Route Navigator
Functional Specification

Name: Chi Huu Huynh

Student Number: C00261172

Supervisor: Dr Oisin Cawley

Table of Contents

Introduction

This report is to outline the main functionality and use cases of the Route Navigator. The report contains the following:

- Application Overview - Describes the main goal of the application.
- Context Diagram - Shows the interactions between the user and the system of the application
- Use Case Diagram - Diagram which shows all the situations on how the application will get used
- Use Cases - The situations of how the application is going to get used
- Model - Contains FURPS+ (Functionality, Usability, Reliability, Performance, Supportability, Plus), which is used to describe the functional and non-functional requirements of the application.

Application Overview

The goal of this application is to allow users to rent a Sunfounder PiCar-V to navigate its surroundings and explore the region.

The room owners who have a Sunfounder PiCar-V can rent theirs to earn credits from other users using it.

Target Audience

The target audience of this application are:

- **Tourists and Travellers** - Tourists visiting a new city can rent small remote cars for a novel and engaging way to experience local attractions up close and personal.

- **Parents and Children** - The application lets families engage kids in remote-controlled car adventures to explore parks, museums, and public space.

While the target audience is very limited, the application can gain traction through social media based on location of the cars; furthering the wider audience.

Context Diagram

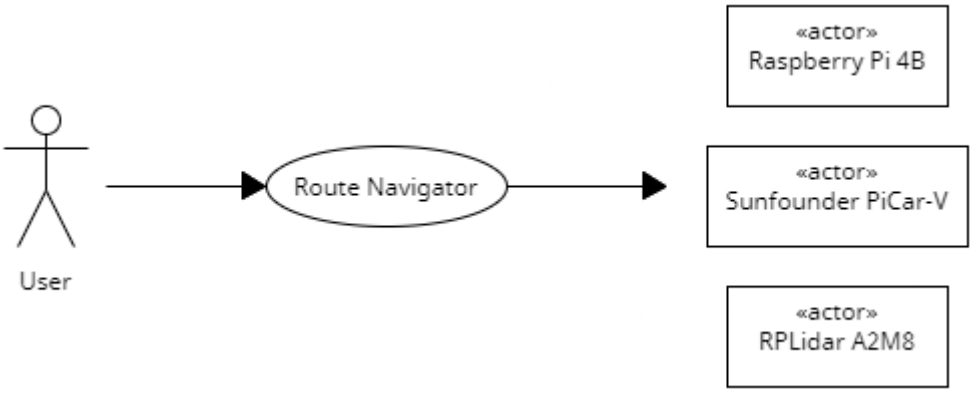

Figure 1, Context Diagram

Use Case Diagram

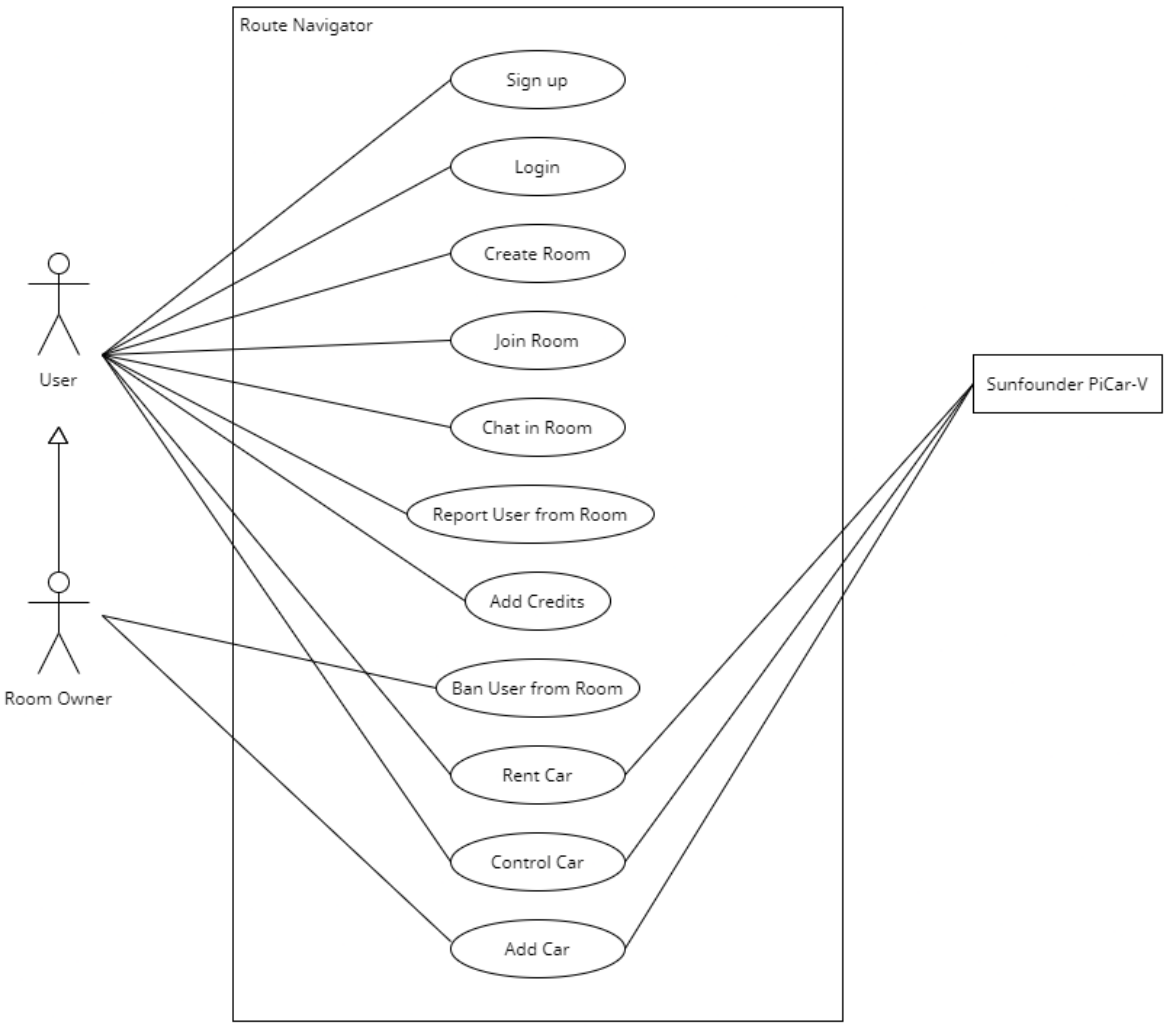

Figure 2, Use Case Diagram

Use Cases

Brief Use Cases

Use Case Name: Sign Up
Actors: User
Description: This use case begins when the user enters the website and requests to sign up. They will input their username and password to create an account.

Use Case Name: Login
Actors: User
Description: This use case begins when a returning user enters the website and requests to login. They will input their username and password to login into their account. If the password is correct, the user can login into their account.

Use Case Name: Create Room
Actors: User
Description: This use case begins when a logged in user requests to create a room. They will input the room name and password to create a room. The room is created.

Use Case Name: Join Room
Actors: User
Description: This use case begins when a logged in user requests to join an already created room. They will input the room name and password to join a room. If the password is correct, the user will be able to join the room.

Use Case Name: Chat in Room
Actors: User
Description: This use case begins when a logged in user inside of an already created room wants to send a chat message. They will input their message into the chat box and send it. The message is sent to other users in the room.

Use Case Name: Report User from Room
Actors: User
Description: This use case begins when a logged in user inside of an already created room wants to report another user. They will write a reason for their report and the report would get added to a report list with the reported user, reason and their messages.

