Functional specification
Smart Turn

By Josh Casey (21361783) & Jakub Czerniejewski (21466494)

Date Completed: 14/11/2024

1.Introduction

1.1 Overview

The Smart Turn system is an IoT enabled wearable device that allows road users such as cyclist and scooter users to indicate which direction they will be turning by swiping across a button to other road users such as cars and buses.On top our product comes with a display that can be mounted to the users bike or scooter, it is connected to the users phone via bluetooth and it displays chosen route directions This screen reduces distractions for the user in comparison to a mobile phone as it only shows the necessary information. Another benefit of using it is to keep the user's phone safe in a city environment where phone theft is on the rise.

1.2 Business Context

Smart turns aim is to improve urban road safety by offering cyclists and scooter users a visually clear way to indicate a change of direction using turn signals without requiring large hand gestures to be performed by them, also by providing a mounted display for turn by turn navigation users can keep their mobile devices out of sight and safe reducing the risk of theft of their mobile device.

1.3 Glossary

Arduino IDE

Development environment for the microcontroller

C

The language used for programming the microcontroller

Flutter

UI Toolkit for the development of the mobile application

Dart

The language user for the development in the flutter application.

UNO

An Arduino made microcontroller.

FLORA

A wearable arduino based microcontroller which can be sewed onto clothes.

2. General Description

2.1 Product / System Functions

User functionality consists of connecting our product to their phone via bluetooth and inputting a location or desired destination/route. The user then mounts their display and gesture button which are connected to an UNO to a desired location on their means of transport and equips the vest with led turn signals and a FLORA. Upon completion the user then has the ability to follow the map directions on their display and use smart gestures such as a swipe on button to indicate their turning direction to other road users.

2.2 User Characteristics and Objectives

The user category can be composed of two main users, cyclists and scooter users. Although scooter users have light indicative hardware already the product can be still beneficial as it is more luminous, visually appealing and is in the direct line of sight of other road users, not near the bottom of the scooter.

2.3 Operational Scenarios

User connects device
The users will open the app on their mobile device pushing a button on the app to connect their phone to the device via bluetooth.

User equips vest
Once the user has successfully connected their phone to the device the user will then equip the vest containing the LED’s and the ADA Flora.

Users mounts screen
The user will then be able to mount their display to their desired location on their mode of transport that best suits their needs

User mounts gesture button
The user will then be able to mount their gesture button to their desired location on their mode of transport that best suits their needs

User inputs desired location
The user will be prompted with the ability to select a location and route that they wish to travel to via the mobile app.

User selects left signal
When the user wishes to change the direction they are heading in they can use the smart gesture feature of the mounted electrodes on their ride by simply swiping left using their thumb or desired finger, causing the LEDs on the vest to indicate left.

User selects right signal
When the user wishes to change the direction they are heading in they can use the smart gesture feature of the mounted electrodes on their ride by simply swiping right using their thumb or desired finger, causing the LEDs on the vest to indicate right.

User turns off turn signal
When the user wishes to turn off the turn signal once they have made their turn they can tap the middle of the gesture button.

2.4 Constraints

Below are some constraints we may encounter during the production process of our product

Testing
In real world scenarios a user may plan on travelling far so for user testing a user will only be able to travel so far for us to obtain and view results

Mobile phone signal.
Poor signal while using the app will impact functionality.

GPS accuracy.
Accompanied with bad signal the gps accuracy will be less accurate with a poor phone signal.

Safety
How to make the product distract the user as little as possible.

Responsiveness
The efficiency of communication between the bluetooth components may impact functionality.

3. Functional Requirements

3.1 Connecting to device

Description

The user will be able to connect to both devices (Arduino uno, ADA Flora) via a Flutter app on their mobile phone. This should be done by a click of a button connecting all 3 devices and syncing them.

Criticality

This is essential to the system as if nothing is connected or synced up there will be 0 functionality provided

Technical Difficulties

Ensuring connection is established.

Dependencies with other requirements

This is our initial functionality so it does not require and prior functionality or requirements

3.2 User equips vest

Description

The user equips their vest containing the ADA Flora which is connected to and controls the LED functionality.

Criticality

The user being able to equip the vest is essential as it provides the main functionality with the LEDs being attached to it displaying the projects main functionality,

Technical Difficulties

Potential thermal and material considerations for the mounting of the leds to cloths.

Dependencies with other requirements

This is dependent on the functionality of 3.1 as if the vest cannot connect to the mobile app it will be rendered unusable.

3.3 User mounts screen

Description

The user will be able to mount touch sensor to their mode of transport in their position of choice to obtain the functionality of the smart turn gesture.

Criticality
This is critical to the system as it eliminates the need for large gestures for signalling providing the main functionality and desire for the product.

Technical Difficulties
Touch sensitivity and smart gesture control will need a custom algorithm to function as we intend and to avoid mis inputs via outside control (wind) and mis input via the user.

Dependencies with other requirements
This is also dependent on the whole system being connected and synced up.

3.4 User inputs location

Description
The user will be able to input their desired location and travel route via the flutter made mobile app which will then be displayed on their mounted display providing them with direction.

Criticality
Very high, without navigation the systems features are greatly reduced

Technical Difficulties
Maintaining real time location services and potential route recalculations

Dependencies with other requirements
3.3 & 3.1

3.5 Updating Route Directions

Description
The user will be able to follow directions from the screen which will update as they proceed towards their destination.

Criticality
High as the product needs to provide accurate navigation information.

Technical Difficulties
Managing and updating the information in a timely manner.

Dependencies with other requirements
Users input depending on 3.4

3.6 User signals left

Description
The users swipes left on the electrodes activating the LEDs to signal left

Criticality
Essential for system functionality and user safety

Technical Difficulties
Sensitivity and calibration challenges for gesture recognition and false signal detection including weather and accidental grips

Dependencies with other requirements
Equipped vest and a fully functional and connected system, 3.1 & 3.2

3.7 User signals right

Description
The users swipes right on the electrodes activating the LEDs to signal right

Criticality
Essential for system functionality and user safety

Technical Difficulties
Sensitivity and calibration challenges for gesture recognition and false signal detection including weather and accidental grips.

