ECE 49595 - Initial Software Development Plan

Fall 2025

Project Overview

Our project aims to develop a smartphone-based indoor navigation system that empowers visually impaired individuals to navigate complex indoor environments safely and independently. The system leverages computer vision and machine learning to interpret live video from the user's smartphone, identifying room signs, obstacles, and environmental cues in real time. By integrating these visual detections with a digital floor plan, *Stride* dynamically localizes the user's position within a building and provides turn-by-turn guidance through audio and haptic feedback.

The primary objective is to create a proof-of-concept for the BHEE building at Purdue University, demonstrating the feasibility of real-time indoor navigation through accessible, non-visual interfaces. While the initial implementation focuses on a single building, the system's modular architecture will be designed to support future expansion across multiple buildings or larger indoor environments. Ultimately, our project seeks to close a critical accessibility gap in existing navigation technologies by providing reliable indoor guidance where GPS-based systems fail.

Scope

- **In Scope**
    - **Indoor navigation:** Focused on indoor navigation to address a critical gap in the current solution space. Indoor spaces often lack detailed maps leaving visually impaired people without reliable indoor navigation options.
    - **Audio and haptic feedback mechanisms** that deliver real-time, step-by-step guidance to the user through non-visual channels. This addresses our accessibility goals allowing visually impaired users to receive feedback.
    - **Camera-based object detection and localization using a pre-trained YOLO model** capable of identifying room number signs and obstacles in the user's path allowing for system localization and hazard detection.
    - **Integration with smartphone GPS and compass** to support continuous navigation updates and localization.
- **Under Evaluation**
    - **User interface design for both visually and non-visually impaired people** to extend usability beyond visually impaired individuals.
- **Out of Scope**
    - **Outdoor navigation**: Our project specifically targets the gap in indoor navigation.

- **Multi-building or large-scale campus integration**: Our scope is intentionally **limited to a single building (BHEE)** to allow focused development, testing, and validation in a controlled indoor environment. This restriction is primarily for practicality within the project timeline, not a limitation of system design. The software architecture, data models, and mapping pipeline will be **intentionally modular and scalable**, enabling easy extension to additional campus buildings in future phases with minimal modification

- **Assumptions**
  - **Users are smartphone owners** with basic familiarity using voice assistants or accessibility apps.
  - **Indoor lighting and signage conditions are consistent** enables reliable visual detection by pretrained model and consistency between room number signs and associated GPS coordinates for localization
  - **Users remaining inside the mapped region of the building** allows us to limit scope of the project.
  - **Reliable cellular connectivity,** allowing our application to stream live video frames for live inference and navigation updates
  - **Users grant camera and microphone permissions**, required for computer vision and voice feedback functionality.
- **Constraints**
  - **Budget constraints for cloud services:** Project will be self funded requiring careful optimization service usage for storage, inference, and compute.
  - **Time constraints:** Development must be completed during a one-semester timeline requiring careful prioritization of features.
  - **Bandwidth and latency**: Continuous video streaming from smartphone to cloud requires stable network connection. Limited cellular coverage or congestion could result in less responsive feedback.
  - **Training constraints:** We will train our model so it functions within BHEE at Purdue. This is due to our limited resources and the time required for collecting building-specific data

Requirements

Functional Requirements:

1. FR-1 Collision Detection Humans
    a. Statement: Under normal load, the system will detect people within 5 feet of the user.
    b. Rationale: The collision detection for other humans should be responsive enough to provide feedback to the user before the collision occurs.
    c. Test Method: Send 100 videos from the client device to perform end-to-end testing of the collision detection without the need for a user. The videos should have humans at varying distances from the camera with clear examples of humans within a 5 foot distance. Requirement met if a response is sent to the client device within 1 second for 97% of the instances in which humans are detected in the collision detection range (5 feet).
    d. Supporting Context: A quick estimation of total computation time for object detection plus distance estimation based on bounding box size gives us a worst case transaction time of 400ms. Thus a 1 second time limit was chosen to give a generous room for error while remaining responsive. Normal load is defined as 50 concurrent users.
    e. Trace: Collision Detection
    f. Priority: Must Have

