RC simulation Based on Unreal through the Arduino

Mr. Areeshana Dissanayake

Submitted in partial fulfilment of the requirements
for the BSC Computer Science degree at the University of Westminster.

INFORMATICS INSTITUTE OF TECHNOLOGY
In Collaboration with
UNIVERSITY OF WESTMINSTER

July 17, 2023

Abstract

Miniature replica vehicles known as RC (Remote Control) automobiles may be operated remotely using a portable device like a remote control or a transmitter. They are now a cherished pastime for people of all ages and have experienced tremendous global growth. These vehicles are available in a variety of forms, dimensions, and styles to suit a wide range of tastes and hobbies.

The realistic appearance of RC automobiles is one of its main draws. Many RC automobiles, including race cars, sports cars, off-road trucks, and even recognizable vehicles from motion pictures or motorsports, are painstakingly made to match their real-life equivalents. Their detailed painting, precise decals, and aerodynamic body forms all contribute to the overall authenticity and visual attractiveness of their design. RC vehicles run on a variety of power sources. Rechargeable batteries power electric remote-control automobiles, making them a practical and green choice. They are appropriate for indoor use since they are frequently cleaner and quieter to use. On the other hand, some RC fans are like internal combustion engines, which run on a mixture of fuel and oil because of their thrill and realism. These gas-powered remote-control automobiles frequently create realistic exhaust fumes and a distinctive engine sound, simulating the sound of a real car. Users can maneuver RC cars precisely thanks to their control system. The automobile is given orders via the portable transmitter or remote-control device, which controls its motions. The controls may include steering, acceleration, braking, and even more specific features like suspension adjustments or lighting effects, depending on how complicated the model is. The fun and performance of an RC car are greatly influenced by the controls' responsiveness and the car's motions' precision. RC vehicles have many ways that they may be modified and customized. To improve the car's performance overall, enthusiasts can update a variety of parts, including the engines, batteries, suspension systems, tires, and electronics. In addition to allowing people to customize their remote-control cars, personalization fosters a thriving community of enthusiasts who exchange insider knowledge and creative modifications. The RC vehicle industry has embraced technological development and now offers cutting-edge features and capabilities. Some contemporary RC vehicles include Bluetooth connectivity, making it possible to operate them using special mobile apps on smartphones or tablets. These apps offer extra features like telemetry data, live video streaming from onboard cameras, or the ability to design and

alter the behavior of the vehicle. These technological developments have improved the RC car experience by introducing new levels of ease, engagement, and immersion.

In general, RC cars provide an interesting fusion of engineering, technology, and creativity. They offer countless hours of amusement by letting people enjoy driving and engaging in competition with toy cars. RC vehicles continue to enthrall hobbyists and enthusiasts all over the world, whether they are used for relaxing drives in the backyard, pleasant races with friends, or competitive competitions on professional circuits.

Declaration

I affirm that the project report and all related materials are a product of my individual effort. They have never been presented previously, nor are they currently under consideration for any academic degree.

Student's Full Name: Areeshana Umanga Jayathilake Dissanayake

Enrollment Number: w1790013

Signature: ....Umanga...........................
Date: 17th July 2023

I acknowledge that the above-named student has carried out his research project under my supervision.

Signature of the Supervisor: ....
Date: 17th July 2023

ACKNOWLEDGEMENT

When doing this research, I had to get help from many respectable people. Therefore, I would first like to express my deepest gratitude to my supervisor Mr. Iresh Bandara and our lecturer Mr. Guhanathan Poravi for providing me with valuable guidelines and relevant knowledge for completing this research. Your excellent supervision, attention, patience, and encouragement have provided me with a perfect platform to move this study forward. I would also like to thank everybody who helped me directly and non-directly for completing this research successfully.

