A drawing of a face

Description automatically generated

ICT2101/2201 Introduction to Software Engineering

Design Specifications

Milestone 2

AY2021/2022, Trimester 1

Code A Car

Lab group P3-8

P3-8

Team Leader: Leong Jin Kuan, Joshua (2002133)

Product Developer: Loh Hui Qi (2001718)

Web Developer: Muhammad Alam Sah Bin Syed Hamzah (2001776)

Web Developer: Tse Kin Ping (2001568)

# Introduction

Describe what is your project about and explain what your project aims to achieve.

## Roles & Responsibilities

## 1.1 Product Scope

*<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals.*

*TO DO: 1-2 paragraphs describing the scope of the product. Make sure to describe the benefits associated with the product.>*

This project aims to teach users of (ages 12-16) computational thinking through gamification and subsequently, block-based programming. Through the use of a web portal, users will be able to implement functionalities and guide the way the car should go. Details such as movement speed, IR sensors functionality and direction will be shown on the web portal. These details are meant to better create a link between the webpage information and the actions reflected by the vehicle.

In learning to manipulate the robot car, users will be taught to think and plan in a logical manner. It will enable users to be better problem solvers as they navigate a set of challenges.

## 1.2 Intended Audience and Document Overview

*<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers (In your case it would probably be the “client” and the professor). Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>*

This document is intended to inform the following users on the development of the robotic car prototype and the web portal. The document contains various information regarding the background of its creation, various functionalities, and guidance on how they can be used.

### 1.2.1 Client:

To provide the client with further information on the market research of the product and the various developmental considerations behind the creation of the robotic car, Section 2.1 can be read first. Following which, section 2.2 will provide an overview of the robotic car and its interaction with the web portal. These include the sequences on how instructions are being inputted and sent to the robotic car for execution. Sections 3 and 4 would provide an in-depth look into the various requirements set forth and their subsequent implementations. Section 4.2 would set forth the security and safety aspects of the project and products. Lastly, an overview of the project timeline can be found in section 6.

Should there be a need for clarification on terminologies, Sections 1.3 and Appendix A can be consulted.

Appendix B would provide insight into the types of challenges set forth and the considerations on their implementation to teach computational thinking to the end-users.

### 1.2.2 Users:

Before operating the car, Section 4.2 is to be consulted. This will provide the user with understanding of the hidden safeguards of the robotic car and the restrictions placed upon the user’s scope of interactivity.

The following sections 3.1.1 and 3.1.2 will provide users with an overview of the web portal and an understanding of the modules on the robotic car respectively. Should any clarifications regarding terminology be required, sections 1.3 and appendix A should be referenced.

Appendix B provides specifications for the modules found on the robotic car, to be used during hardware diagnosis.

## 1.3 Definitions, Acronyms and Abbreviations

*<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.*

*TO DO: Please provide a list of all abbreviations and acronyms used in this document sorted in alphabetical order.>*

|  |  |
| --- | --- |
| Term | Definition |
| IR | Infrared |
| MAC | Media Access Control |
| SPI | Serial Peripheral Interface |
| SSID | Service Set Identifier |

## 1.4 Document Conventions

*<In general, this document follows the IEEE formatting requirements. Use Arial font size 11, or 12 throughout the document for text. Use italics for comments. Document text should be single spaced and maintain the 1” margins found in this template. For Section and Subsection titles please follow the template.*

### 1.4.1 Formatting Conventions

1. General Text: Calibri, Font Size 11, Alignment: Justified, Single Spaced
   1. Lists or pointers should include a spacing after the last paragraph
2. Headers: Arial, Font Size 20, Bolded
3. Sub-Headers: Arial, Font Size 16, Bolded
4. Image or Figure Titles: Arial, Font Size 9, Italicised
5. Page Margins: 1”
6. Comments are to be italicised

### 1.4.2 Citations

All citations, both in-text, and that within the “References” section shall adhere to IEEE formatting convention [1].

TO DO: Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. Sometimes, it is useful to divide this section to several sections, e.g., Formatting Conventions, Naming Conventions, etc.>

# Overall Description

This is the paragraph content of your section. Include any background research or literature research you have done here.

## 2.1 Product Overview

* *<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. In this part, make sure to include a simple diagram that shows the major components of the overall system, subsystem interconnections, and external interface. In this section it is crucial that you will be creative and provide as much information as possible.*
* *TO DO: Provide at least one paragraph describing product perspective. Provide a general diagram that will illustrate how your product interacts with the environment and in what context it is being used. This is not a formal diagram, but rather something that is used to illustrate the product at a high level.>*

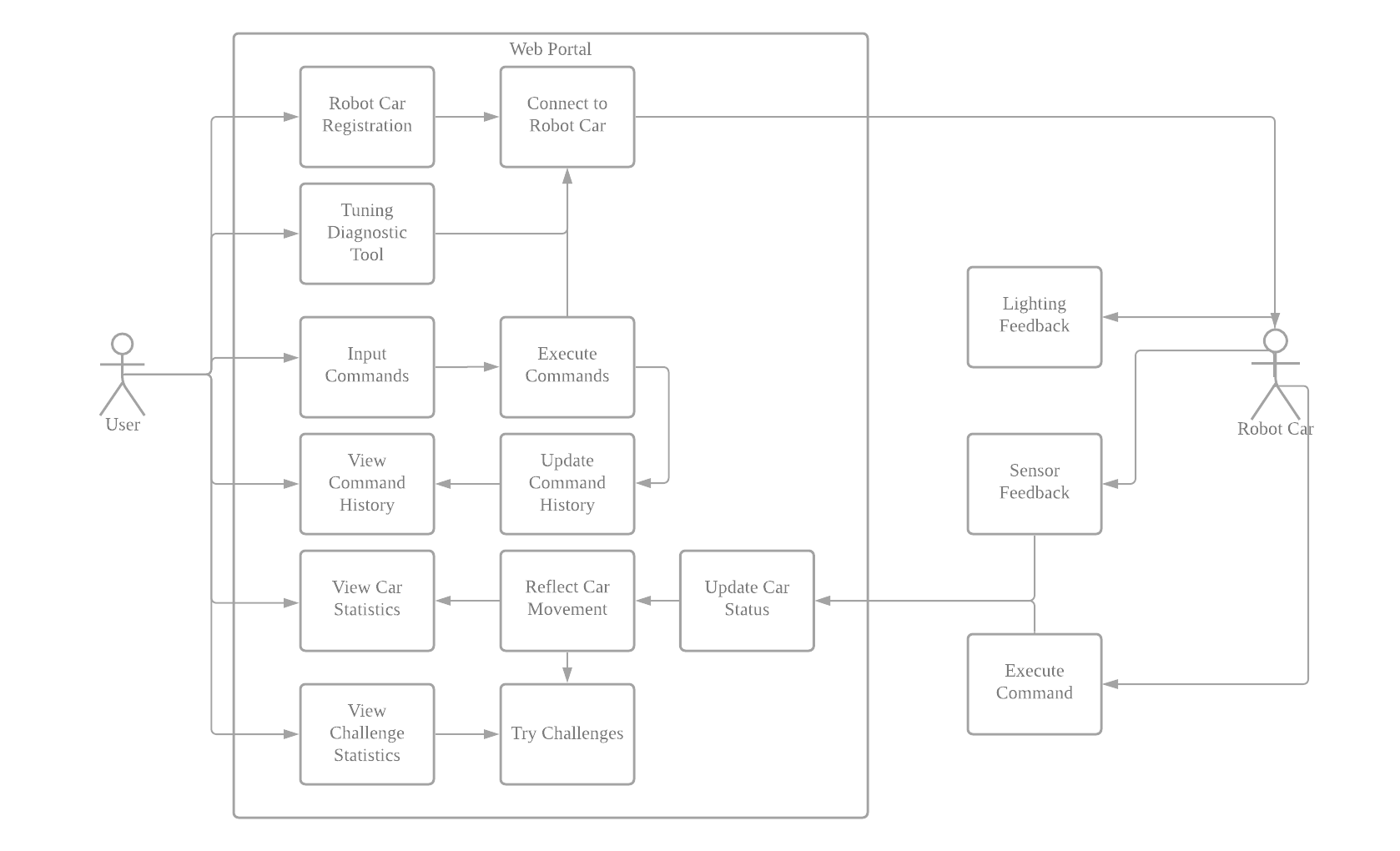


Figure shows the high-level diagram of our product

This product is a stand-alone and consists of a web portal and a robotic car, as shown in figure 1 above. The product aims to allow users to explore the ideas of computational thinking by allowing free play or using challenges which will be solved by manoeuvring of the toy car.