Dependencies with other requirements
Equipped vest and a fully functional and connected system, 3.1 & 3.2

3.8 User turns off turn signal

Description
The taps the middle on the electrodes deactivating any active LED turn signal.

Criticality
Essential for system functionality and user safety

Technical Difficulties
Output processing from touch sensor to avoid false signals.

Dependencies with other requirements
Equipped vest and a fully functional and connected system, 3.6 & 3.7

3.9 Displaying Turn Signal Status

Description
A left/right arrow is displayed on the screen indicating that a turn signal is on.

Criticality
Moderate to High as the user should be notified wether a turn signal is on to not confuse other road users.

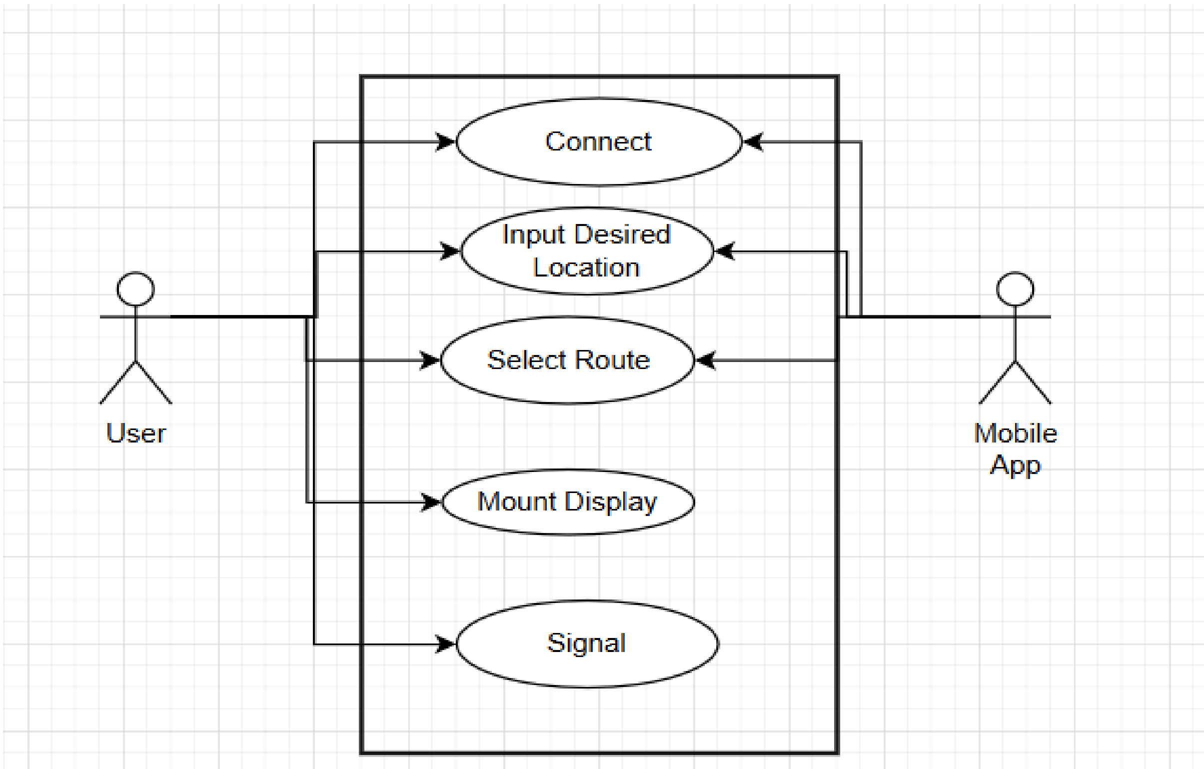
Technical Difficulties
Checking wether turn signal is on.

Dependencies with other requirements
Equipped vest and a fully functional and connected system, 3.6 & 3.7