Table of Contents

List of Tables

List of Figures

Chapter 1: INTRODUCTION

1.1. Chapter Overview

This proposal's initial goals are to give a succinct overview of the issue area involving Arduino and Unreal and to undertake a preliminary literature review. The issues with the state of the art are then used to provide explanations. Robotics has been used for solving several problems with the help of Arduino and Unreal. The author hopes to make the research gap and relevance concrete by doing this.

1.2. PROBLEM DOMAIN

Robotics has been explosive growth in recent years and will continue to grow. This growth leads to Robotics inevitable penetration in our day-to-day lives. To provide some it helps to do some work easily in difficult times. Provided this growth, the paramount quantities of data generated is already overwhelming conventional Arduino. However, Combining Robotics with Gaming will be a new challenge to the world that can provide solutions for many problems.

The problem domain of this project is heading Online competition instead of physical. Some competitions such as RC championship that can only held by physical. For solution for that using Unreal Engine for creating a simulation and connect to the unreal to communicate with the device.

1.2.1 Unreal Engine

Unreal Engine is the latest gaming creation engine these days. There are several games created by Unreal Engine 4, such as Mortal Kombat 11, Player Unknown's Battlegrounds, and Black myth - Wukong. Unreal Engine 4 gives the best graphic experience and can create objects easier. As much as it is helping to create games there are some features that can be used for architecture, 3D designs, and making movies. In this research it will helps to connect with Arduino to Develop the Device.

1.2.2 Arduino

Arduino is the best Hardware for develop the device and it will be running by Arduino IDE. Arduino IDE will be help for transferring the data between Unreal and the Device. And it helps to get the respond time correctly to playable the exact time.

And setting up a camera in a device might make it easy to get the view to the Unreal through the Arduino Board.

1.2.3 Serial COM

Serial COM Plug-in is for Connecting Unreal through the Arduino to get Accuracy easily. The Serial COM 3.0.0 Plugin now works natively in Unreal Engine 4.27/4.26/4.24 (PARTIAL SUPPORT WITH UE 5EA*) without having to re-build the project and possibly not be able to re-build for the lack of missing SDK's requirements. Unreal Engine 5 still requires rebuilding the plugin. UE5 can rebuild the plugin. Until a new method to rebuild the plugin for UE5 is available, this is the only version compatible with UE5.The new plugin will appear in your plugin list as "Communication Serial Port (Serial COM)". On your blueprints, is going to be listed in the "Communication Serial" category list instead of "UE4Duino". Search for "Arduino, Serial, Communication" will also show the results for these functions.

1.3. PROBLEM DEFINITION

As mentioned, Section 2, new research combines Unreal Engine and Arduino, especially for updating gaming models. Collaborative with Unreal might be easy for control the device within a range problem and helps to future Autonomous vehicle system. This research is beyond help for robot fighting, Racing, and drone controlling too. The hardest challenge will be developing the map which has to use Real time mapping to get exact route and put it into Unreal. RC vehicles were controlled in real-time with a model-predictive contouring approach, which minimized the vehicle path deviations from the circuit middle line and maximized the travelled distance. Getting Accuracy in real time will be difficult to get in the process.

1.3.1 Problem Statement

Due to the Pandemic situations some competition can be held in physical for that finding a solution that can be useful to do this type of competition in online.

1.4. MOTIVATION

After creating many games and getting experience about several Game engine done research that there are few Games related Research that can be found. For that this problem can be done by gaming path and Arduino. Instead of remote controller can be use an application for play the Drivable simulation that can also drive the device too. Thus, it can be playable in any machine after Building whole simulation in Game Engine. For that Unreal Engine is the best engine that can be used for building the simulation.

1.6. RESEARCH GAP

The RC car championship can be referred to as one of the most popular championships in the world. With the pandemic situation, all the events related to this championship have been limited as many difficulties occur with the organization. To address this problem, the solution I brought was implementing a car game in the unreal engine as it connects with the RC car through Arduino and plays the RC car game online.