As the target users range from age 6-16, the web portal is designed to be accessible and minimalistic. Users accessing the web portal will be requested to input the mac address of the WiFi card on the robot car. Pressing a “Connect” button will initiate a connection and validation between the portal and the robot car. This will help our portal to filter out unrelated devices in the network and identify the desired robot car if there are multiple connected to the same network.

Upon establishing connection, the user may then access various challenges. After selecting a challenge, users will be able to create a list of commands for execution by pressing directional buttons (up, down, left, right) on the web portal. The “Run” button will then be used to send the list of commands through an API towards the robot car. The robot car will move according to the list of commands given by the user and will concurrently send the car's current status back to the web portal. The web portal will then reflect the car’s current status, such as speed, distance travelled, the wheels’ direction and connectivity status. Additionally, the executed list of commands will be added to a command history such that user may refer back to the directions the car took. A pop-up text will update the user on the car’s current actions, as well as informing them of current objectives or if the car has failed in doing so.

A tuning or diagnostic tool would be included in the web portal. This would allow users to easily diagnose the various functionalities of the car should any issues occur.

## 2.2 Product Functionality

* *<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary is needed here. These can be at the level given in the project description.>*
* *TO DO:*
* *1. Provide a bulleted list of all the major functions of the system*

Web Portal:

1. Attempt and solve challenges, robot car to reflect decisions made.
2. Interact and manoeuvre the robot car through commands.
3. View the car’s current status information and command history.
4. Allow tuning and diagnostic of robot car hardware.

Robot Car:

1. Execute the command list given by the web portal.
2. Send multiple updates back to the web portal regarding the car’s current status.
3. Interact with provided materials for extended learning.

## 2.3 Design and Implementation Constraints

* *<Describe any items or issues that will limit the options available to the developers. These might include: hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software). You can be creative here to some degree. At a minimum, you need to identify that you must use the COMET method for software design and the UML modeling language. Make sure you provide references for both. >*

## 2.4 Assumptions and Dependencies

* *<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.*
* *TO DO: Provide a short list of some major assumptions that might significantly affect your design.>*

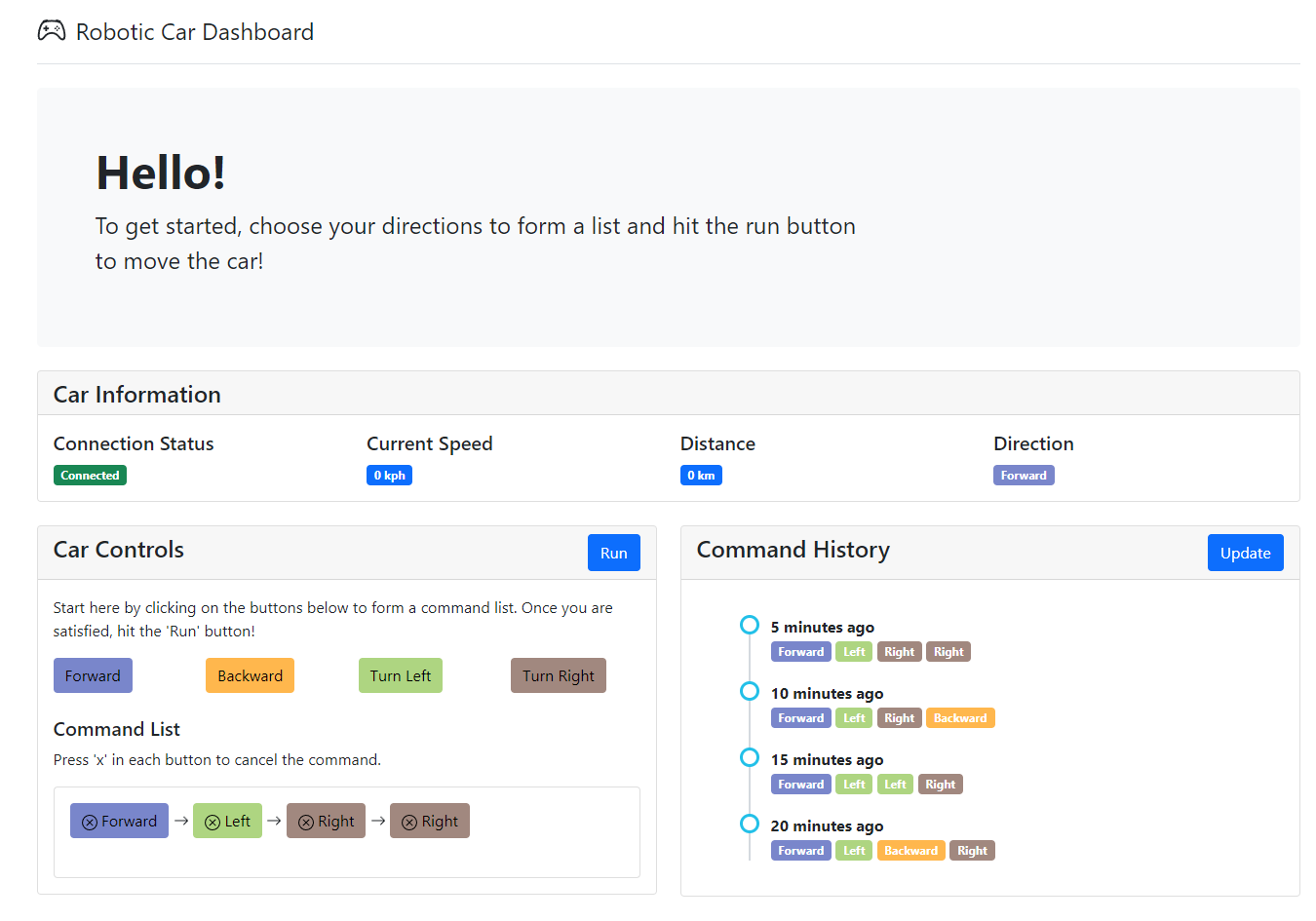
# Specific Requirements

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

*<Describe the logical characteristics of each interface between the software product and the users. For your project, you only need to be concerned with the main thermostat (not the mobile app) and can use the graphic from the project description as the basis for your user interface..*

*TO DO: Provide the graphic for the thermostat user interface and provide a basic description as to how users will interact (e.g., tough screen, menus, etc.).>*



*Figure 1.1 Dashboard*

Figure 1.1 above shows the main dashboard, which will provide the user with an overview of the controls and various statistics from the car.

1. Hello

In this section, we greet the user and display basic instructions on how to operate the website and the robotic car

1. Car Information

In this section, the user will be able to view the robotic car’s information and statistics. It mainly consists of the connection status between the web portal and the robotic car, the current speed which will be updated as the car is moving, the total distance travelled by the car since the first command list execution and lastly, the direction of where the car is currently facing, which gets updated as the car is executing the command list.