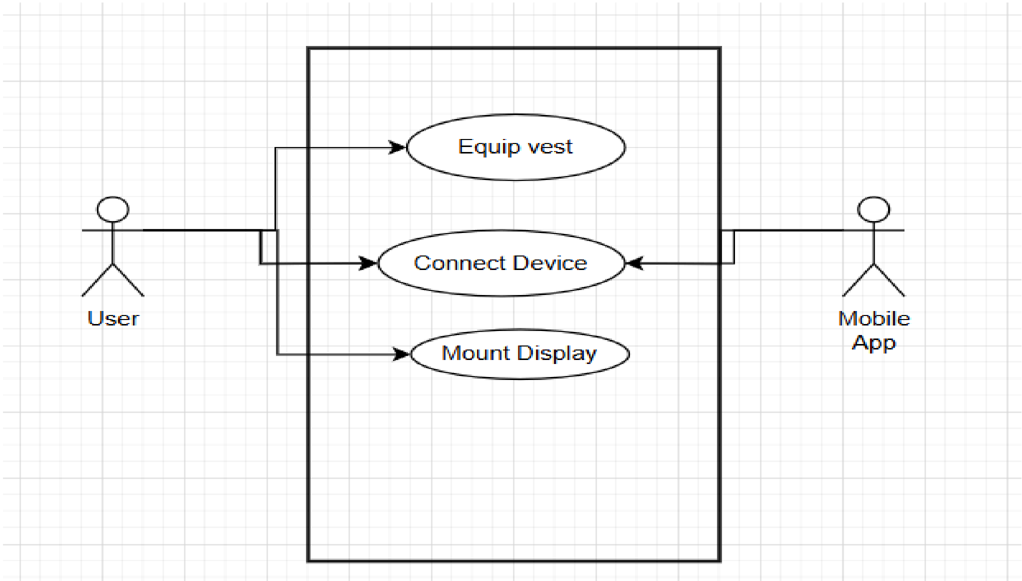
4. System architecture

Use case diagrams

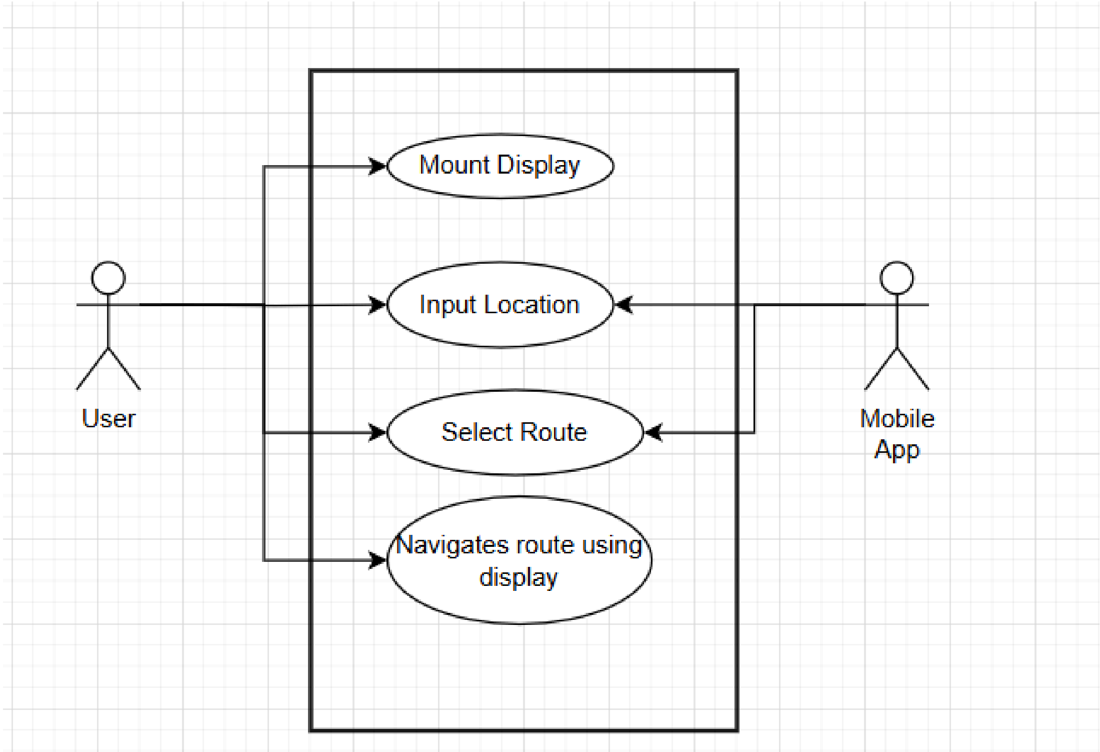
System overview use case



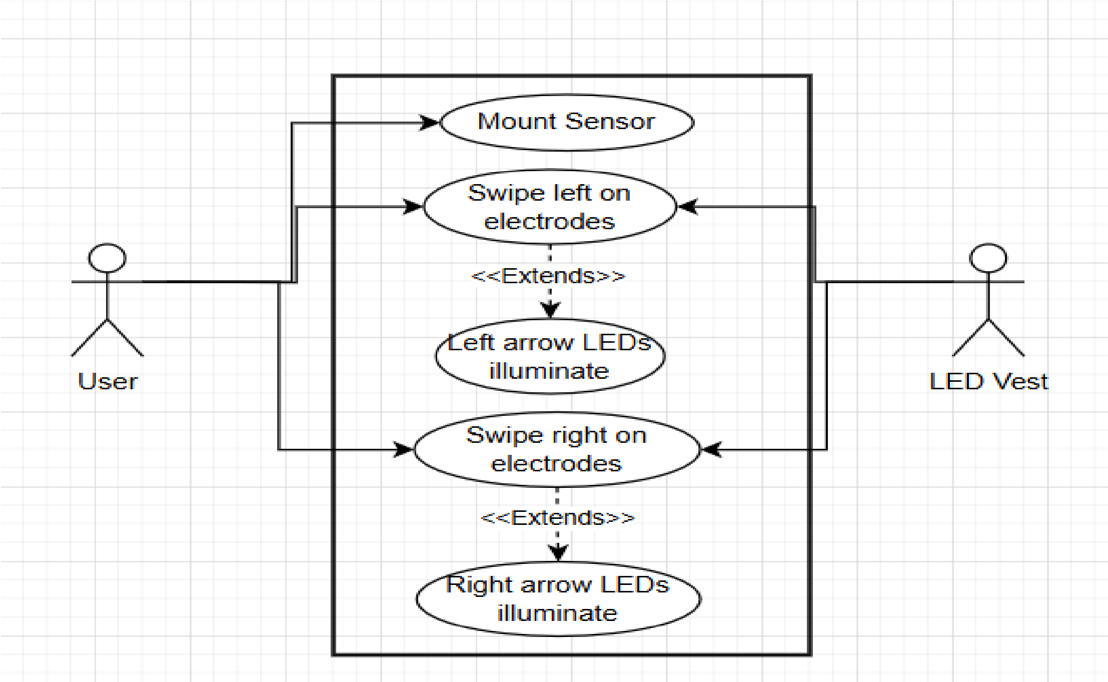
Bluetooth connection use case



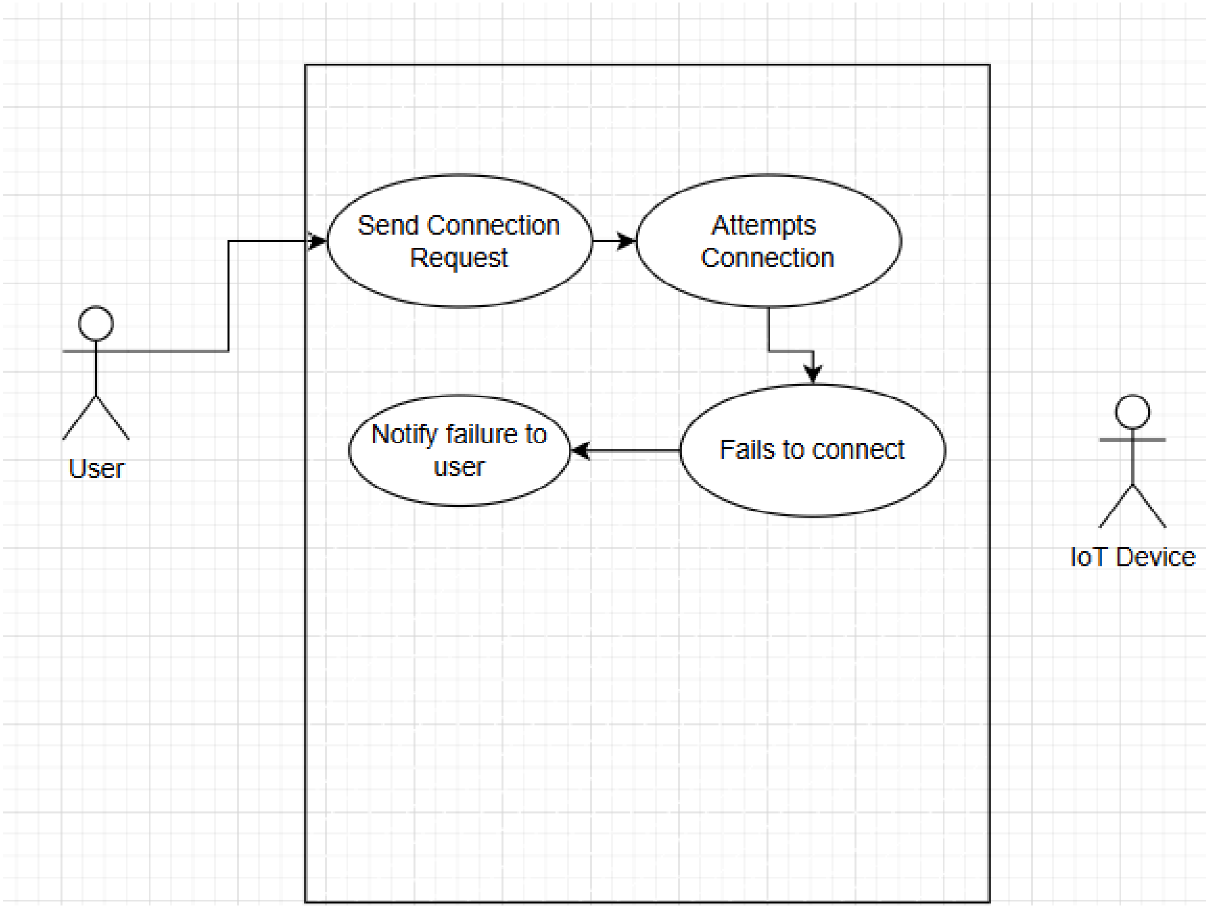
Inputs desired location



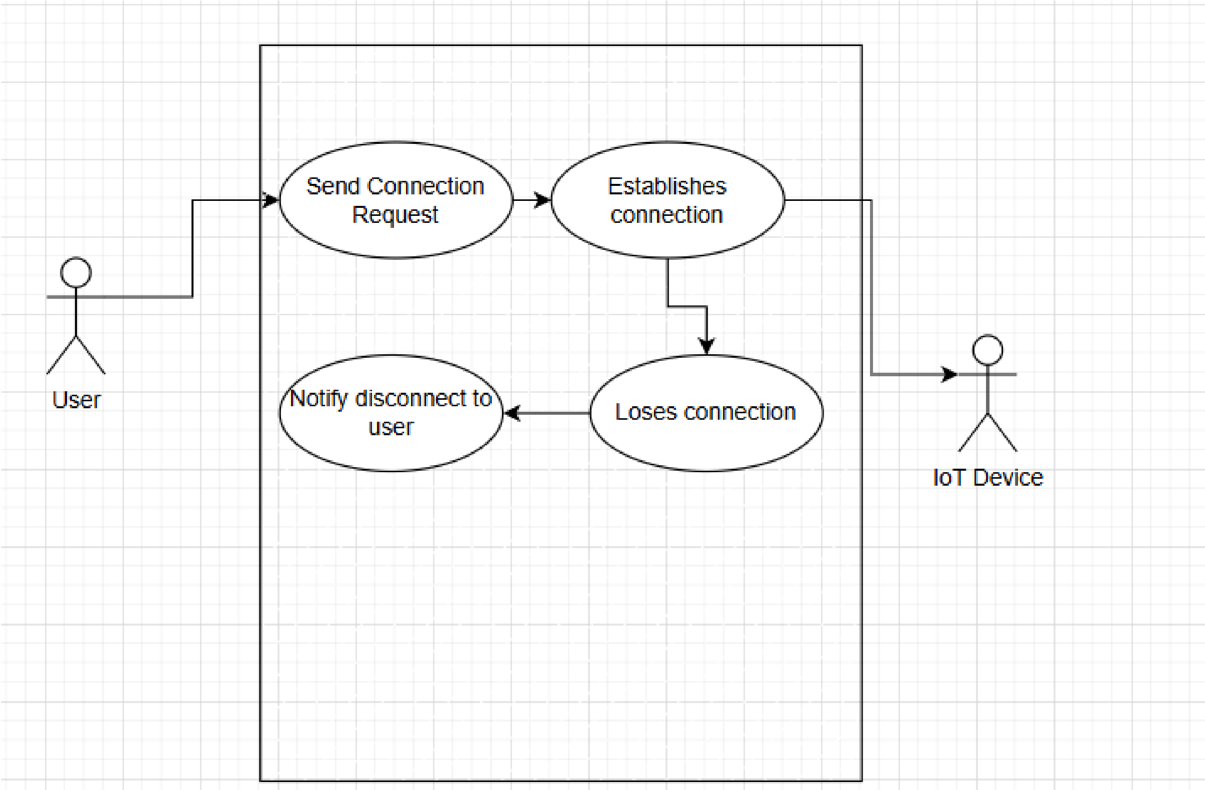
Indication use case



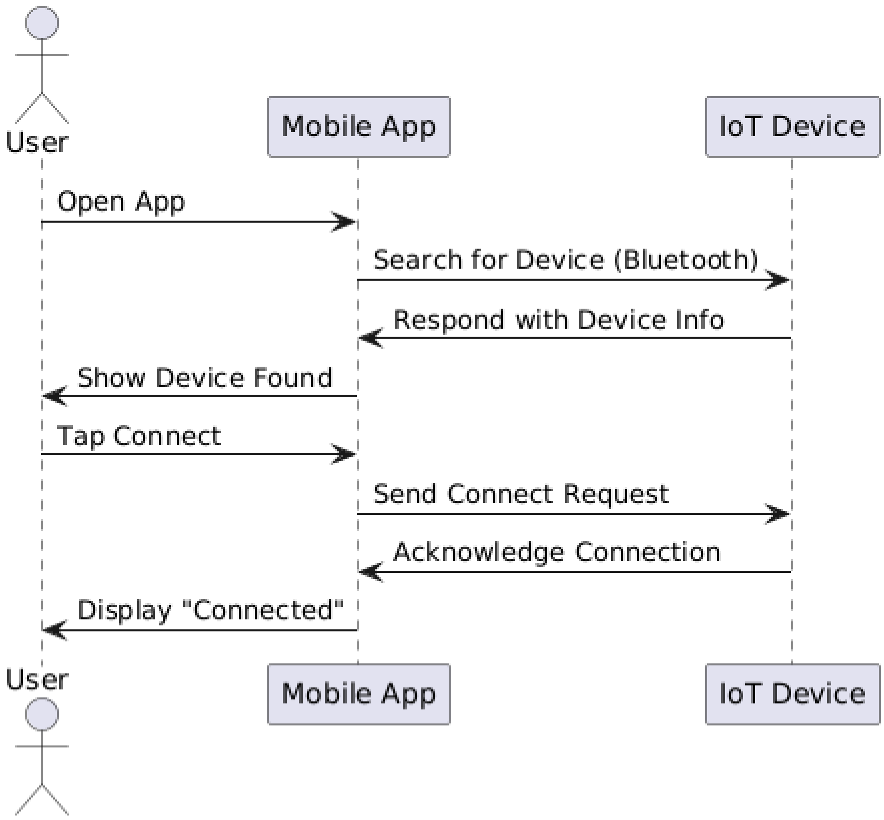
Fails to connect



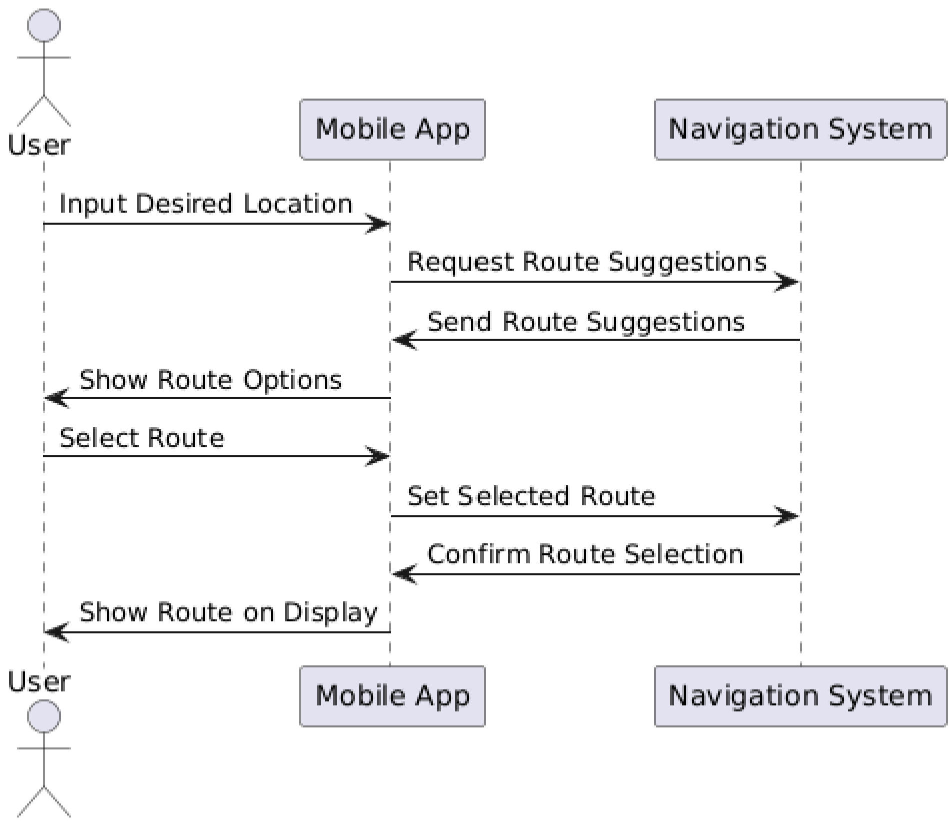
User loses connection



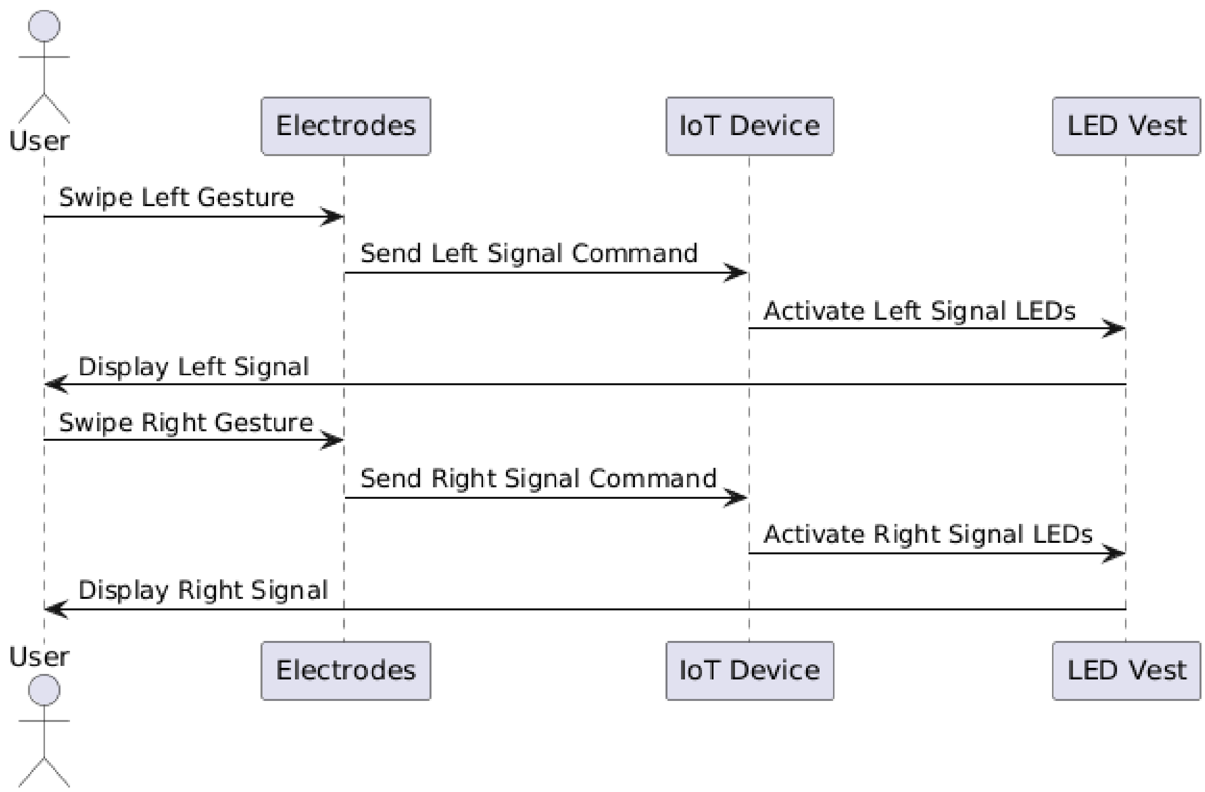
Sequence diagrams



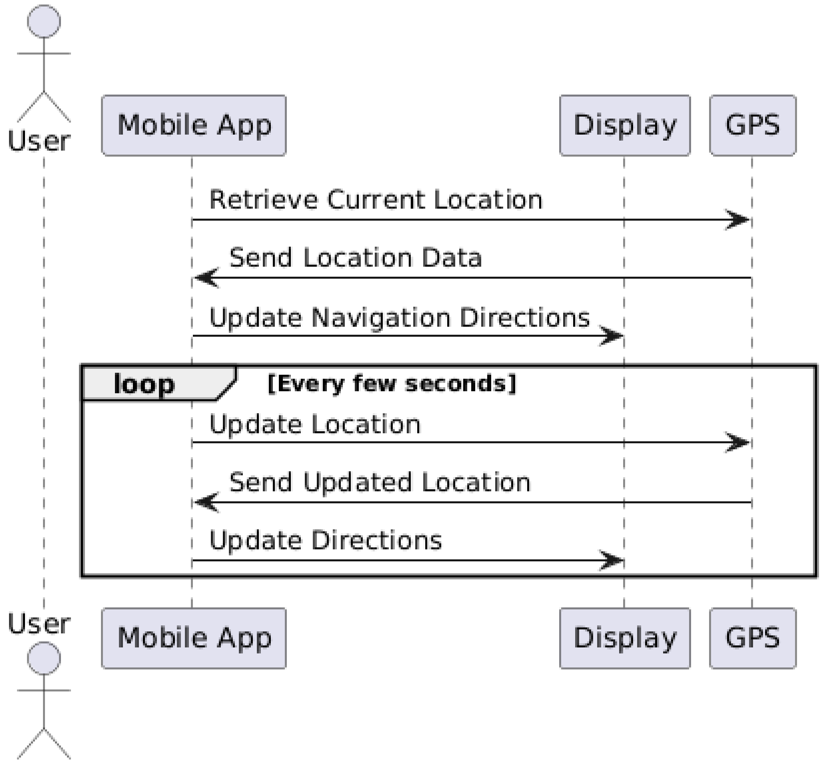
Location & Direction