1.7. CONTRIBUTION TO THE BODY OF KNOWLEDGE

By enabling simulation sharing, more connected applications and research are possible. possibilities. Therefore, the following are the contributions of this research:

1.7.1 Contribution to the problem domain

A massive transformation brought about by the industrial revolution will be assisted by RC automobile simulation powered by Arduino. It may be used for parking systems and autonomous self-driving vehicles. The results of this study will afterwards contribute to the development of IOT through Game AI applications and research.

1.7.2 Contribution to the research domain

Contribution of this research domain is improved online session that can be done easily through the difficult times.

1.8. RESEARCH CHALLENGE

1. **Lack of Resources –** It can be challenging to locate specific equipment for the Arduino section of the project, such as an Arduino Board and an HD camera for mounting in a RC car. Resources also differ based on the manufacturer and use case due to the heterogeneity of devices. Thus, such a scheme should be incredibly lightweight with little overhead.
2. **Connecting Issues –** In this research, connecting Arduino and Unreal presented several difficulties.

3. **Environmental factors -** Many embedded devices are deploy-and-forget types of equipment. Considering that, there is no end to the places where such devices can be installed. Therefore, the use of standardized communication systems can be influenced by a variety of contextual circumstances.

1.9. RESEACH QUESTION

RQ 1 - What is the respond time between Unreal and Arduino?

RQ 2 - What are the best methods for data transfer to the vehicle?

RQ 3 - How to get Camera view to Unreal?

RQ 4 - How to get sensors to Unreal through Arduino?

1.10. RESEARCH AIM

This research supports autonomous self-driving systems, which have the potential to revolutionize the world, as well as winning online competitions.

1.11. RESEARCH OBJECTIVES

Research Objectives	Description	Learning Outcomes
Literature Review	To achieve the following objectives, a review of the body of literature is applied. • To specify evaluation standards and metrics. • To Identify the research gaps on the project. • To systematically analyze the exciting system in the Arduino and Unreal.	L01, L06
Requirement Elicitation	There was a thorough review of the prerequisites to,	L02, L03, L04, L05, L06

	• To use surveys to specify the requirements for hardware and software. • To compile the demands of researchers and domain professionals to comprehend what they anticipate the system to achieve.	
Design	The proposed framework for cognitive networking's design intends to, • To create a driving environment for an unreal world.	L03, L04, L06
Implementation	creation of the conceptual model in accordance with the architectures and models created during the design stage using the determined hardware and software components. • To develop the task switching mechanism.	L03, L04, L06
Evaluation	Putting the prototype to the test and having researchers and specialists in the field examine it, • To evaluate how the system performs in the real world and simulations. • To test individual components of the prototype	L04, L06

Table 1: Research Objectives

1.14 Chapter SUMMARY

The project proposal for the research that will be done in the next months was provided in this paper. First, problem definition and problem domain elaboration were carried out. The problems of the planned study were then thoroughly explored, as well as the current systems.

Chapter 2: Literature review

This chapter is more user-specific so there is no specific structure to be followed.

2.1 Chapter overview

Past research and relevant sources are vital to finding a solution to the problem we refer to. The research related to Arduino, Unreal engine, and other discussions of the specialists for increasing user-friendliness has helped the author to give a better solution with high value. The understanding and knowledge gained from the relevant sources are deeply described in this chapter.

2.2 Concept map

Please refer APPENDIX A – Concept map

2.3 PROBLEM DOMAIN

Robotics has been explosive growth in recent years and will continue to grow. This growth leads to Robotics inevitable penetration in our day-to-day lives. To provide some it helps to do some work easily in difficult times. Provided this growth, the paramount quantities of data generated is already overwhelming conventional Arduino. However, Combining Robotics with Gaming will be a new challenge to the world that can provide solutions for many problems.

The problem domain of this project is heading Online competition instead of physical. Some competitions such as RC championship that can only held by physical. For solution for that using Unreal Engine for creating a simulation and connect to the unreal to communicate with the device.