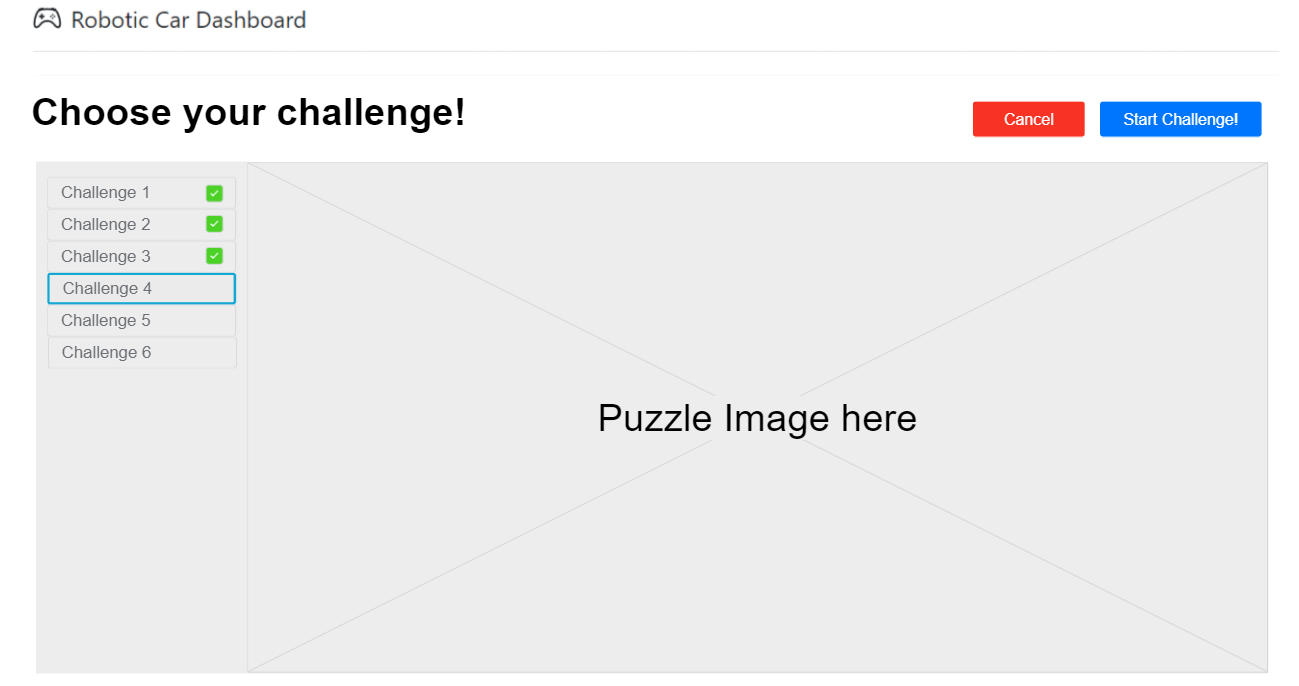
1. Car Controls

In this section, the user will be able to create the command list by clicking on the directional buttons. For every click, the direction chosen will be added into the end of the list in the “Command List” box. This is so that the user is able to follow through the list and make amendments, when necessary, by pressing "x” in each button. Once satisfied, the user may click on the “Run” button to execute the command list and move the robotic car.

1. Command History

In this section, it shows all of the user’s previous executed command list in a timeline manner. The list is sorted by the most recent command list first for the ease of user to refer.

1. Challenges



*Figure 1.2 shows the Challenge Screen in our web portal*

Figure 1.2 above shows the challenge page where users will be able to select their desired challenges.

In this challenge screen, the user is able to view all the available challenges as well as see the challenge that he/she has completed which is reflected by the green tick mark beside the challenge. Once a challenge has been selected, the user may then click on the “Start Challenge” button to be redirected back to the dashboard with the new challenge in progress.

*The figures are a work in progress and may not reflect the final product.*

### 3.1.2 Hardware Interfaces – Update with buzzer (if receive)

*<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware. You are not required to specify what protocols you will be using to communicate with the hardware, but it will be usually included in this part as well.*

*TO DO: Please provide a short description of the different hardware interfaces. This can simply be a list of the devices you must interact with at this point. For sensors, you can assume that each has a basic “read” interface to get the current values. Each sensor in this system uses English units.>*

1. MSP432 Board

This will provide the computing and main functionalities of the robot car. It interacts with the following devices and provides data to be sent back to the web portal via the WiFi Serial Transceiver Module. There is a RGB LED on-board which will be used to send real time feedback to the user, its functionalities can be customised.

1. WiFi Serial Transceiver Module

To allow communication with the web portal. This will allow receive the MSP432 to receive input commands from the web portal via SPI.

1. Motor Driver

The MSP432 board will interact with this module to initiate movement for the robot car. This will be used to provide and coordinate directional movement. To be used in conjunction with the IR Optical Speed Sensor.

1. IR Optical Speed Sensor

This device will provide users with movement information on the car. It will be used in conjunction with the black slotted discs on the robot car to evaluate the directional position of the robot car as well as current speed. The output will be formatted to meters per second.

1. Ultrasonic Ranging Module

To provide users with the ability to evaluate distances of objects or boundaries from the front of the car. Output will be reading from the interface to get current distance. The output will be formatted into meters.

1. IR Line Tracking Module

This will be used to provide the robot car with line tracking functionalities.

1. IR Strips

These will be used in conjunction with the IR line tracking module to provide added interactivity between the robot car and the environment.

### 3.1.3 Software Interfaces

*<Describe the connections between this product and other specific software components (in your case, just the mobile app that can send commands).>*

*In Progress: To be filled up once the WiFi Configuration and communication between car and web portal is established*

## 3.2 Functional Requirements – Move new car functional requirement

< Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform. This section is the direct continuation of section 2.2 where you have specified the general functional requirements. Here, you should list in detail the different product functions. >

Evaluate based on **who, input, output, alt scenarios, exceptions**

Web portal in-depth functional requirements:

1. The system shall allow the user to register their robotic car using the MAC address of the robotic car
2. The system shall be able to connect to the robotic car via local WiFi using the indicated MAC address
3. The system shall be able to maintain a consistent connection while the robot car is turned on
4. The system will prompt user if the car is unable to be reached
5. Users shall be able to initiate a diagnostic procedure on robot car hardware
6. Users shall be able to create a list of commands using the directional buttons
7. Users shall be able to initiate execution of the command list by clicking the run button
8. The system shall be able to send the command list to the robotic car via local WiFi
9. Users shall be able to view connectivity status with the car on demand
10. Users shall be able to view the current sensor data from the robot car
11. Users shall be able to view live updates of the car’s current manoeuvring action
12. Users shall be able to view a history of commands that has been executed
13. Users shall be able to view challenges
14. Users shall be able to attempt and complete challenges
15. The system shall be able to check if a challenge is completed
16. Users shall be able to create custom challenges

Robotic car in-depth functional requirements:

1. The system shall be able to receive and send information from the web portal via the local WiFi
2. The system shall be able to maintain a stable connection to the web portal
3. The system shall be able to receive a list of commands given by the web portal
4. The system shall be able to execute the commands listed
5. The system shall be able to send updates to the web portal regarding its current status and various sensors
6. The system shall be able to detect obstacles such as IR strips using the ultrasonic sensor and IR tracker
7. The system shall be able to determine if the user successfully reached the objective or failed to do so and send subsequent information to the web portal
8. Users shall be able to interact with the robotic car outside the web portal via the onboard switches