Use Case Name: Ban User from Room
Actors: Room Owner
Description: This use case begins when a room owner wants to ban a user from their room. They will go through the report list in their room to decide whether to ban a user. The user would then get banned from the room owner’s room.

Use Case Name: Add Credits
Actors: User
Description: This use case begins when a logged in user wants to add credits to their account. They will add credits to their account by connecting to their bank and selecting an option to add credits. Afterwards the chosen amount of credits is added to their account.

Use Case Name: Rent Car
Actors: User, Sunfounder PiCar-V
Description: This use case begins when a logged in user inside of an already created room wants to rent a car. The user will use their credits to rent the car. The system accepts the credits and allows the user to rent the car.

Use Case Name: Control Car
Actors: User, Sunfounder PiCar-V
Description: This use case begins when a logged in user inside of an already created room has rented a car. They will draw a path for the car to follow. The car will then follow the path.

Use Case Name: Add Car
Actors: Room Owner
Description: This use case begins when a room owner requests to add a car. The system will check if it can connect to the car and if it can, it will add the car to the system.

Detailed Use Cases

Use Case Name: Sign up

Actors: User

Brief Description: This use case begins when the user enters the website and requests to sign up.

Pre-Conditions:

Main Success Scenario:

1. The user wants to sign up.
2. The user enters their username, password and email.
3. The system checks the username, password and email.
4. The user is now at the home page with their created account.

Alternatives:

3a. The username already exists.

1. An error message is displayed telling the user that the username already exists.
2. The user is redirected back to step 2.

3b. The email is already used.

1. An error message is displayed telling the user that the email is already used.
2. The user is redirected back to step 2.

Post-Conditions: None

Use Case Name: Login

Actors: User

Brief Description: This use case begins when a returning user enters the website and requests to login.

Pre-Conditions:

- The user's account is already created.

Main Success Scenario:

1. The user wants to login.
2. The user enters their email and password.
3. The system verifies the email and password.
4. The user is now at the home page with their account.

Alternatives:

3a. The email / password is incorrect.

1. An error message is displayed telling the user that the email or the password is incorrect.
2. The user is redirected back to step 2.

Post-Conditions: None

Use Case Name: Create Room

Actors: User

Brief Description: This use case begins when a user requests to create a room.

Pre-Conditions:

- The user is already logged in.

Main Success Scenario:

1. The user wants to create a room.
2. The user enters a room name and password.
3. The system checks the room name and password.
4. The user has created a room with the given room name and password.

Alternatives:

3a. The room name already exists.

1. An error message is displayed telling the user that the room name already exists.
2. The user is redirected back to step 2.