2.3.1 Unreal Engine

Unreal Engine is the latest gaming creation engine these days. There are several games created by Unreal Engine 4, such as Mortal Kombat 11, Player Unknown’s Battlegrounds, and Black myth - Wukong. Unreal Engine 4 gives the best graphic experience and can create objects easier. As much as it is helping to create games there are some features that can be used for architecture, 3D designs, and making movies. In this research it will helps to connect with Arduino to Develop the Device.

2.3.2 Arduino

Arduino is the best Hardware for develop the device and it will be running by Arduino IDE. Arduino IDE will be help for transferring the data between Unreal and the Device. And it helps to get the respond time correctly to playable the exact time. And setting up a camera in a device might make it easy to get the view to the Unreal through the Arduino Board.

2.3.3 Serial COM

Serial COM Plug-in is for Connecting Unreal through the Arduino to get Accuracy easily. The Serial COM 3.0.0 Plugin now works natively in Unreal Engine 4.27/4.26/4.24 (PARTIAL SUPPORT WITH UE 5EA*) without having to re-build the project and possibly not be able to re-build for the lack of missing SDK's requirements. Unreal Engine 5 still requires rebuilding the plugin. UE5 can rebuild the plugin. Until a new method to rebuild the plugin for UE5 is available, this is the only version compatible with UE5.The new plugin will appear in your plugin list as "Communication Serial Port (Serial COM)". On your blueprints, is going to be listed in the "Communication Serial" category list instead of "UE4Duino". Search for "Arduino, Serial, Communication" will also show the results for these functions.

2.4 EXCITING WORK

Citation	Brief => about their research in nutshell	Limitation of their research	Contributions of their research	Critical review
Fawwad, A., Shabbir, Shyan, L. and Thangasamy, V. (2017). Real-Time Indoor Tracking. *Journal of Applied Technology and Innovation*, [online] 1(2), pp.49–57. Available at: https://dif7uuh3zqcps.cloudfront.net/wp-	In this paper, the development of an autonomously navigating robot car, where the robot car would autonomously navigate towards a user-defined destination location, using a UWB based indoor localization	The advances in localization-based technologies and the increasing importance of ubiquitous computing and context-dependent	Location awareness is becoming highly important in the daily lives of humans (Rainer, 2012). From the exact location of one's	In this research will be helping to get a knowledge of localization of real time tracking with Arduino. Also, can get a critical idea about Accuracy

content/uploads/sites/11/2018/07/17035741/2017_Issue2_Paper5.pdf [Accessed 18 Oct. 2022].	system is described. For accurate 3-axis indoor localization, Pozyx indoor localization system was used, which uses Decawave DWM1000 chips for UWB communication. Time of Arrival distance calculation method, and Least Linear Square Algorithm were used for calculation of location co-ordinates of the Pozyx mobile tag, which was integrated with the robot car. A GUI was developed on Unreal Engine, to provide real-time visual representation of location of the robot car. The user was allowed to set destination co-ordinates using pick and place in the developed Unreal	information have led to a growing business interest in location-based applications and services. Today, most application requirements are locating or real-time tracking of physical belongings inside buildings accurately; thus, the demand for indoor localization services has become a key prerequisite in some markets.	children, to the exact coordinates of an Unmanned Aerial Vehicle, these have become integral for smooth and efficient performance of highly important daily tasks. GPS has revolutionized outdoor localization, and has ensured the possibility of self-navigating outdoor vehicles. Indoor localization is much more complex due to the highly variable	and the respond time.

