A drawing of a face

Description automatically generated

ICT2101/2201 Introduction to Software Engineering

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 6-16 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.>*

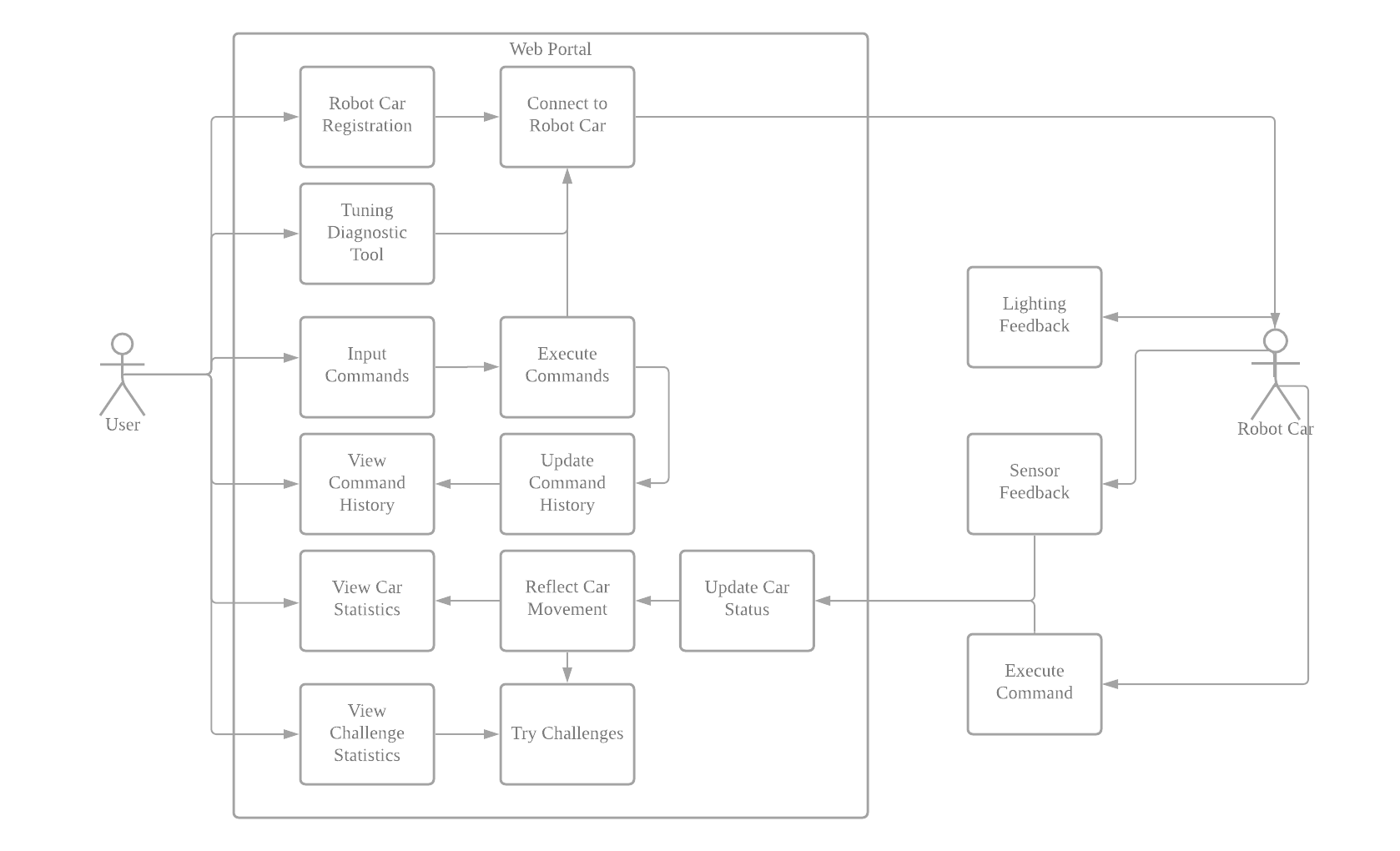


Figure 1 shows the high-level diagram of our product

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.).>*

### 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.>*

### 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).>*

## 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. >

**F1: The system shall …**

**F2: The system shall…**

…

## 3.3 Use Case Model

* *TO DO: Provide a use case diagram that will encapsulate the entire system and all actors.*

### 3.3.1 Use Case #1 (use case name and unique identifier – e.g. U1)

*TO DO: Provide a specification for each use case diagram*

**Author –** Identify team member who wrote this use case

**Purpose** - What is the basic objective of the use-case. What is it trying to achieve?

**Requirements Traceability –** Identify all requirements traced to this use case

**Priority** - What is the priority. Low, Medium, High. Importance of this use case being completed and functioning properly when system is depolyed

**Preconditions** - Any condition that must be satisfied before the use case begins

**Post conditions** - The conditions that will be satisfied after the use case successfully completes

**Actors** – Actors (human, system, devices, etc.) that trigger the use case to execute or provide input to the use case

**Extends –** If this is an extension use case, identify which use case(s) it extends

**Flow of Events**

* 1. Basic Flow - flow of events normally executed in the use-case
  2. Alternative Flow - a secondary flow of events due to infrequent conditions
  3. Exceptions - Exceptions that may happen during the execution of the use case

**Includes** (other use case IDs)

**Notes/Issues** - Any relevant notes or issues that need to be resolved

### 3.3.2 Use Case #2

…

## 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”>

## 4.2 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.

## 4.3 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

## 5.1 Test Cases

## 5.2 User Acceptance Testing (Only for Milestone 3)

# Work Distribution & Plan

# Conclusion

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

# Individual Reflections

# References

Some references here

# 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.>

# Appendix B

Gantt - Use previous one

Burndown chart

Estimation Functional Points/Usecases (avg. estimated hours)