2. FR-2 Collision Detection Objects
    a. Statement: Under normal load, the system will detect objects within 10 feet of the user.
    b. Rationale: The collision detection for objects should be responsive enough to provide feedback to the user before the collision occurs, however, since objects are stationary, the computation times don't need to be as fast as human collision detection.
    c. Test Method: Send 100 videos from the client device to perform end-to-end testing of the collision detection without the need for a user. The videos should have humans at varying distances from the camera with clear examples of objects within a 10 foot distance. Requirement met if a response is sent to the client device within 2 seconds for 97% of the instances in which objects are detected in the collision detection range (10 feet).
    d. Supporting Context: A quick estimation of total computation time for object detection plus distance estimation based on bounding box size gives us a worst case transaction time of 400ms. Thus a 2 second time limit was chosen to give a generous room for error while remaining responsive. Normal load is defined as 50 concurrent users.
    e. Trace: Collision Detection
    f. Priority: Must Have

3. FR-3 Indoor Navigation Instructions

a. Statement: The user must be able to navigate from current position to final destination inside a building with assistance from haptics and audio-directions.
b. Rationale: To make sure that the blind user is able to reach their destination, the app must be able to navigate the user successfully using camera and collision detection model for humans and objects.
c. Test Method: Using the navigation system with both the human and object collision detection models, the user must be able to reach their destination with haptic feedback and audio-directions at least 80% of the time
d. Supporting Context: This requirement assumes that the collision detection models for objects and humans work and the haptic feedback and audio control works as well.
e. Trace: User Navigation, Collision Detection
f. Priority: Must Have

4. FR-4 Indoor Landmark Detection
    a. Statement: Under normal load, the system will detect landmarks such as door signs/room numbers within a 5-10 feet distance to provide navigational instructions on time.
    b. Rationale: To properly localize the user digitally and relay the next instructions, the system needs to know where the user is in real life.
    c. Test Method: Provided a pre-recorded video feed of a user walking around a building, the system should detect and relay 95% of the rooms seen in the video as the playback is ongoing within the same 2-second latency as FR-2.
    d. Supporting Context: This requirement assumes the user has entered a destination to navigate to.
    e. Trace: User Navigation
    f. Priority: Must Have

5. FR-5 User Login
    a. Statement: Under normal load, the system should authenticate an existing user within 1 second of the user providing credentials through Face ID.
    b. Rationale: To provide a good user experience, the application should not be loading for long periods of time. A user count is required to optimize and scale the backend according to the number of concurrent/signed up users.
    c. Test Method: Perform 100 login attempts with valid credentials; measure response time. Requirement is met if 95% of attempts succeed within 1 second.
    d. Supporting Context: This requirement assumes the user already has an account created. Normal load is defined as 50 users using the product at the same time.
    e. Trace: User Login, User Authentication
    f. Priority: Could Have

6. FR-6 User Signup
    a. Statement: Under normal load, the system should create a use within 1 second of

the user providing credentials through face id
    b. Rationale: To have the user's data protected and secure, the application should be able to create a user that saves data. Privacy and security is key.
    c. Test Method: Being able to login with Face ID when opening the app for the first time and having the application create a user automatically within a second. Requirement is met if a user is created 100% of the time when opening the app for the very first time.
    d. Supporting Context: In order to login, there must be the creation of a user. Hence, this requirement will ensure that users will have an account that will store their data securely.
    e. Trace: User Authentication, User Signup
    f. Priority: Could have

7. FR-7 User Interface for Visually Impaired Users
    a. Statement: The user interface when the user is on the app is easy to use follows ADA requirements
    b. Rationale: Because the user is blind, the app needs to be accessible enough so that they can control the app using audio and speech.
    c. Test Method: The user is able to speak into the app to set the destination and the haptic feedback and audio-directions work 90% of the time
    d. Supporting Context: This requirement takes into consideration that user authentication works and the user is blind
    e. Trace: User Interface
    f. Priority: Must Have

8. FR-8 User Interface for Non-Visually Impaired Users
    a. Statement: The user interface when the user is on the app is easy to follow and visually acceptable
    b. Rationale: The user should be able to enter a destination and the navigation system should lead the user to the correct location
    c. Test Method: The user is able to enter a direction and the app should load directions to the destination within 10 seconds
    d. Supporting Context: This requirements takes into consideration that user authentication works and the user is not blind
    e. Trace: User Interface
    f. Priority: Could Have