	Engine GUI. The autonomous navigation robot was programmed, using an Arduino Uno microcontroller.		indoor environment layouts (Faird, Nordin & Ismail, 2013). Overcoming this complexity can lead to increased accuracy and efficiency in indoor localization.	
Ashton, J., Hunt, S. and Spencer, M. (n.d.). *WORCESTER POLYTECHNIC INSTITUTE MAJOR QUALIFYING PROJECT Autonomous RC Car Platform*. [online] Available at: https://web.wpi.edu/Pubs/E-project/Available/E-project-032219-094723/unrestricted/MQP_Final_Paper.pdf [Accessed 2 Nov. 2022].	This project explores building an autonomous research robot on a 1/10 scale RC car platform. The goals of the project were to build an easy-to-use system that allowed for the exploration of techniques such as localization, object detection, mapping, and more.	Sitting outside in Mountain View, California by a public road, there's a good chance that you may see multiple cars drive by with a wide array of sensors attached to the vehicle. Companies like Waymo, Lyft, Cruise, and Tesla, and	A small-scale RC Car platform was successfully developed that allows for future students to continue work on the car as well was documented with the possibility to create multiple	This research will be got a Idea about camera view system and sensors in Unreal through the Arduino.

		universities including Stanford and Carnegie Mellon are all researching self-driving vehicles for use on public roads.	platforms if desired.	

Table 2: Exciting work

2.5 Evaluation and benchmarking

Evaluating and benchmarking the methods and techniques discussed in the literature review is an essential aspect of understanding their strengths, weaknesses, and potential for improvement. This section provides an overview of the evaluation and benchmarking approaches employed in the literature related to Arduino based projects and unreal related projects.

2.5.1 Performance metrics for Arduino Project

- Execution time: The amount of time it takes for the project to run. This can be measured in milliseconds or seconds.
- CPU utilization: The percentage of the CPU that is being used by the project. This can be measured in percentage points.
- Memory usage: The amount of memory that is being used by the project. This can be measured in kilobytes or megabytes.
- Throughput: The number of tasks that the project can complete per unit of time. This can be measured in tasks per second or tasks per minute.
- Latency: The amount of time it takes for the project to respond to a request. This can be measured in milliseconds or seconds.
- Bandwidth: The amount of data that the project can transfer per unit of time. This can be measured in kilobytes per second or megabytes per second.

These are just some examples of performance metrics that can be used to evaluate Arduino projects. The specific metrics that are used will depend on the specific project and its intended purpose.

For example, if you are evaluating an Arduino project that is designed to control a robot, you might want to focus on the execution time, CPU utilization, and memory usage of the project. However, if you are evaluating an Arduino project that is designed to collect data from sensors, you might want to focus on the throughput and latency of the project.

The most important thing is to choose the metrics that are most relevant to the specific project that you are evaluating. By doing so, you can ensure that you are getting a fair and objective assessment of the project's performance.

Additional pointers for selecting performance indicators for Arduino projects are provided below:

- Consider the project's intended purpose. What are the key performance requirements for the project?
- Identify the key bottlenecks. What are the parts of the project that are taking the most time or using the most resources?
- Measure the metrics that matter. Don't just measure everything. Focus on the metrics that are most important for the project.
- Collect data over time. Performance can vary depending on the workload. Collect data over time to get a better understanding of the project's performance.

By following these tips, you can choose the right performance metrics for your Arduino projects and get a fair and objective assessment of their performance.

2.6 Chapter summary

In this chapter, the existing literature, and technologies relevant to the development of Arduino unreal related projects have been reviewed. The insights gained from this review will be valuable in guiding the proposed system's design,
implementation, and evaluation.

Chapter 3: Methodology

3.1 Chapter overview

This chapter's explanation of research methodology covers tactics and techniques used for data collection, time-bound task identification, task management, and other resources needed to finish the study.

Three sections make up the chapter. Give the reader an overview of the methodologies and research-conducting techniques employed in the study, starting with the section on research methodology. The project management methodology, which describes development approaches, the development methodology, which details how the author has managed the project tasks in detail, and the resources section, which lists the physical and intangible resources used to carry out the research, are all included in the second section.

3.2 Research methodology

Research Philosophy	**Pragmatism** was chosen as the preferred research philosophy above positivism, interpretivism, realism, and positivism. This choice has been made. by looking at the assessment measures (such as accuracy, latency, etc.), current surveys and research comparisons that combine quantitative and qualitative data. qualitative data as well. Additionally, this study's findings demonstrate. positivist the theory with facts and empirical proof, making it the philosophy that is most suited to the intended research.
Research Approach	Out of the available research approaches deductive and inductive, the **inductive** approach was chosen due to the final analysis being a quantitative analysis. That way can use sample data to get accuracy for the localization.