## 3.3 Use Case Model

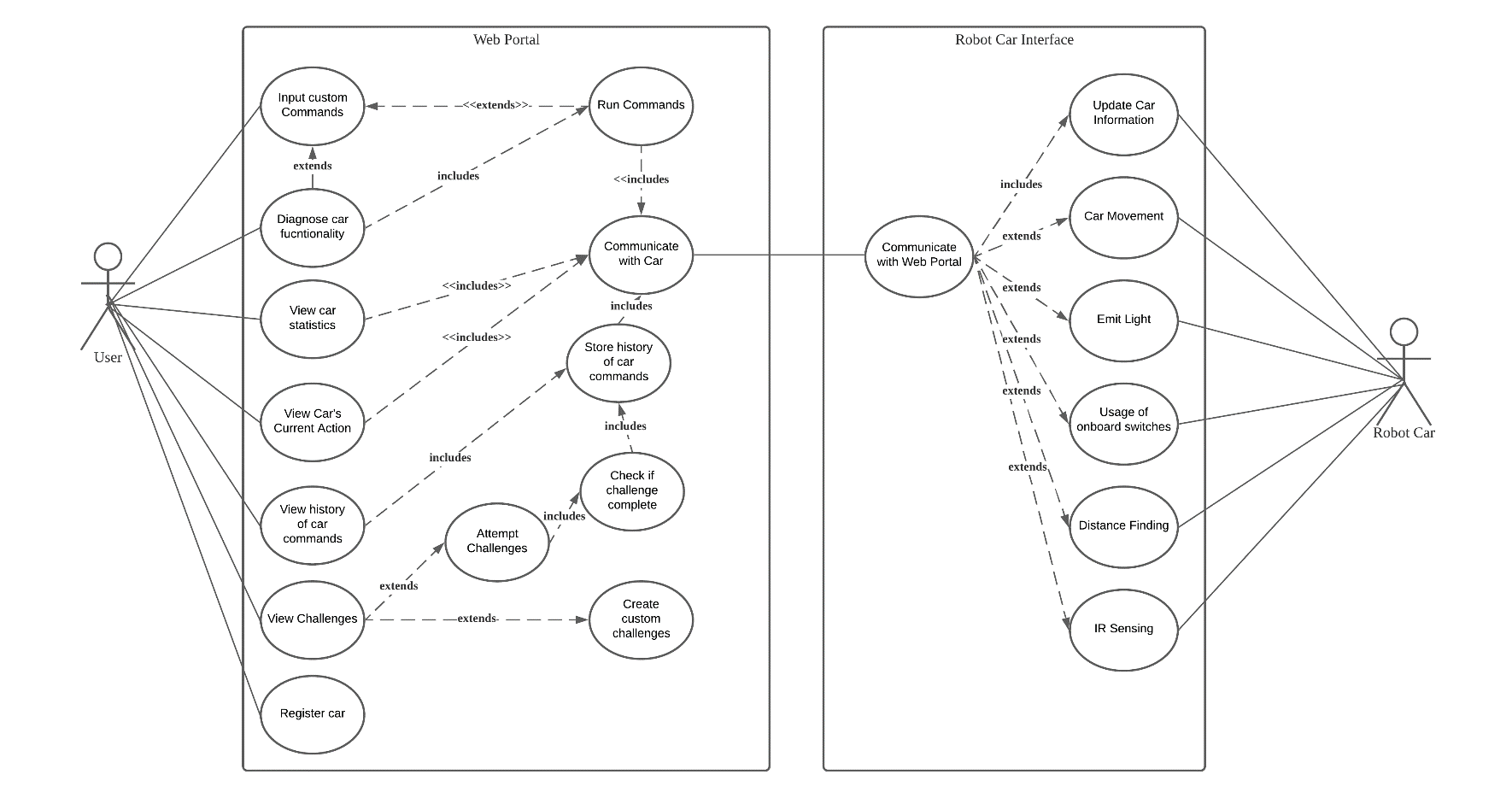


Figure 3 shows the Use Case Diagram

### 3.3.1 UC1 - Register Car

|  |  |
| --- | --- |
| **Author** |  |
| **Purpose** |  |
| **Requirements Traceability** |  |
| **Priority** |  |
| **Preconditions** |  |
| **Post conditions** |  |
| **Actors** |  |
| **Flow of Events** |  |

|  |
| --- |
| **Author: Md Alam Sah** |
| **Purpose:** To register and connect the car with the web portal |
| **Requirements Traceability:**  F1: The system shall allow the user to register their robotic car using the MAC address of the robotic car  F2: The system shall be able to connect to the robotic car via local WiFi using the indicated MAC address  F3: The system shall be able to maintain a consistent connection while the robot car is turned on  F4: The system will prompt user if the car is unable to be reached within 3 tries, or after 10 seconds |
| **Priority:** High |
| **Preconditions:**   1. The user must have a robotic car 2. Both web portal and robotic car must be connected to the same WiFi 3. User should know the MAC address of the robotic car |
| **Post conditions:** The robotic car will successfully be connected to the web portal. |
| **Actors:** User, Robot Car |
| **Flow of Events**  Basic Flow:   1. User enters the mac-address of his/her robotic car in the web portal 2. User clicks submit button 3. System will find for the robotic car with the mac-address 4. Once found, the system will initialize a connection to the robotic car interface 5. System will then reflect a “Connected” status to the portal dashboard   Alternative Flow:   1. System fails to initialize a connection to the robotic car    1. System will reflect an error message saying that it is unable to find a robotic car with the mac-address given 2. System is unable to find robotic car due to invalid mac address    1. System will reflect an error message saying the mac-address is invalid   Exceptions:   1. System loses the WiFi connection    1. Once reconnected, the system will repeat steps 1-5 |
| **Total Number of Transactions:** 6 |
| **Use Case Complexity:** Average |

### 3.3.2 UC2 – Input Custom Commands

|  |  |
| --- | --- |
| **Author** |  |
| **Purpose** |  |
| **Requirements Traceability** |  |
| **Priority** |  |
| **Preconditions** |  |
| **Post conditions** |  |
| **Actors** |  |
| **Flow of Events** |  |

|  |
| --- |
| **Author: Md Alam Sah** |
| **Purpose:** Allows the user to choose their own directional commands for the robotic car |
| **Requirements Traceability:**  F6: Users shall be able to create a list of commands using the directional buttons |
| **Priority:** High |
| **Preconditions:**   1. The robotic car must be connected to the web portal |
| **Post conditions:** The user will have formed a command list for the toy car to run |
| **Actors:** User |
| **Extends:** UC3 – Run Commands, UC5 – Diagnose Car Functionality |
| **Flow of Events**  Basic Flow:   1. The user clicks on the directional buttons sequentially according to how he/she will like the car to move 2. System will display the directional buttons clicked so user can view the command list   Alternative Flow: -  Exceptions:   1. System is unable to capture the buttons clicked to display them in the list    1. System will inform the user saying that there is an error in capturing the directions and prompt user to refresh the portal |
| **Total Number of Transactions:** 3 |
| **Use Case Complexity:** Simple |

