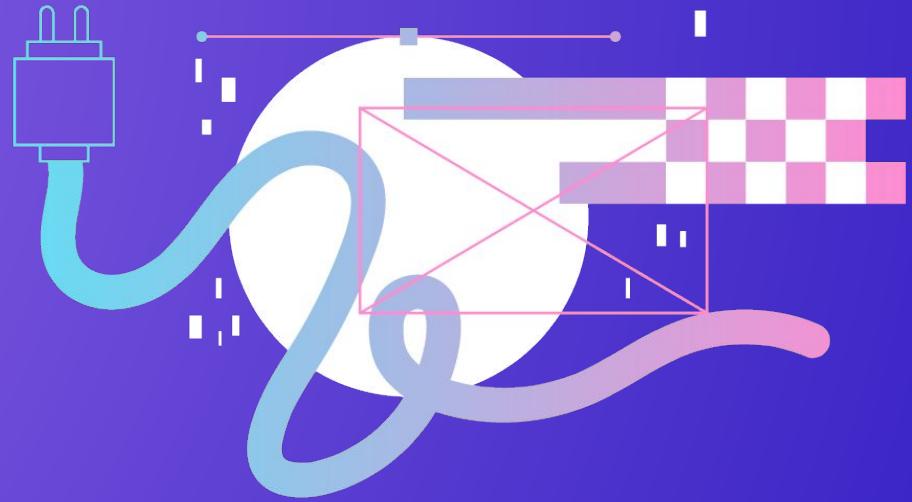
Our stakeholders and motivation behind the project

Key Stakeholders

- **Visually Impaired Clients:** Primary users
 - Their feedback shapes product design and functionality
 - Understand their specific needs, preferences, and challenges, ensuring that our project effectively addresses their requirements
- **Nonprofit Organizations:** Resources and support for reaching target user base
 - Focus on accessibility and inclusivity
- **Insurance Companies and Healthcare:**
 - Potential stakeholders in the financial support and accessibility of our product
 - Understanding their policies can help us position our product effectively in the market
- **Device Manufacturers:** Partnerships for hardware development and compatibility with existing devices
- **Sponsors:** Stevens Institute of Technology provides funding, resources, and mentorship
- **Regulators:** Ensure compliance with accessibility standards and safety regulations

Project Motivation

- **Enhanced Mobility Solutions:** Addressing the critical need for improved mobility tools for visually impaired individuals
- **Limitations of Current Technologies:** Existing assistive devices often depend on limited sensory input or are overwhelming, highlighting the need for better user experience, accuracy, and real-time feedback
- **Bridging the Technological Gap:** Our goal is to combine various technologies to empower visually impaired individuals, enabling them to navigate their environments with increased confidence, independence, and safety
- **Societal Impact:** The potential to make a meaningful difference in the lives of visually impaired individuals inspires our commitment to this project and innovation in the assistive technology field