Research Strategy:	The onion model's third layer focuses on data gathering. **Surveys** will serve as the main research method for this aim since they offer the opportunity to gather large amounts of quantitative data. Additionally, throughout the project's later stages, such as assessment, testing, and feedback, interviews will be employed as a secondary method of data collecting.
Research Choice:	The **mixed method** was picked in this case out of the three alternatives of mono, mixed, and multi approaches. To further illustrate, both quantitative (surveys) and qualitative (interviews) methodologies were chosen throughout the research to support each other.
Time horizon	The duration of the investigation is determined by the time horizon. The cross-sectional time horizon was chosen out of the two-time horizons. Data for this project will be acquired throughout the Requirement Analysis phase of the research. Cross-sectional temporal horizon is therefore a superior possibility.

Table 2: Research Methodology

3.3 Development methodology

3.3.1 Software Development Model

The prototype model was chosen for the research study out of the development models. This technique features an iterative design, trial-and-error, and assessment procedure to support it, which allows for adjustments, changes, or new needs to be addressed. Additionally, because the assessment phase comes after the design and development phases, using this methodology allows for the seamless inclusion of the changes called for by researchers and industry professionals.

3.3.2 Design Methodology

For the Design Methodology, Unreal Engine was chosen. This is primarily due to the model's re-usability and intractability among objects.

3.3.3 Requirement Elicitation

This section describes how the requirements for the proposed project will be acquired. For this
purpose the following elicitation methods will be used.

1. **Literature Review** - Understanding the underlying technology will come by analyzing similar systems and technologies.
2. **Observations** - The unique study topic of connecting Unreal to Arduino calls for experimentation prior to final implementation. The findings from the experimental phase will be applied to the finished product.
3. **Surveys** - Surveys will be handed out to selected individuals who are sufficiently proficient in the domains of Unreal and Arduino. The data is mostly quantitative, meaning that it can be represented in various graphical forms.

    Interview - To acquire qualitative information on the project, interviews with industry professionals will be undertaken. However, because interview data is qualitative, it will need thematic analysis to glean crucial information from the interview subjects.

3.4 Project Management Methodology

Out of the many project management methodologies, Unreal Documentation Serial COM was chosen. It specifies the connection between Arduino and came with the plug in too.

3.4.1 Gantt chart

Please refer APPENDIX B - Gantt chart

3.4.2 Deliverables

Deliverable	Date
Submission of Project Proposal	3rd Nov 2022
Initial Project Specifications Design and Prototype (PSDP) Submission	23rd Jan 2023

Project Specifications Design and Prototype (PSDP)	2nd Feb 2023
Test and Evaluation Report	23rd Mar 2023
Submission of Draft Project Reports	30th Mar 2023
Final Project Report	17th July 2023

Table 3: Deliverable

3.5 Resource Requirement

Based on the afore defined objectives, research methodologies and high-level architecture, the

resources necessary to complete the project are identified. The required software, hardware, and data resources with justifications as to why they are required are mentioned below.

3.5.1 Skill Requirement

- **Knowledge of Unreal –** In order to create the simulation Unreal knowledge is needed.
- **Knowledge of Arduino –** To connect and share the data to unreal and device Arduino knowledge is needed.

3.5.2 Software Requirement

- **Operating System -** The resource rich nature of the OS allows us to run essential programs such as the web browsers and IDEs required for the project.
- **C# -** C# will be used for primary programming language for build the connection between Unreal and Arduino
- **Blueprints –** Blueprint is the main language of the unreal. For that Blueprints must be used to create the movement of the vehicle in simulation.
- **Arduino IDE –** Arduino IDE used for building the connection between Unreal and the device.
- **Zotero -** A research tool used to manage and backup research papers and relevant artifacts to the local drive or cloud.