Non-Functional Requirements:

1. NFR-1 Application Performance
    a. Statement: The mobile application should maintain optimal performance with minimal power consumption, ensuring smooth operation without significant impact on device battery life.
    b. Rationale: Considering users will be dependent on this mobile app for navigation for long periods (several hours to a day), the app shouldn't drain the device

battery.

c. Test Method: Measure the percentage change in battery level before starting the application and after running it continuously for 10-minute intervals under normal usage conditions. Average the results across multiple trials to ensure consistency.
d. Supporting Context: This requirement assumes no other high power requiring applications are running on the device to relate the power usage to the accessibility app performance.
e. Trace: UI/UX
f. Priority: Should Have

2. NFR-2 Mobile Application/ADA Accessibility
    a. Statement: The mobile application packaging the system must comply with accessibility standards such as WCAG 2.1, ensuring compatibility with assistive technologies.
    b. Rationale: To make the product responsive to user needs, the mobile application should be controllable with voice assistants.
    c. Test Method: Perform the mobile app functions using 3 most used voice assistants (Siri, Google, Alexa)
    d. Supporting Context: This requirement assumes the user has prior experience using a voice assistant and that the voice assistant is activated on the device the application is running on.
    e. Trace: UI/UX
    f. Priority: Must Have

3. NFR-3 Mobile Application Deployability
    g. Statement: The mobile application must be deployable across multiple devices and operating systems.
    h. Rationale: To make the product interactable with real users, the application must be deployable and used in real time.
    i. Test Method: Check if the app runs on multiple OS such as Android and iOS.
    j. Supporting Context: This requirement assumes the device is either Android or iOS, with all functional updates installed.
    k. Trace: UI/UX
    l. Priority: Must Have

Deliverables

- Stride Mobile Application Package (.apk)
    - Deliverable Description: The fully compiled and deployable software package for iOS and Android. This standalone product includes the accessibility-compliant user interface, haptic/audio feedback engines, and the integrated client-side logic required to communicate with the AWS backend. It is separate from the whole, end-to-end system (mobile application plus backend system).
    - Relevant Requirements: FR-4 to FR-8, NFR-2, NFR-3
- Optimized YOLO Model Artifacts:
    - Deliverable Description: The final set of trained model weights and configuration files for the YOLO-based object detection system. These artifacts will be fine-tuned for the BHEE building, ready for deployment on an inference endpoint, or release as an open-source artifact.
    - Relevant Requirements: FR-1, FR-2, FR-4
- System Architecture and Developer Documentation
    - Deliverable Description: A comprehensive technical guide designed for future contributors or maintenance teams. This will include the AWS serverless architecture, API specific, React Native environment setup, and deployment procedure documentation.
    - Relevant Requirements: NFR-3
- Verification and Validation Test Report
    - Deliverable Description: A report documenting the results of user and system testing. This will verify that we have met our requirements and the system is safe for users. It will include quantitative data on collision detection latency, navigation success rates, and battery performance metrics.
    - Relevant Requirements: NFR-1

Development Methodology

Chosen methodology: Scrum
We have chosen Scrum to manage the complex development of Stride. This methodology's core strength lies in its flexibility and adaptability. Both of which are essential given our single-semester timeline for implementation and the expectation that our requirements may evolve and shift as implementation progresses. By breaking work into short sprints of 3 weeks, we will be able to continuously deliver potentially shippable increments, enabling fast feedback loops with early validation from user testing. Although the project carries high technical and integration risk across the Computer Vision, cloud, and mobile components, Scrum provides a more agile framework than the risk-focused Spiral model. The time-boxed structure of Scrum will allow us to maintain momentum, absorb uncertainty, and prioritize the delivery of the core requirements.

Verification and Validation Plan:

- Include a link to your verification and validation plan. This should be a PDF in your GitHub doc folder.

https://github.com/AkashK321/Stride/blob/main/docs/Final%20Verification%20%26%20Validation%20Plan.pdf

Gantt Chart:

https://www.notion.so/Stride-Gantt-Chart-2a20d7343e2f80ecb020e8f644bf9d33