### 3.3.3 UC3 – Run Commands

|  |  |
| --- | --- |
| **Author** |  |
| **Purpose** |  |
| **Requirements Traceability** |  |
| **Priority** |  |
| **Preconditions** |  |
| **Post conditions** |  |
| **Actors** |  |
| **Flow of Events** |  |

|  |
| --- |
| **Author: Md Alam Sah** |
| **Purpose:** Sends the command list to the API connected to the robotic car interface |
| **Requirements Traceability:**  F8: The system shall be able to send the command list to the robotic car via local WiFi |
| **Priority:** High |
| **Preconditions:**   1. The robotic car must be connected to the web portal 2. The user must have formed a command list according to UC2 3. The user must have clicked on the “Run” button |
| **Post conditions:** The API will have the command list created by the user |
| **Actors:** - |
| **Extends:** UC2 – Input Custom Commands |
| **Flow of Events**  Basic Flow:   1. System will validate the command list for any invalid inputs 2. System will sanitize and forward the list to the API   Alternative Flow:   1. System detects an invalid/malicious input in the command list    1. System returns an error message to the user saying that there was an error in submitting the directions and prompts the user to retry again   1.3 Exceptions: - |
| **Includes:** UC5 – Diagnose Car Functionality |
| **Total Number of Transactions:** 1 |
| **Use Case Complexity:** Simple |

### 3.3.4 UC4 – Communicate with Car

|  |  |
| --- | --- |
| **Author** |  |
| **Purpose** |  |
| **Requirements Traceability** |  |
| **Priority** |  |
| **Preconditions** |  |
| **Post conditions** |  |
| **Actors** |  |
| **Flow of Events** |  |

|  |
| --- |
| **Author: Md Alam Sah** |
| **Purpose:** Sends/Receives data to/from the Robotic Car Interface API |
| **Requirements Traceability:**   1. Users shall be able to initiate a diagnostic procedure on robot car hardware within 10 seconds 2. The system shall be able to send the command list to the robotic car via local WiFi 3. Users shall be able to view the current sensor data from the robot car 4. Users shall be able to view live updates of the car’s current manoeuvring action within 10 seconds 5. Users shall be able to view a history of commands that has been executed |
| **Priority:** High |
| **Preconditions:**   1. The robotic car must be connected to the web portal 2. The user must have initiated a function that requires communication with the robotic car |
| **Post conditions:** The robotic car or web portal interface will have the requested data |
| **Actors:** - |
| **Flow of Events**  Basic Flow:   1. Receives data from sender to be transferred to the receiver 2. Use the appropriate method to forward the data accordingly 3. Update sender on success   Alternative Flow:   1. System unable to forward the data    1. System returns an error message to the sender saying how it is unable to send the data and the reason for it so that sender class may update user   Exceptions:   1. Lost communication with car while sending/receiving data    1. System will reply to the sender class with the error message in which the sender class will prompt the user to retry |
| **Includes:** UC3 – Run Commands, UC5 – View Car Statistics, UC6 – View Car Statistics, UC8 – View Car’s Current Action |
| **Total Number of Transactions:** 3 |
| **Use Case Complexity:** Medium |

### 3.3.5 UC5 – View Car Statistics

|  |  |
| --- | --- |
| **Author** |  |
| **Purpose** |  |
| **Requirements Traceability** |  |
| **Priority** |  |
| **Preconditions** |  |
| **Post conditions** |  |
| **Actors** |  |
| **Flow of Events** |  |

|  |
| --- |
| **Author: Md Alam Sah** |
| **Purpose:** To allow the user to view the car’s statistics on the portal |
| **Requirements Traceability:**   1. F6: The system shall allow the user to view dashboard which consists of the car’s current status information |
| **Priority:** High |
| **Preconditions:**   1. The robotic car must be connected to the web portal 2. The user must have executed a command list for the robotic car to move |
| **Post conditions:** The robotic car’s statistics will be displayed |
| **Actors:** User |
| **Flow of Events**  1.1 Basic Flow:   1. Receive car statistics from the web portal API (UC4) 2. System displays the statistics on the dashboard for the user to view   1.2 Alternative Flow:   1. There are already statistics being displayed on the dashboard    1. System will update the statistics displayed with the newly received statistics   1.3 Exceptions: - |
| **Includes:** UC4 – Communicate with Car |
| **Total Number of Transactions:** 2 |
| **Use Case Complexity:** Simple |

### 3.3.6 UC6 – Store History of Car Commands

|  |
| --- |
| **Author: Md Alam Sah** |
| **Purpose:** To obtain and store information regarding the history of car commands user have executed |
| **Requirements Traceability:**   1. F8: The system shall allow the user to view a history of commands that has been executed |
| **Priority:** High |
| **Preconditions:**   1. The robotic car must be connected to the web portal 2. The user must have executed a command list for the robotic car to move |
| **Post conditions:** The command list history will be stored here to be called by UC9 – View History of Car Commands |
| **Actors:** - |
| **Flow of Events**  1.1 Basic Flow:   1. Receive executed command list from the web portal API (UC4) 2. System stores the list in the history list   1.2 Alternative Flow: -  1.3 Exceptions: - |
| **Includes:** UC9 – View History of Car Commands, UC12 – Check if Challenge Complete |
| **Total Number of Transactions:** 1 |
| **Use Case Complexity:** Simple |

### 3.3.7 UC7 – Diagnose Car Functionality

|  |
| --- |
| **Author: Joshua Leong** |
| **Purpose:** To provide users with a quick and easy tool to diagnose car functionality |
| **Requirements Traceability:**  The system shall allow the user to view dashboard which consists of the car’s current status information  The system shall allow the user to diagnose hardware sensors of the robot car |
| **Priority:** High |
| **Preconditions:**   1. The robotic car must be connected to the web portal |
| **Post conditions:** The user should be able to diagnose car functionality |
| **Actors:** - |
| **Flow of Events**   * 1. Basic Flow:  1. User initiates diagnosis process with the car 2. System presents user with list of hardware for diagnosis 3. User selects desired area for diagnosis 4. System communicates with car (UC4) 5. System indicates to user action has been performed    1. Alternative Flow:   5.1 If any error were to occur, the system would indicate that to the user via error message on the web portal  1.3 Exceptions: - |
| **Includes:** |
| **Total Number of Transactions: 3** |
| **Use Case Complexity:** Simple |

### 3.3.8 UC8 – View Car’s Current Action

|  |
| --- |
| **Author: Tse Kin Ping** |
| **Purpose:** To view current state of car |
| **Requirements Traceability:**   1. F10: Users shall be able to view current sensor data from the robot car 2. F11: Users shall be able to view live updates of the car’s current manoeuvring action within 10 seconds |
| **Priority:** High |
| **Preconditions:**   1. The robot car must be connected to the web portal |
| **Post conditions:** The current state of car should be displayed on the web portal |
| **Actors:** - |
| **Flow of Events**  1.1 Basic Flow:   1. System communicates with car (UC4) 2. System receives current car information 3. System displays current car status   1.2 Alternative Flow: -  1.3 Exceptions: - |
| **Includes:** |
| **Total Number of Transactions:** 1 |
| **Use Case Complexity:** Simple |

### 3.3.9 UC9 – View history of car commands

|  |
| --- |
| **Author: Tse Kin Ping** |
| **Purpose:** To obtain display a history of commands that user issued. |
| **Requirements Traceability:**   1. F12: The system shall allow the user to view a history of commands that has been executed |
| **Priority:** High |
| **Preconditions:**   1. The robotic car must be connected to the web portal 2. The user must have executed at list one command to the robotic car |
| **Post conditions:** The command list history will be obtained and displayed to the user. |
| **Actors:** User |
| **Flow of Events**  1.1 Basic Flow:   1. User selects to view history of car commands 2. System reads in the history of car commands stored (UC6) 3. System displays history of car commands   1.2 Alternative Flow:   1. User refreshes view history of car commands 2. System reads in the latest history of car commands stored (UC6) 3. System displays a refreshed list of car commands history   1.3 Exceptions: - |
| **Includes:** UC6 – View History of Car Commands |
| **Total Number of Transactions: 2** |
| **Use Case Complexity:** Average |