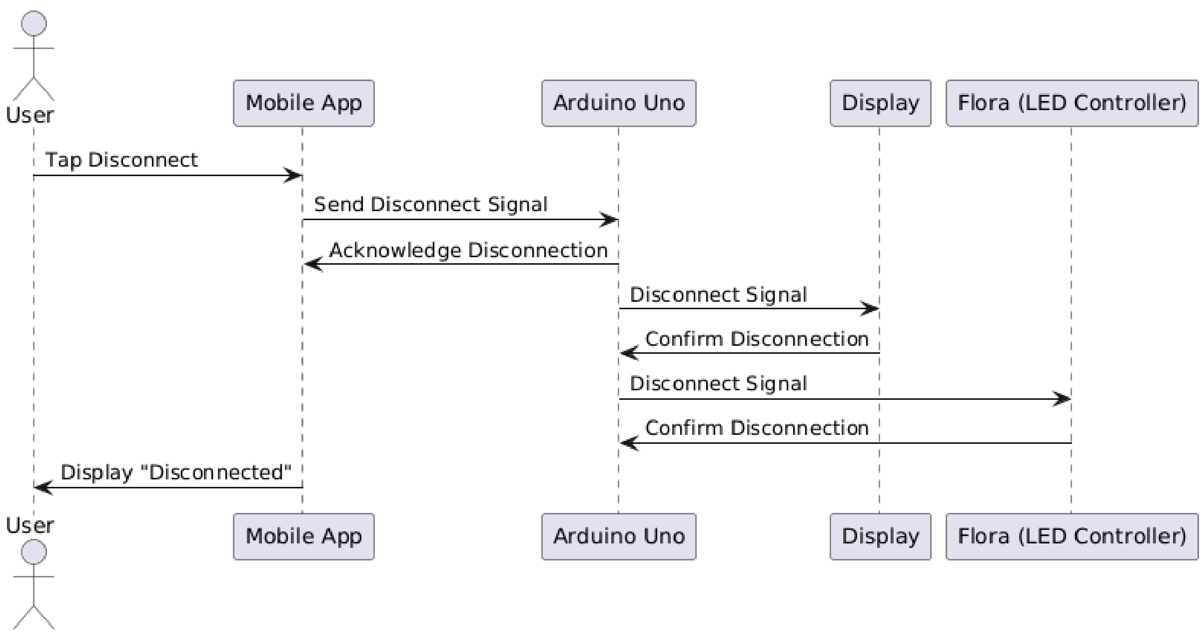
Indication



Navigation

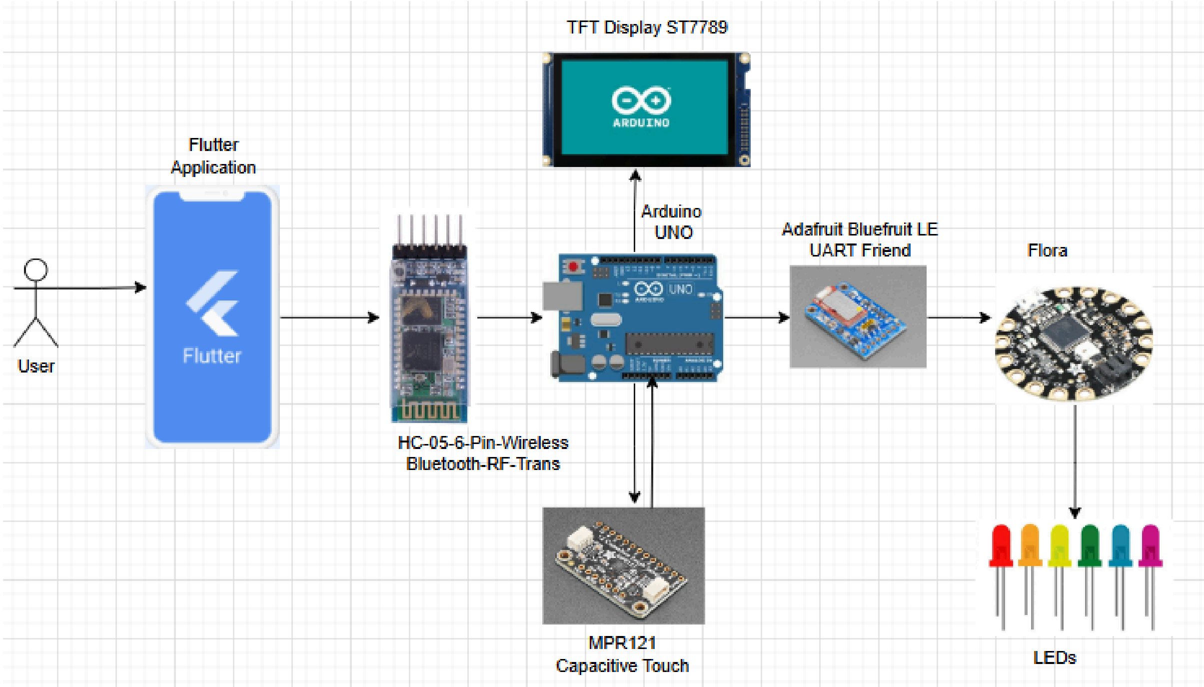


Disconnecting

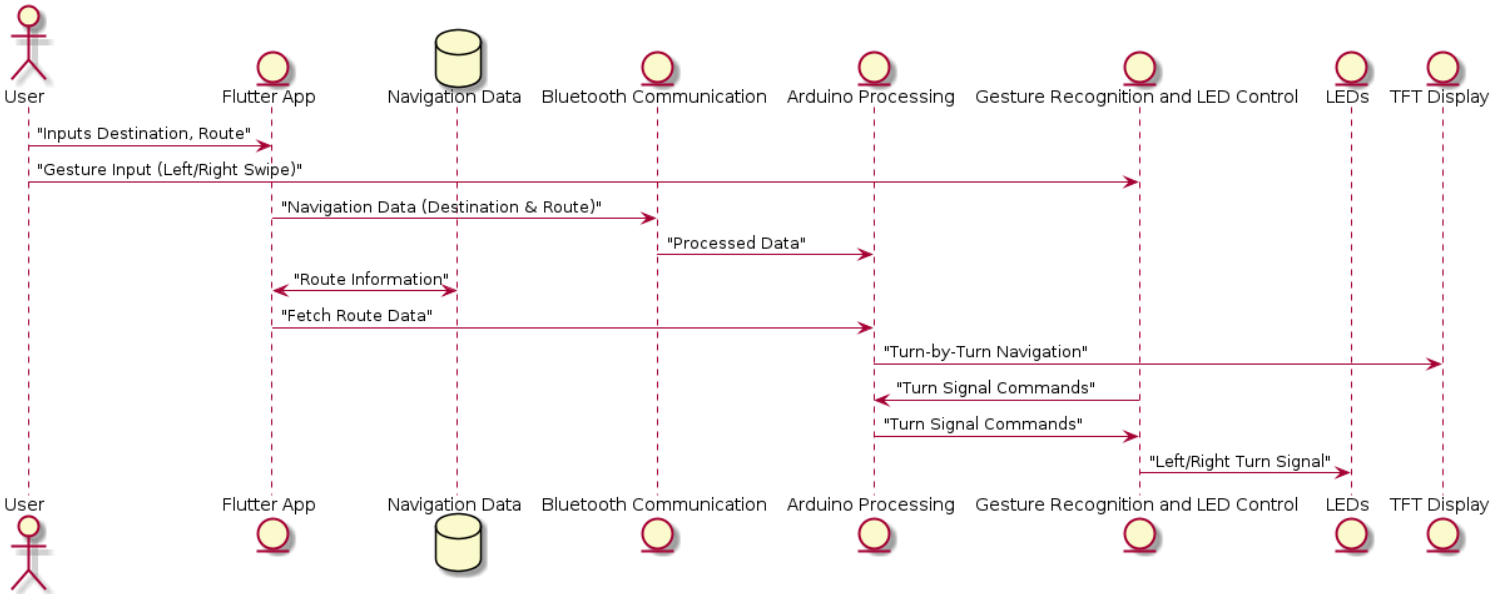


5. High Level Design

System Architecture



Data Flow Diagram



6. Test Plan

Our project will have the following categories of testing performed on it:

Unit Testing

1. Testing Flutter map api by querying locations and getting directions.
2. Allowing the app to use the phone's bluetooth.
3. Processing API responses to retrieve relevant information.
4. Accessing HC-05 bluetooth module from UNO.
5. Accessing data from MPR121 capacitive module.
6. Testing display operation.