Post-Conditions: None

Use Case Name: Join a Room

Actors: User

Brief Description: This use case begins when a user requests to join a room.

Pre-Conditions:
- The user is already logged in.

Main Success Scenario:
1. The user wants to join a room.
2. The user enters a room name and password.
3. The system verifies the room name and password.
4. The user has joined a room with the given room name and password.

Alternatives:

3a. The room password is incorrect.
   1. An error message is displayed telling the user that the room password is incorrect.
   2. The user is redirected back to step 2.

Post-Conditions: None

Use Case Name: Chat in Room

Actors: User

Brief Description: This use case begins when a user wants to send a chat message.

Pre-Conditions:
- The user is already logged in.
- The user is inside of a room.

Main Success Scenario:
1. The user wants to send a chat message.
2. The user enters in their message.
3. The system checks the message.
4. The user has sent the message.

Alternatives:

3a. The message is inappropriate.
   1. An error message is displayed telling the user that their message is inappropriate.
   2. The user is redirected back to step 2.

Post-Conditions: None

Use Case Name: Report User from Room
Actors: User
Brief Description: This use case begins when a user wants to report another user.
Pre-Conditions: ● The user is already logged in ● The user is inside of a room. ● There are two users in this room.
Main Success Scenario: 1. The user ‘A’ wants to report user ‘B’. 2. The user ‘A’ selects the user ‘B’ and reports them. 3. The system adds the user ‘B’ to a report list. 4. The user ‘A’ has successfully reported user ‘B’. **Alternatives:** None
Post-Conditions: None

Use Case Name: Ban User from Room
Actors: Room Owner
Brief Description: This use case begins when a room owner wants to ban a user.
Pre-Conditions: ● The room owner is already logged in ● The room owner is inside of a room. ● There are two users in this room including the room owner. ● There is a report on the user.
Main Success Scenario: 1. The room owner wants to ban a user from the room. 2. The room owner goes through the report list. 3. The room owner finds and selects the reported user. 4. The room owner decides whether to ban the user. 5. The user is banned from the room. **Alternatives:** None
Post-Conditions: None

Use Case Name: Add Credits

Actors: User

Brief Description: This use case begins when a user wants to add credits to their account.

Pre-Conditions:

- The user is already logged in.

Main Success Scenario:

1. The user wants to add credits to their account.
2. The user selects an option to add credits to their account.
3. The user inputs their bank details to purchase the credits.
4. The system checks if the bank details are correct.
5. The user has added credits to their account.

Alternatives:

3a. The bank details are incorrect.

1. An error message is displayed telling the user that the bank details are incorrect.
2. The user is redirected back to step 3.

Post-Conditions: None

Use Case Name: Rent Car

Actors: User

Brief Description: This use case begins when a user wants to rent a car.

Pre-Conditions:

- The user is already logged in.
- The user is inside of a room.
- A car is available in the room.

Main Success Scenario:

1. The user wants to rent a car.
2. The user inputs the amount of credits to rent a car.
3. The system checks the requested amount of credits and the balance of the user.
4. The user is now renting the car.

Alternatives:

3a. The user does not have enough credits in balance to rent a car.

1. An error message is displayed telling the user that they do not have the required credits.
2. The user is redirected back to step 2.

Post-Conditions: None

Use Case Name: Control Car
Actors: User
Brief Description: This use case begins when a user wants to rent a car.
Pre-Conditions: • The user is already logged in. • The user is inside of a room. • A car is available in the room. • The car is already rented by the user.
Main Success Scenario: 1. The user wants to control the car. 2. The user inputs commands to control the car. 3. The system receives the commands. 4. The user is now controlling the car. **Alternatives:** None
Post-Conditions: None

Use Case Name: Add Car

Actors: User

Brief Description: This use case begins when the room owner wants to add a car.

Pre-Conditions:

- The room owner is already logged in.
- The room owner is inside of a room.
- No car is available in the room.

Main Success Scenario:

1. The room owner wants to add a car.
2. The room owner inputs the amount of credits to rent the car.
3. The room owner gives the system the car's credentials.
4. The system checks the car.
5. The room owner has now added the car to the room.

Alternatives:

4a. The room owner inputs incorrect car credentials.

1. An error message is displayed telling the room owner that they did not input the correct credentials to the car.
2. The room owner is redirected back to step 3.

Post-Conditions: None

Model

Using FURPS+, we can get all the requirements for this application:

Functionality

- Car must be able to use Lidar
- Users must be able to control a car
- Users must be able to rent a car
- Users must be able to view through the front camera of the car
- Users must be able to sign up
- Users must be able to login
- Users must be able to add credits to their account
- Users must be able to create a room
- Users must be able to chat with another user
- Users must be able to report another user
- Users must be able to ban another user from their room

Usability

- The application must look visually appealing

Reliability

- The application must remain online even if there are no cars online.

Performance

- Users must be able to move the car within 5 seconds

Supportability

- The application has to be available on any popular platform (Windows, Mac, Android and IPhone).

Plus

- The application must be able to be used on a browser to allow for cross platform as mentioned in Supportability
- Login's passwords must be salt hashed