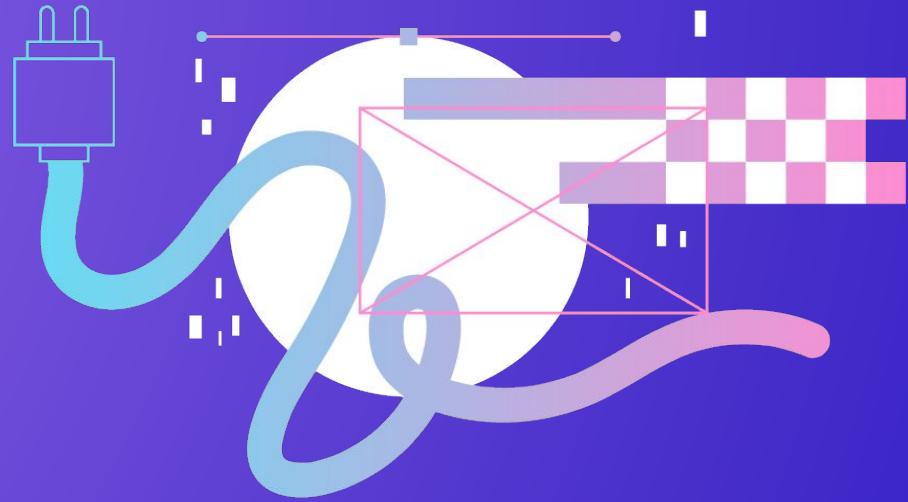


Agile Process

Planning to follow the Agile Unified Process

Agile Process

- **Overview of Agile Unified Process**
 - Framework: We are following the Agile Unified Process (AUP) for our project
 - Emphasizes iterative development, stakeholder collaboration, and adaptive planning
- **Currently** in “elaboration” period where stakeholders are determined, requirements are to be defined, and related use cases diagrams and specifications are created
 - Initial designs of hardware prototype have been rendered and printed
 - Software platforms are in the set-up process and initial software architecture plans have been identified



Development Plan Overview

Brief overview of development plan from the last demo

Development Plan Overview

Phase 1: Basic Obstacle Avoidance

- Glove-Mounted Device: Develop using Arduino and two servos for directional control
- iOS Application Development: Create a basic app in Swift and Xcode
- Obstacle Avoidance: Implement straight path maintenance while avoiding obstacles
- Data Collection and Processing: Use ARKit for a 3D depth map
 - Process depth data on a desktop using Python
- Communication: Relay instructions from desktop to app
 - Use Bluetooth for command transfer to the glove.
- Heading Maintenance: Utilize CoreLocation compass data to maintain direction

Development Plan Overview

Phase 2: Navigation System Implementation

- Destination Input: Allow users to enter desired locations in the app
- Route Mapping: Compute routes and establish compass headings
- Guidance System: Guide users along the path and adjust for obstacles

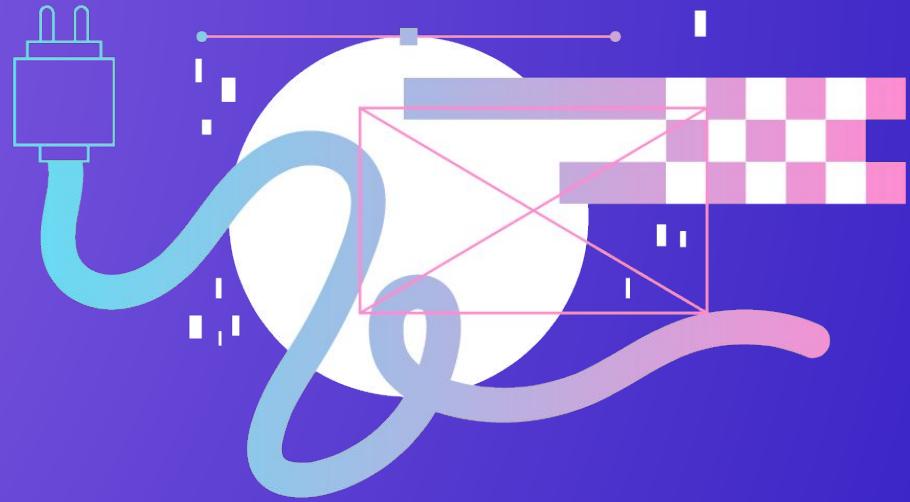
Phase 3: Accessibility Enhancements and Hardware Update

- App Accessibility: Improve ease of use for visually impaired users
- Hardware Improvement: Compact the glove design for better usability

Development Plan Overview

Phase 4: Optimize Data Processing and Transfer

- Data Transfer Concerns: Address speed issues in communication
- Solution Exploration:
 - Option 1: Use AWS cloud services for efficient processing
 - Option 2: Perform computations directly on the iPhone
- Testing and Selection: Test solutions for performance evaluation



Requirements

Must have, should have, could have & want to have:

Prioritize functionalities based on their importance and urgency

User Requirements

Table 5.1: User Requirements Table

Requirement	Priority	Use Case(s)
Functional Requirement 1 (reqfDirections) <i>The system shall give directions to a destination.</i>	MustHave	<i>UC₁</i>
Functional Requirement 2 (reqfObstacleAvoid) <i>The system shall give directions to avoid obstacles.</i>	MustHave	<i>UC₂</i>