### 3.3.10 UC10 – View Challenges

|  |
| --- |
| **Author: Tse Kin Ping** |
| **Purpose:** To obtain and display the challenges available |
| **Requirements Traceability:**   1. F13: Users shall be able to view challenges |
| **Priority:** High |
| **Preconditions:**   1. The robot car must be connected to the web portal |
| **Post conditions:** The user shall be able to view challenges and select which to attempt |
| **Actors:** - |
| **Flow of Events**  1.1 Basic Flow:   1. User selects to view challenges 2. System reads in challenges 3. System displays challenges   1.2 Alternative Flow: -  1.3 Exceptions: - |
| **Includes: -** |
| **Total Number of Transactions:** 1 |
| **Use Case Complexity:** Average |

### 3.3.11 UC11 – Attempt Challenges

|  |
| --- |
| **Author: Tse Kin Ping** |
| **Purpose:** To allow user to create a list of commands to attempt and complete the challenge |
| **Requirements Traceability:**   1. F6: Users shall be able to create a list of commands using directional buttons 2. F7: Users shall be able to initiate execution of the command list by clicking the run button 3. F8: The system shall be able to send the command list to the robotic car via local WiFi 4. F11: Users shall be able to view live updates of the car’s current manoeuvring action within 10 seconds 5. F14: Users shall be able to attempt and complete challenges |
| **Priority:** High |
| **Preconditions:**   1. The robotic car must be connected to the web portal 2. The user must have selected a challenge to attempt |
| **Post conditions:** The challenge shall be attempted and system will check if challenge is completed (UC12) |
| **Actors:** - |
| **Flow of Events**  1.1 Basic Flow:   1. User create a list of commands to be executed 2. User run the commands 3. System sends the list of commands to robot car (UC4) 4. System displays current car status (UC8) 5. System checks if challenge completed (UC12)   1.2 Alternative Flow: -  1.3 Exceptions: - |
| **Includes:** UC12 – Check if Challenge Complete |
| **Total Number of Transactions:** 2 |
| **Use Case Complexity:** Complex |

### 3.3.12 UC12 – Check if challenge complete

|  |
| --- |
| **Author: Tse Kin Ping** |
| **Purpose:** To check if the challenge is completed after user executed a list of commands |
| **Requirements Traceability:**   1. F14: Users shall be able to attempt and complete challenges 2. F15: The system shall be able to check if a challenge is completed |
| **Priority:** High |
| **Preconditions:**   1. The robotic car must be connected to the web portal 2. The user must have attempted the challenge (UC11) |
| **Post conditions:** The command list history will be stored (UC6 - Store history of car commands) |
| **Actors:** - |
| **Flow of Events**  1.1 Basic Flow:   1. System checks if challenge is completed 2. System stores history of car commands (UC6)   1.2 Alternative Flow: -  1.3 Exceptions: - |
| **Includes:** UC6 – Store history of car commands |
| **Total Number of Transactions:** 2 |
| **Use Case Complexity: Simple** |

### 3.3.13 UC13 – Create custom challenges

|  |
| --- |
| **Author: Tse Kin Ping** |
| **Purpose:** User to be able to create their own custom challenges |
| **Requirements Traceability:**   1. F16: Users shall be able to create custom challenges |
| **Priority:** High |
| **Preconditions:**   1. The robotic car must be connected to the web portal |
| **Post conditions:** The custom challenge will be saved and can be view in UC10 – View Challenges |
| **Actors:** - |
| **Flow of Events**  1.1 Basic Flow:   1. User selects to create custom challenge 2. User sets custom challenge 3. User select to save custom challenge 4. System stores custom challenge   1.2 Alternative Flow: -  1.3 Exceptions: - |
| **Includes:** UC10 – View Challenges |
| **Total Number of Transactions:** 1 |
| **Use Case Complexity:** Complex |

### 3.3.14 UC14 – Communicate with Web Portal

|  |
| --- |
| **Author: Loh Hui Qi** |
| **Purpose:** Send/Receive data to/from the Web Portal |
| **Requirements Traceability:**   1. F17: The system shall be able to receive and send information from the web portal via the local WiFi 2. F18: The system shall be able to maintain a stable connection to the web portal |
| **Priority:** High |
| **Preconditions:**   1. The web portal must be connected to the robotic car 2. Both web portal and robotic car must be connected to the same WiFi |
| **Post conditions:** The robotic car will receive/send data from/to web portal |
| **Actors:** - |
| **Flow of Events**  1.1 Basic Flow:   1. Receive data from sender to be transferred to the robotic car 2. Use appropriate method to forward the data 3. Update sender on success   1.2 Alternative Flow:  2. System unable to forward the data  2.1 System prompts an error message to sender the reason why unable to read data and ask to try again  1.3 Exceptions:   1. Lost communication with web portal while sending/receiving data    1. System prompts an error message to sender to retry connection |
| **Total Number of Transactions:** 3 |
| **Use Case Complexity:** Simple |

### 3.3.15 UC15 - Update Car Information

|  |
| --- |
| **Author: Loh Hui Qi** |
| **Purpose:** Update robotic car information to web portal |
| **Requirements Traceability:**   1. F17: The system shall be able to receive and send information from the web portal via the local WiFi 2. F21: The system shall be able to send updates to the web portal regarding its current status and various sensors |
| **Priority:** High |
| **Preconditions:**   1. The web portal must be connected to the robotic car 2. Both web portal and robotic car must be connected to the same WiFi 3. The robotic car must have sensors data of UC16 to UC20 |
| **Post conditions:** The robotic car information will successfully be updated to the web portal |
| **Actors:** Robot Car |
| **Flow of Events**  1.1 Basic Flow:   1. The robotic car receives data from sensors (UC16 to UC20) 2. The system transfers receive data to the web portal 3. The system updates the robotic car information about its current status in the web portal   1.2 Alternative Flow:  2. System unable to transfer the data to web portal  2.1 System prompts an error message to the reason why unable to transfer data and ask to try again  1.3 Exceptions:  2. Lost communication with web portal while transferring data  2.1 System prompts an error message to retry connection |
| **Includes:** UC14 – Communication with Web Portal |
| **Total Number of Transactions:** 3 |
| **Use Case Complexity:** Simple |

### 3.3.16 UC16 – Car Movement

|  |
| --- |
| **Author: Loh Hui Qi** |
| **Purpose:** Detect where the robotic car is running |
| **Requirements Traceability:**   1. F19: The system shall be able to receive a list of commands given by the web portal 2. F20: The system shall be able to execute the commands listed |
| **Priority:** High |
| **Preconditions:**   1. The web portal must be connected to the robotic car 2. Both web portal and robotic car must be connected to the same WiFi |
| **Post conditions:** The robotic car will successfully run according to the command received |
| **Actors:** Robot Car |
| **Extends:** UC14 – Communication with Web Portal |
| **Flow of Events**  1.1 Basic Flow:   1. The robotic car receives a list of commands from the web portal 2. The system executes the list of commands 3. The robotic car will successfully run according to the execution   1.2 Alternative Flow:  2.The system is unable to execute the list of commands  2.1 System prompts an error message the reason why unable to execute the data and ask to try again  1.3 Exceptions:   1. Lost communication with web portal while receiving data    1. System prompts an error message to retry connection |
| **Total Number of Transactions:** 3 |
| **Use Case Complexity:** Simple |