- **GitHub** – GitHub will be used for getting the plug-in and Forum that is used for Serial COM.

3.5.3 Hardware Requirement

- **Intel** 5th **Gen I series processor or above -** To perform long running, intensive simulations to play the Game Engine.
- **Arduino Board –** To develop the Car.
- **Serial COM -** To transfer the data between Device and Unreal through Arduino IDE.
- **NVIDIA GTX 1650 or more –** To get better Graphics in Engine.
- **8GB Ram or more -** To be able to manage relatively large volumes of data.
- **Minimum of 20GB –** To run the unreal engine and Arduino IDE exact time to maintain the respond time.

3.6 Risk Management

By examining the risks involved in undertaking this endeavor, one may better prepare for such situations, should they occur.

- Lack of knowledge in Arduino – To build the device must follow stack overflow, GitHub, online courses, and tutorials.
- Lack of resources for the building process – To develop the project you have to gadget that helps to create the device and communicate with the computer.

3.7 Chapter summary

This chapter also described the research project management methodology used and highlighted how he had managed the project by testing and delivering on time.

Chapter 04: SOFTWARE REQUIREMENTS SPECIFICATION

4.1 Chapter Overview

By analyzing all conceivable points of interaction with the system, this chapter seeks to identify all the stakeholders engaged in the project. This will be accomplished by the creation of a rich picture diagram and the collation of stakeholder perspectives. Following the information gathering process, prospective use cases and functional and non-functional prototype requirements will be developed.

4.2 Rich Picture Diagram

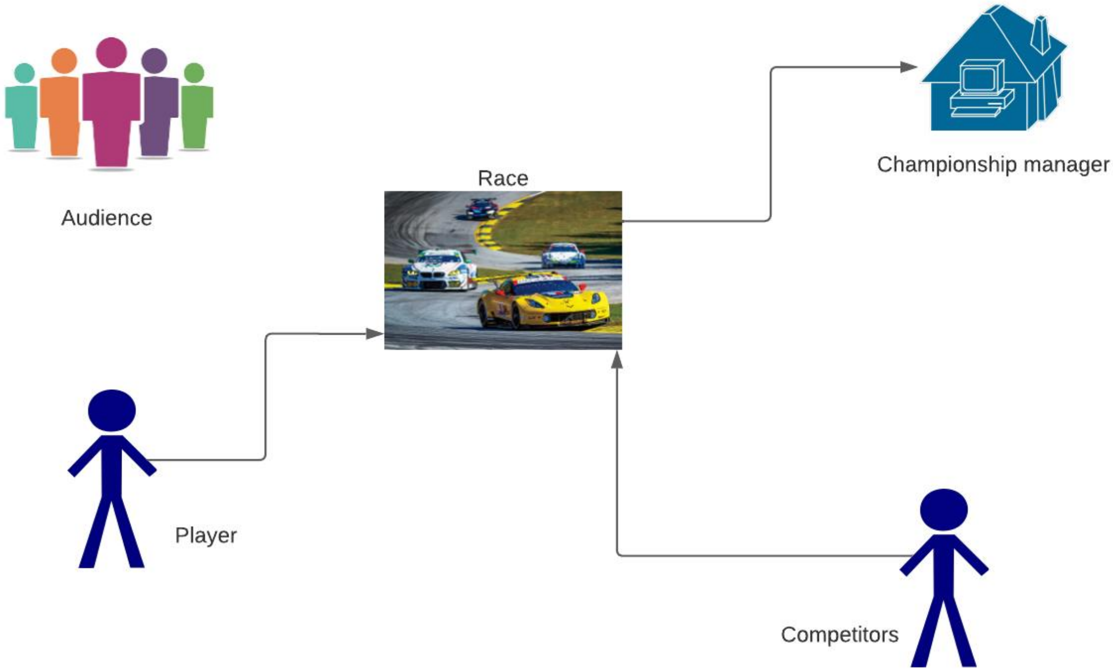


Figure 1: Rich picture Diagram

4.3 Stake Holder Analysis

The Stakeholder Onion Model visualizes the stakeholders connected to the system and details their level of involvement with the system in the form of Stakeholder Viewpoints. This model provides an overview of the stakeholders and their respective roles within the system.