User Requirements

Requirement	Priority	Use Case(s)
Functional Requirement 3 (reqfOffline) <i>The system shall operate obstacle avoidance without the need for internet connection.</i>	MustHave	<i>UC₂,UC₃</i>
Interface Requirement 1 (requiReset) <i>The system shall have a reset button.</i>	ShouldHave	<i>UC₃</i>
Interface Requirement 2 (requiPower) <i>The system shall have a power button.</i>	MustHave	<i>UC₄</i>
Interface Requirement 3 (requiPairing) <i>The system shall have a Bluetooth pairing button or automatically become discoverable when disconnected from a phone.</i>	MustHave	<i>UC₃,UC₄</i>
Interface Requirement 4 (requiAccessibility) <i>The mobile application shall have accessibility features integrated from the iPhone.</i>	MustHave	<i>UC₁,UC₂,UC₄</i>

System Requirements

Table 5.2: System Requirements Table

Requirement	Priority	Use Case(s)
Constraint Requirement 1 (reqcAppleRequirement) <i>The system shall only be accessible through Apple iOS applications.</i>	MustHave	<i>UC₄</i>

Requirement	Priority	Use Case(s)
Constraint Requirement 2 (reqcBluetooth) <i>The system shall only communicate with the mobile application through Bluetooth.</i>	MustHave	<i>UC₄</i>

Non-Functional Requirements

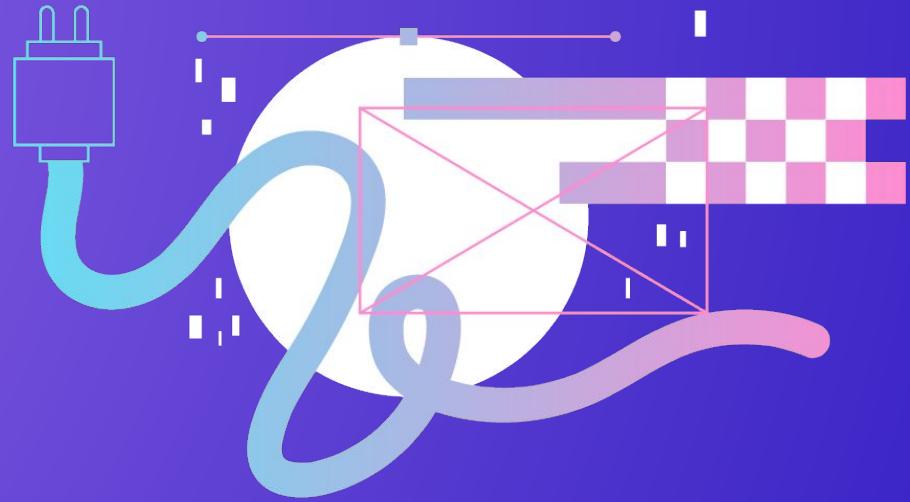
Table 5.3: Non-functional Requirements Table

Requirement	Priority	Use Case(s)
Quality Requirement 1 (reqqCommunicationSpeed) <i>The system shall perform the task with minimal communication delay.</i>	MustHave	<i>UC₁,UC₂</i>
Quality Requirement 2 (reqqBatteryLife) <i>The system shall perform the task with ample battery usage.</i>	ShouldHave	<i>UC₁,UC₂</i>