### 3.3.17 UC17 – Emit Light

|  |
| --- |
| **Author: Joshua Leong** |
| **Purpose:** To allow the user to operate the LEDs on the robotic car |
| **Requirements Traceability:**   1. The system shall be able to execute the commands listed 2. The system shall be able to send updates to the web portal regarding its current status and various sensors |
| **Priority:** High |
| **Preconditions:**   1. The web portal must be connected to the robotic car 2. Both web portal and robotic car must be connected to the same WiFi |
| **Post conditions:** The LED on the robotic car will emit light based on the commands from the user |
| **Actors:** Robot Car |
| **Extends:** UC14 – Communication with Web Portal |
| **Flow of Events**   * 1. Basic Flow:  1. The system receives the list of commands from the web portal via the local WiFi 2. The system will interpret and execute the commands 3. The respective LED will light up according to the commands 4. The system will indicate execution of commands to the web portal via the local WiFi    1. Alternative Flow: -    2. Exceptions:    3. The system cannot detect the LED    4. The system will send the error message to the web portal |
| **Total Number of Transactions:** 2 |
| **Use Case Complexity:** Simple |

### 3.3.18 UC18 – Usage of onboard Switches

|  |
| --- |
| **Author: Joshua Leong** |
| **Purpose:** To provide the user with additional interactivity with the robotic car via the two built-in switches |
| **Requirements Traceability:**   1. Users shall be able to interact with the robotic car outside the web portal via the onboard switches |
| **Priority:** High |
| **Preconditions:**   1. The web portal must be connected to the robotic car 2. Both web portal and robotic car must be connected to the same WiFi |
| **Post conditions:** The user will be able to interact with the robotic car via switches |
| **Actors:** User, Robot Car |
| **Extends:** UC14 – Communication with Web Portal |
| **Flow of Events**   * 1. Basic Flow:  1. The system receives the list of commands from the web portal via the local WiFi 2. The system will interpret and execute the commands 3. The user will be able to interact with the onboard switches 4. The system will indicate execution of commands to the web portal via the local WiFi    1. Alternative Flow: -    2. Exceptions: - |
| **Total Number of Transactions:** 1 |
| **Use Case Complexity:** Simple |

### 3.3.19 UC19 – Distance Finding

|  |
| --- |
| **Author:** Joshua Leong |
| **Purpose:** To allow the user to gauge distances from the robotic car to its environment |
| **Requirements Traceability:**   1. The system shall be able to send updates to the web portal regarding its current status and various sensors 2. The system shall be able to detect obstacles such as IR strips using the ultrasonic sensor and IR tracker |
| **Priority:** High |
| **Preconditions:**   1. The user must have a robotic car 2. Both web portal and robotic car must be connected to the same WiFi |
| **Post conditions:** The robotic car will measure distances between the system and objects and send subsequent data to the web portal. |
| **Actors:** Robot Car |
| **Extends:** UC14 – Communication with Web Portal |
| **Flow of Events**   * 1. Basic Flow:  1. The system receives the list of commands from the web portal via the local WiFi 2. The system will interpret and execute the commands 3. The ultrasonic sensor will detect and measure distances between the robot car and objects 4. The system will indicate resulting data from the sensors to the web portal 5. The system will indicate execution of commands to the web portal    1. Alternative Flow: -    2. Exceptions: - |
| **Total Number of Transactions:** 1 |
| **Use Case Complexity:** Simple |

### 3.3.20 UC20 – IR Sensing

|  |
| --- |
| **Author:** Joshua Leong |
| **Purpose:** To allow the user more interactivity with the robotic car using IR strips |
| **Requirements Traceability:**   1. The system shall be able to send updates to the web portal regarding its current status and various sensors 2. The system shall be able to detect obstacles such as IR strips using the ultrasonic sensor and IR tracker |
| **Priority:** High |
| **Preconditions:**   1. The user must have a robotic car 2. Both web portal and robotic car must be connected to the same WiFi |
| **Post conditions:** The robotic car will be able to interact with the IR Sensing |
| **Actors:** Robot Car |
| **Extends:** UC14 – Communication with Web Portal |
| **Flow of Events**   * 1. Basic Flow:  1. The system receives the list of commands from the web portal via the local WiFi 2. The system will interpret and execute the commands 3. The ultrasonic sensor will detect and measure distances between the robot car and objects 4. The system will indicate resulting data from the sensors to the web portal 5. The system will indicate execution of commands to the web portal    1. Alternative Flow: -    2. Exceptions: - |
| **Total Number of Transactions:** 1 |
| **Use Case Complexity:** Simple |

## 3.4 Class Diagram

## 3.5 Components Diagram

## 3.6 Communication Diagram

## 3.6 Architecture (Base on slides)

# Other Non-functional Requirements

## 4.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.

TODO: Provide performance requirements based on the information you collected from the client/professor. For example, you can say “P1. The secondary heater will be engaged if the desired temperature is not reached within 10 seconds”>

### 4.1.1 Web Portal Non-Functional

* The website’s load time should not be more than one second for users
* The connection time with the robotic car should not exceed 3 minutes
* Users should be able to access and use the web portal 100% of the time without failure
* Web portal shall complete the sending of the command list to robotic car in 10 seconds
* Should any updates be implemented, the user shall be informed 3 days in advance
* If any crashes occur, the website should be able to be restarted within 1 hour of downtime
* The website’s interface has to be user-friendly and easy to understand

### 4.1.2 Robotic Car Non-Functional

* Robotic car shall communicate with web portal via WIFI
* Upon receiving the command list, the robotic car shall execute command list in 5 seconds
* Robotic car shall send/receive command list in 10 seconds
* Robotic car IR line tracking module sensor shall detect range between 0 to 3 cm
* Robotic car ultrasonic ranging module sensor shall measure the distance of a target object between 2 cm to 4 metres

## 4.2 Safety and Security Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied. Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements.

TODO:

• Provide safety/security requirements based on your interview with the client - again you may need to be somewhat creative here. At the least, you should have some security for the mobile connection.

### 4.2.1 User Safety and Operability

As this product is expected to be used by young users, it is important to ensure that the product will not compromise the safety of the users. To achieve this, it is important to allow users the ability to control the car’s speed, provide users with clear lightings on the car to indicate the actual state of the car, as well as collision detection functions to prevent the car from colliding into any objects.

The car’s collision detection function should enable the car to detect any objects that is obstructing its projected path and if the object is not removed from the path, the car should stop at a small distance away from the object. This distance is set in the program, and it will override any user commands and will remain stationary until the obstructing object is removed or the car’s path is changed to avoid the object. With this function, damages to the car and any surrounding objects can be prevented. Furthermore, this will also prevent any injuries that could be caused if the obstruction was caused by the user or anyone around.

### 4.2.2 Security

As the product is required to work wirelessly using WiFi connection, it is therefore important to ensure that the connection between the remote-controlled device and the control device is secured. An insecure connection between both devices may result in connection being intercepted by others intentionally or unintentionally [5]. To create a secure connection, a mutual trust must be achieved by both devices, and this can be done by setting a strong password which can be reset by the user. Furthermore, setting a trusted MAC address for connection will ensure that the connection will only be successful if both devices’ MAC addresses are trusted.

## 4.3 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.

TODO: Use subsections (e.g., 4.3.1 Reliability, 4.3.2 Adaptability, etc…) provide requirements related to the different software quality attributes. Base the information you include in these subsections on the material you have learned in the class. Make sure, that you do not just write “This software shall be maintainable…” Indicate how you plan to achieve it, & etc…Do not forget to include such attributes as the design for change (e.g. adapting for different sensors and heating/AC units, etc.). Please note that you need to include at least 2 quality attributes. You can Google for examples that may pertain to your system.>

# Testing and Evaluation

You would have to draw the full System State Diagram that represents the dynamics of your entire web app/portal, but to help us focus, please do this:

1. choose 2 meaningful use cases from your use case diagram, and

2. highlight, in the report, which states and transitions in the full state diagram reflects those 2 chosen use cases.