7. Accessing the Bluefruit bluetooth module from FLORA.
8. Lighting up LEDs which are connected to FLORA.

Integration Testing

1. Connecting UNO and Flutter app via Bluetooth
2. Sending and Retrieving data on UNO from Flutter app
3. Inputting desired destination and passing it to UNO via Flutter app.
4. Displaying navigation data from Flutter app on IPS screen.
5. Receiving and processing output from touch sensor(MPR121).
6. Displaying left/right arrow depending on touch sensor output.
7. Connecting UNO and FLORA via Bluetooth.

8. Sending Data from UNO to FLORA.
9. Testing that right/left LED’s light up in desired fashion.
10. Saving and sending MPR121 output to FLORA to turn on the corresponding turn signal.

System Testing

1. Opening our Flutter app on a mobile, establishing a connection with our UNO device inputting a desired destination.
2. Sending desired location to UNO, activating and turning off both left and right turn signals.
3. Displaying route directions and turn signal status.
4. Updating route directions as progress is being made towards the goal destination, using turn signals to indicate change of direction..
5. All of the above in sequence (1-4).

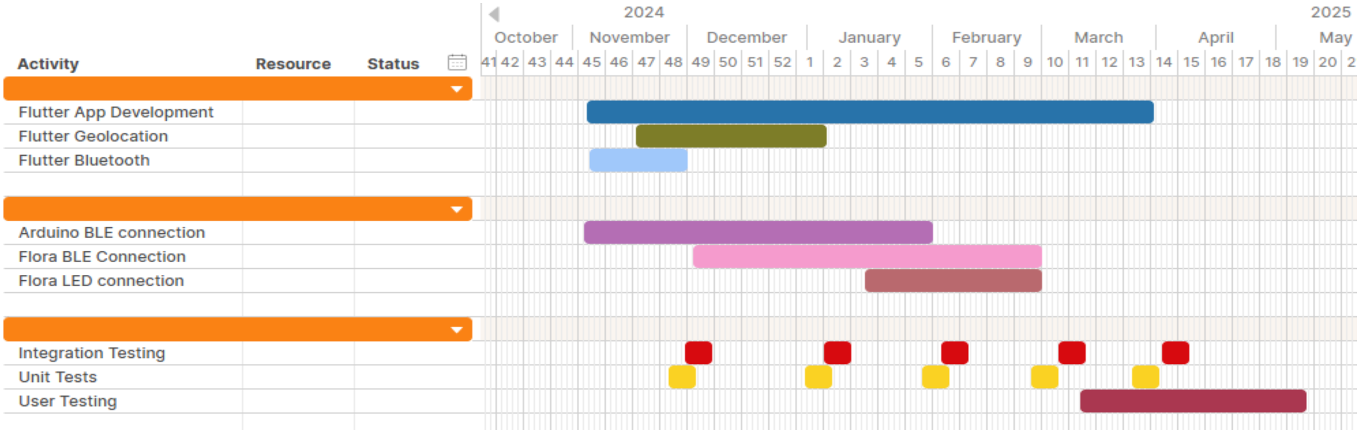
User Testing

1. Ask the user to navigate our Flutter app to evaluate ease of use and UI layout.
2. Mount the UNO, screen and touch sensor on a bicycle/scooter and ask the user to put on a jacket with a FLORA and LEDs to evaluate whether activating the turn signals feels natural while cycling/using the scooter.
3. Ask the user to input a destination using the flutter app followed by them cycling/using a scooter using the directions provided on our screen and indicating change of direction using our touch sensor and turn signals mounted on the jacket to evaluate: do our users think it improves safety and is it distracting to use.
4. Ask the user to fill out a survey asking questions regarding their experience of using our product.

All of the testing will be recorded in a directory structure with a separate section for each kind of testing. We also plan on using Jenkins pipelines for CI/CD when new features are being added to the project.
The test cases consist of: verifying application features, boundary cases and percentage coverage of codebase.

7. Preliminary Schedule

Gantt chart overview



8. Appendix

Data Flow diagram - [PlantUML]
Use Case Diagrams - [Draw.io]
System Architecture Diagram - [PlantUML]
Gantt Chart - [TeamGantt]