Domain Requirements

Table 5.4: Domain Requirements Table

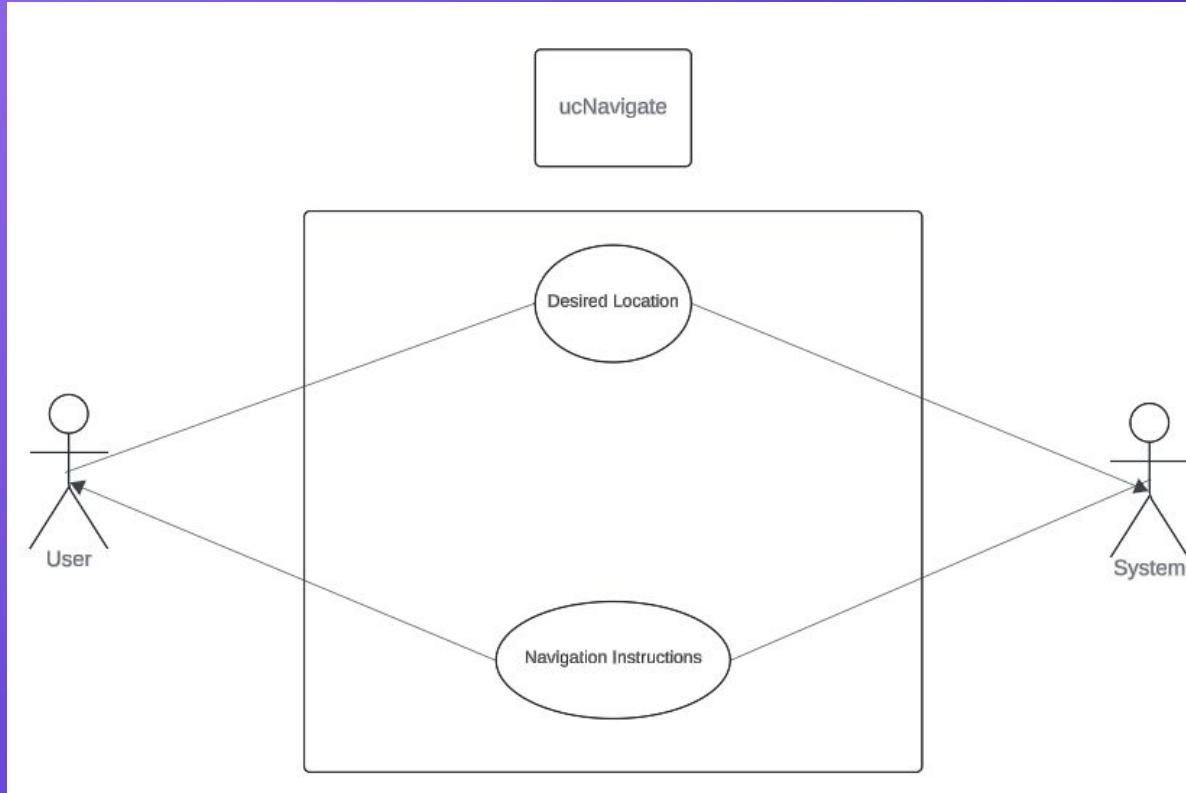
Requirement	Priority	Use Case(s)
Business Requirement 1 (reqbTutorial) <i>The system shall have a tutorial for stakeholder understanding.</i>	CouldHave	<i>UC₁,UC₂,UC₄</i>
Business Requirement 2 (reqbUpdates) <i>The mobile application shall offer software and firmware updates to the system for maintenance.</i>	ShouldHave	<i>UC₃,UC₄</i>