## 5.1 System State Diagram

## 5.2 System Test Cases (User Acceptance Testing)

# Work Distribution & Plan

The AGILE model is employed for this project as it suits small to medium-sized projects with new requirements. In addition, as the technology is new for the team members working on the robot car, the breaking down of the system into smaller modules to be tackled in sprints would ease the number of uncertainties. As the team members function in pairs, communication between the sub-teams would facilitate better communication and cooperation.

In addition, as the project is time-critical, rapid prototyping will be crucial for testing of the final product. The subsequent feedback will drive team discussion to allow for early corrections on the design and its subsequent implementation.

We decided to use Gantt Chart, as shown in figure 4 below, to monitor our progress. This helped us to schedule, manage and monitor the tasks for our project at the same time, prevents us from missing out on any unassigned task.

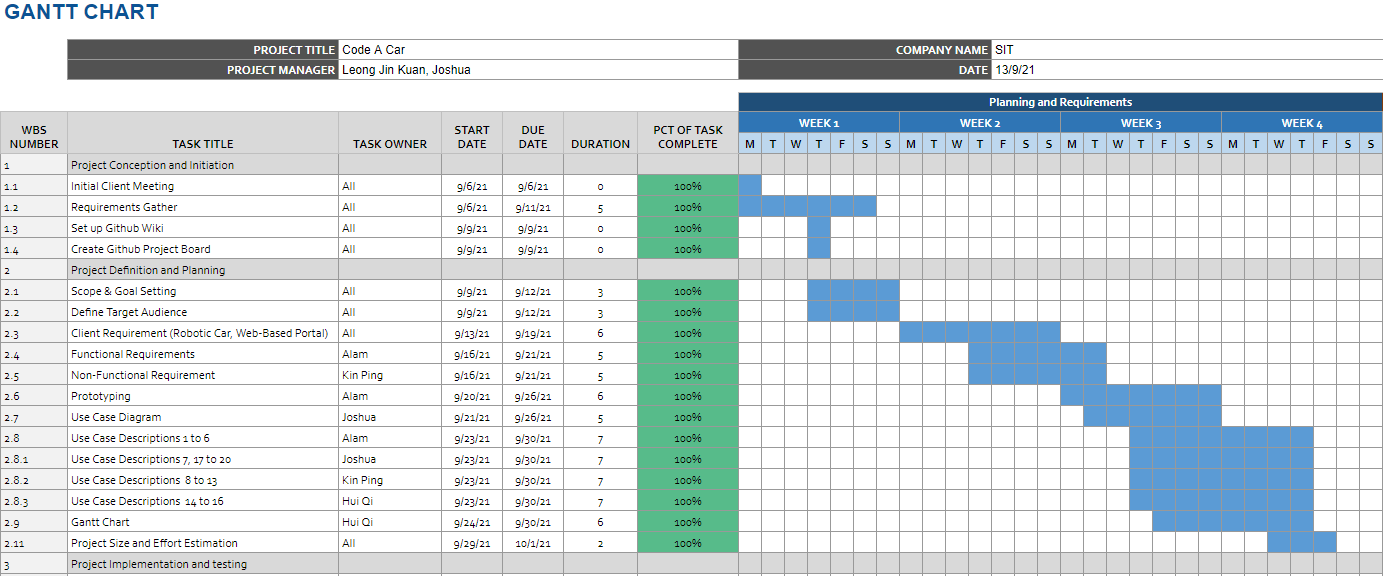


Figure 4 shows the Gantt Chart of Milestone 1

# Conclusion

Summarise the project and highlight the key features and timeline required.

Requirements Estimation:  
For our project’s requirement estimation, we decided to implement the use case points technique. According to our use case descriptions and the calculations done, we came up with the following summarised table.

|  |  |
| --- | --- |
| Title | Results |
| Use Case & Actors | Unadjusted Use Case Weight: 105  Actors Weight: 46  Total Unadjusted Weight: 151 |
| Complexity Factors | Total Weight: 38  TCF: 0.98 |
| Environmental Factors | Total Weight: 24  EF: 0.68 |
| Total UCP | 151\*0.98\*0.68 = 100.63 |

The table for each factor of this use case point can be found in Appendix A.

# Individual Reflections

# References

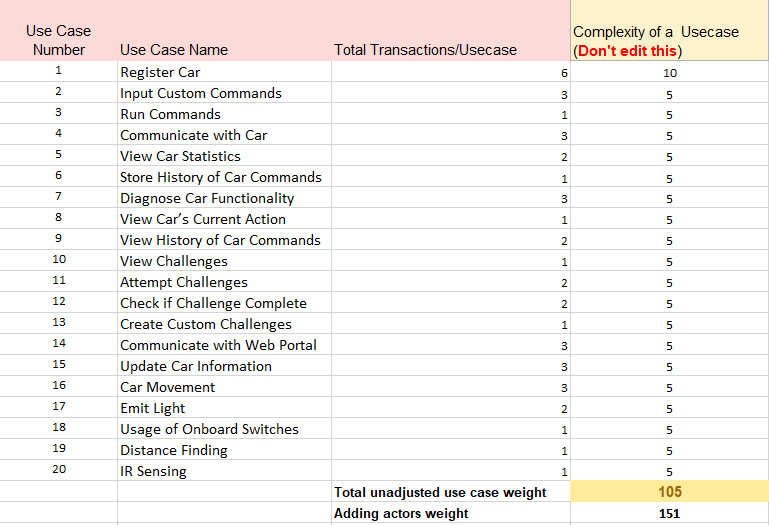
Some references here

[1] Monash University. “Citing and referencing: IEEE”. IEEE – Citing and referencing – Subject guides at Monash University. <https://guides.lib.monash.edu/citing-referencing/ieee> (Accessed Sept. 12, 2021)

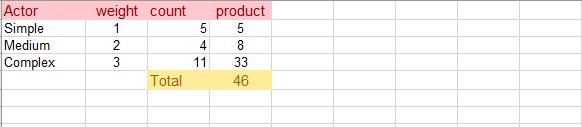
# Appendix A – Data Dictionary

<Data dictionary is used to track all the different variables, states and functional requirements that you described in your document. Make sure to include the complete list of all constants, state variables (and their possible states), inputs and outputs in a table. In the table, include the description of these items as well as all related operations and requirements.>

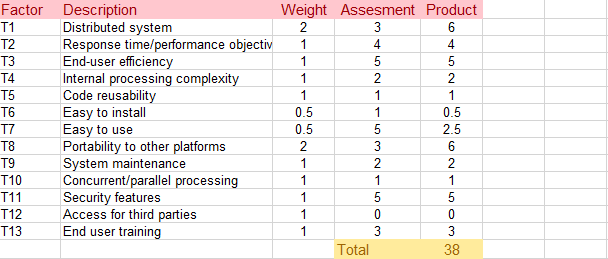
## A.1 Unadjusted Use Case Weight



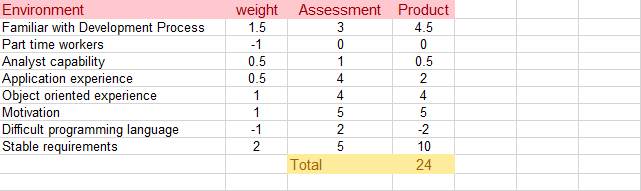
## A.2 Actors Weight



## A.3 Complexity Factors



## A.4 Environmental Factors



# Appendix B

Gantt - Use previous one

Burndown chart

Estimation Functional Points/Usecases (avg. estimated hours)