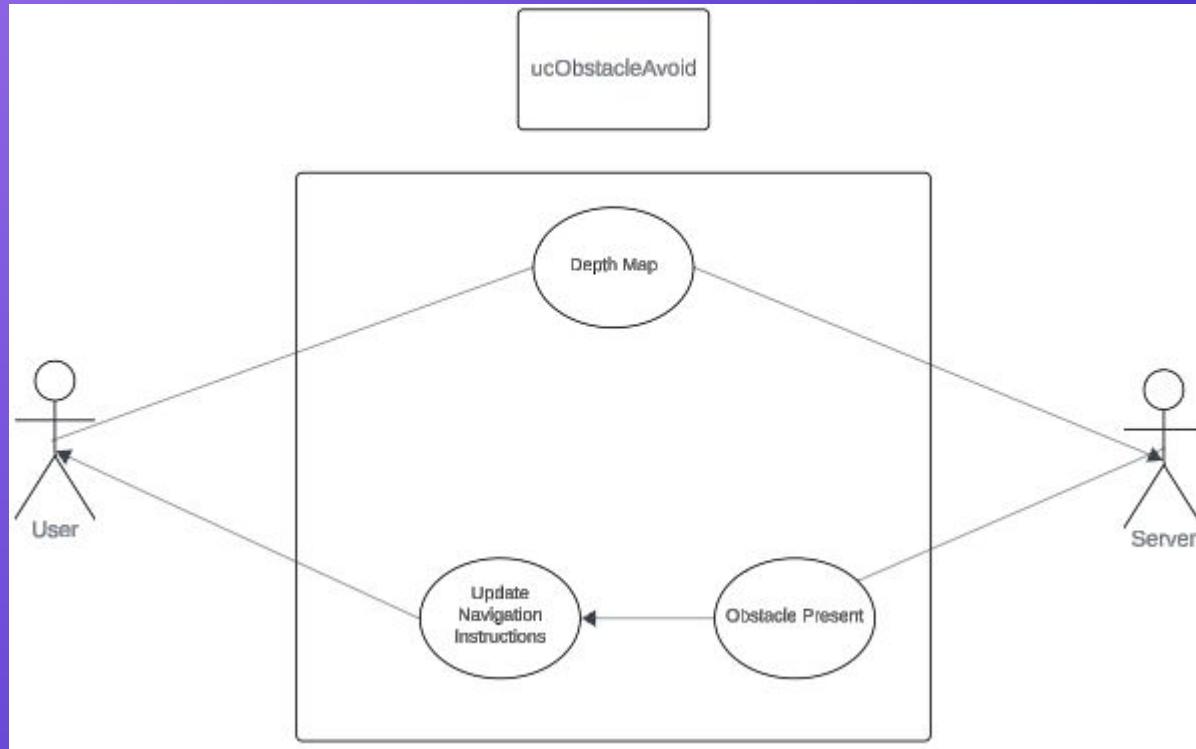
Use Case Modeling

Summarize details of system's users and their interactions with the system

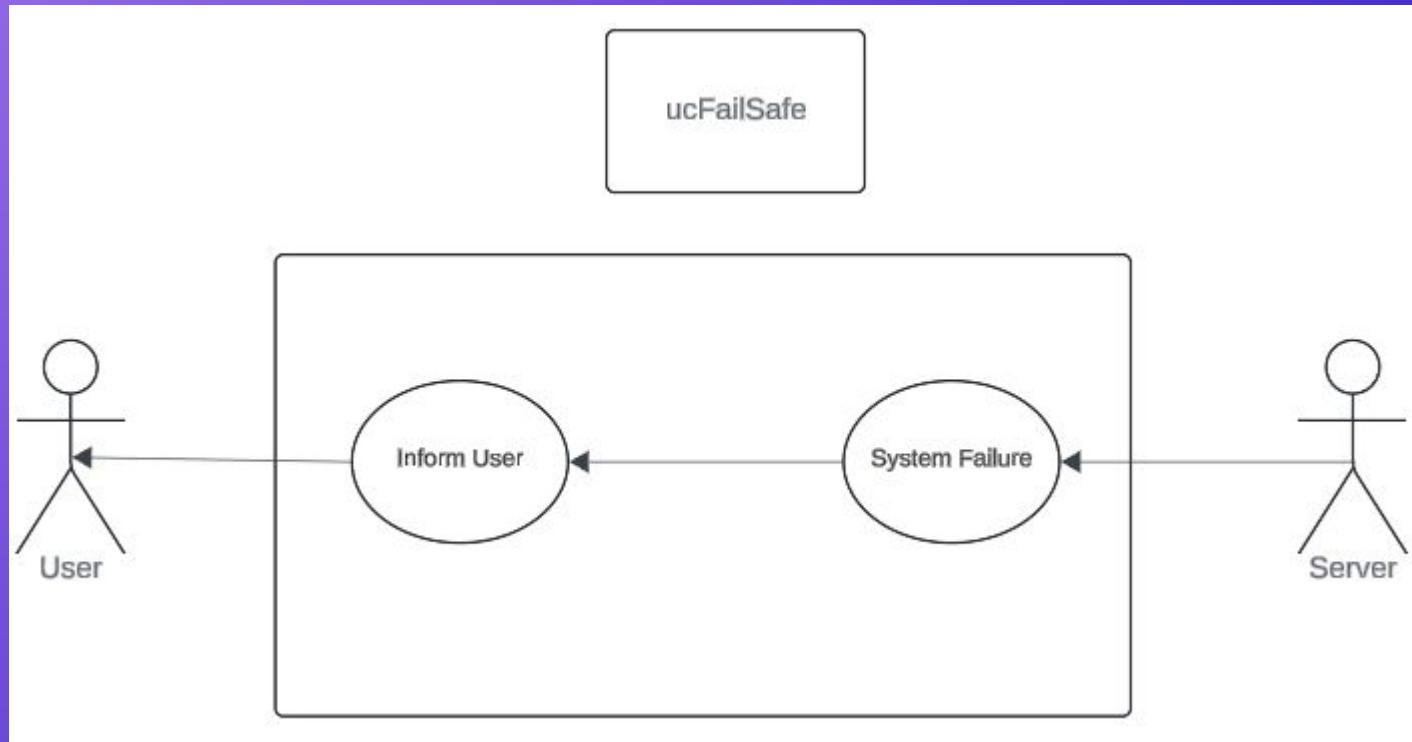
Use Case Diagram



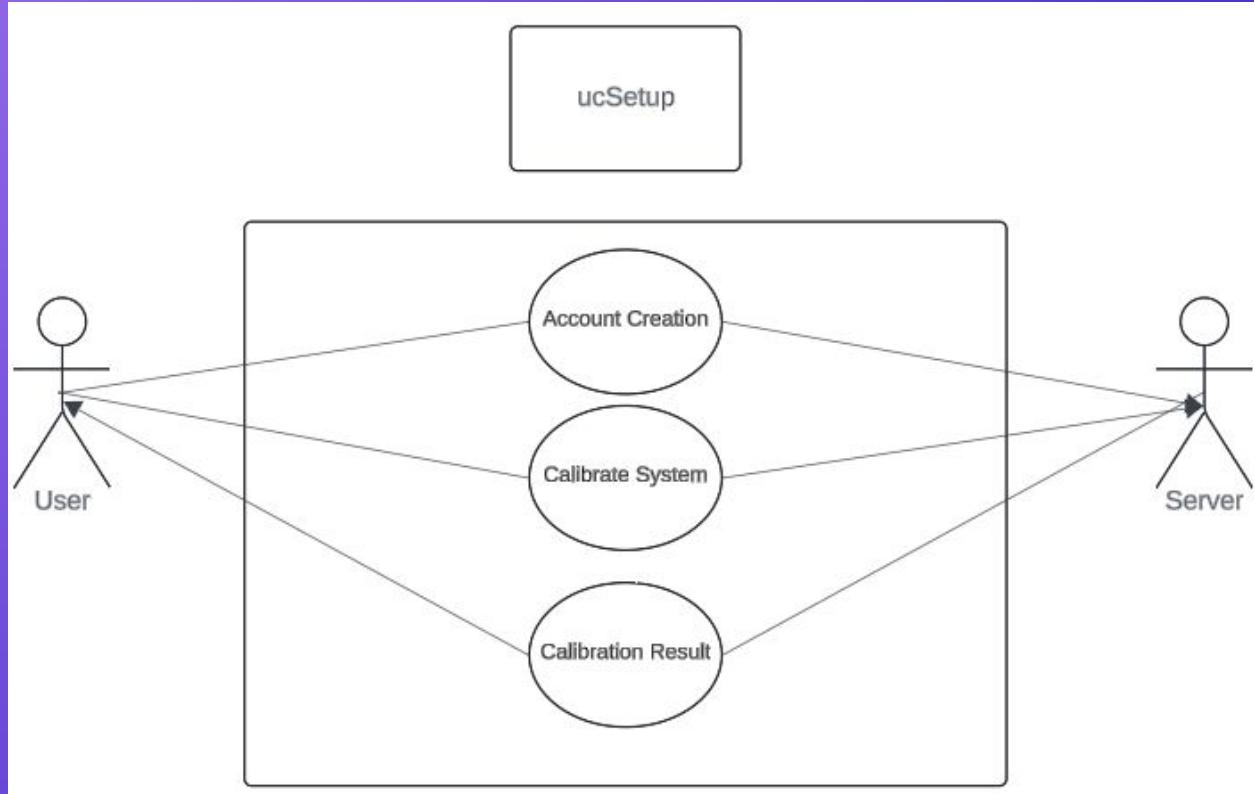
Use Case Diagram

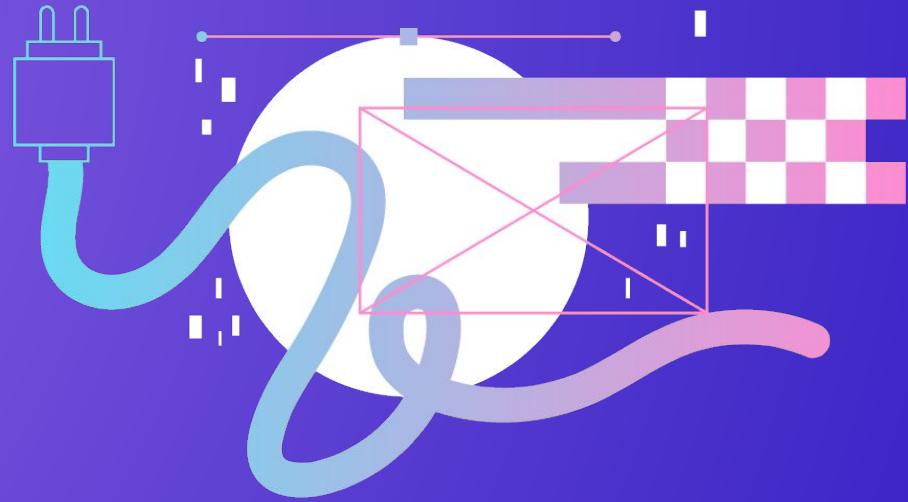


Use Case Diagram



Use Case Diagram





Use Case Specifications

Describe how a system interacts with external entities (users or other systems) to achieve specific goals

Use Case Specifications

<p>Use Case 1 (ucNavigate) <i>Navigate to Destination</i></p>
<p>Requirements: <i>reqkFunctional₁</i>, <i>reqkInterface₄</i>, <i>reqkQuality₁</i>, <i>reqkQuality₂</i>, <i>reqkBusiness₁</i>,</p>
<p>Diagrams: Figure 6.1</p>
<p>Brief description:</p>
<p>Give directions to the User to navigate to their desired destination</p>
<p>Primary actors:</p>
<p>User</p>
<p>Secondary actors:</p>
<p>Feedback device</p>
<p>Preconditions: The system has been set up properly.</p>
<p>Main flow:</p> <ol style="list-style-type: none">1. Input destination2. Send data to the server3. Send directions to the feedback device
<p>Postconditions: None</p>
<p>Alternative flows: None</p>

Use Case Specifications

<p>Use Case 2 (ucObstacleAvoid) <i>Avoid Obstacles</i></p>
<p>Requirements: <i>reqkFunctional₂</i>, <i>reqkFunctional₃</i>, <i>reqkInterface₄</i>, <i>reqkQuality₁</i>, <i>reqkQuality₂</i>, <i>reqkBusiness₁</i>,</p>
<p>Diagrams: Figure 6.2</p>
<p>Brief description: Give directions to the User to avoid obstacles</p>
<p>Primary actors: User</p>
<p>Secondary actors: Feedback device</p>
<p>Preconditions: The system has been set up properly.</p>
<p>Main flow:</p> <ol style="list-style-type: none">1. Input destination2. Send data to the server3. If obstacle then:<ol style="list-style-type: none">3.1. Calculate directions to avoid4. Send directions to the feedback device
<p>Postconditions: None</p>
<p>Alternative flows: None</p>

Use Case Specifications

Use Case 3 (ucFailSafe) Failure Notification

Requirements: *reqkFunctional₃*, *reqkInterface₁*, *reqkInterface₃*, *reqkBusiness₂*,

Diagrams: Figure 6.3

Brief description:

Notify the User of a system failure if directions can no longer be given reliably

Primary actors:

User

Secondary actors:

Feedback device

Preconditions: None

Main flow:

1. Detect system error
2. Notify the user
3. If offline system still available then:
 - 3.1. Continue to give directions to avoid obstacles

Postconditions: None

Alternative flows: None

Use Case Specifications

Use Case 4 (ucSetup) *System Setup*

Requirements: *reqkInterface₂*, *reqkInterface₃*, *reqkInterface₄*, *reqkConstraint₁*,
reqkConstraint₂, *reqkBusiness₁*, *reqkBusiness₂*,

Diagrams: Figure 6.4

Brief description:

Set up and calibrate the system for each user

Primary actors:

User

Secondary actors:

Feedback device, Distributors

Preconditions: None

Main flow: None

Postconditions: Have a fully operational system.

Alternative flows: None

Thank you!

Any Questions?