4.3.1 Stake Holder Onion Model

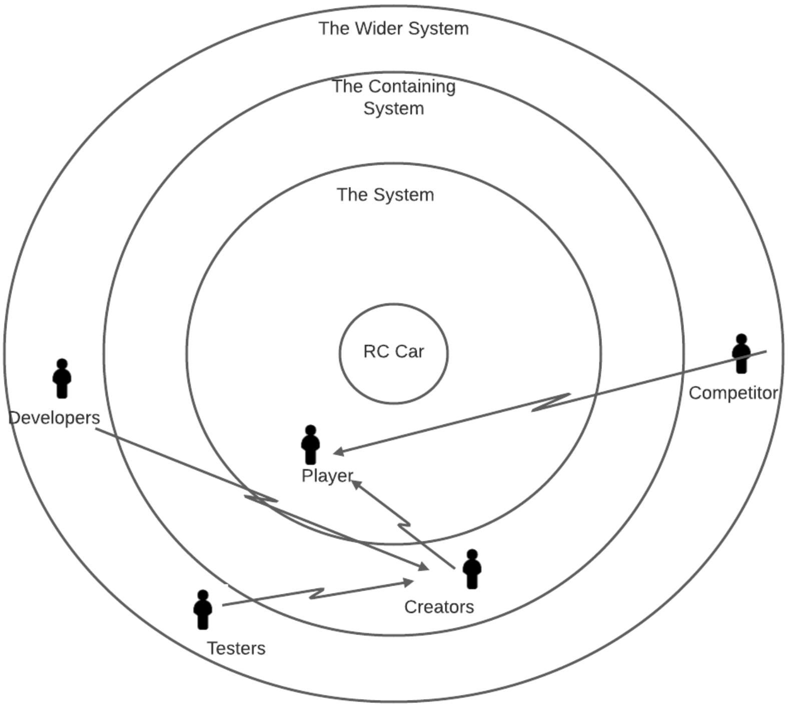


Figure 2: Stakeholder Onion Model

4.3.2 Stake Holder Viewpoints

Stakeholder	Roles	Benefit/ Role description
Developer	Quality regulator, Experts	Develop the system.
Tester		Testing the application.
Competitor	Negative stakeholder	May build a similar product that outperform pricing.
Creators	Product deployment & maintains	Help running the system correctly and stable the situation.
Player	Fundamental beneficiary	The person who invests the project and run it.

Table 4: Stake holder viewpoints

4.4 Selection of Requirement Elicitation

The study effort used a range of requirement elicitation methodologies to collect data. A literature study, interviews, questionnaires, and prototyping were among the approaches used. The justification for picking these techniques has been described and clarified below.

Method 1: Literature Review
At the beginning of the project, the author conducted an extensive literature review to uncover gaps in the chosen field of study and domain of interest. The purpose of this review was to understand any lacking areas in current technologies that could be addressed. This involved studying existing systems and related technologies that could be applied to these systems, as described in the relevant literature.
Method 2: Survey
A survey was conducted using a questionnaire as a tool to gather requirements and insights from prospective users of the proposed system. This survey aimed to provide the author with an understanding of the thought processes and expectations of potential users for the prototype. It also served as a way for the author to confirm if the proposed solution would be useful and beneficial to its intended users.
Method 4: Prototyping
Since the project followed the Agile Software Development Life-cycle approach, prototyping offered the opportunity for the author to repeatedly test and evaluate alternative implementations, allowing for the identification of areas for improvement. This iterative process would help to refine the prototype and ensure that it meets the needs of the stakeholders.

Table 5: Requirement Elicitation Methodologies

4.6 Discussion of Findings

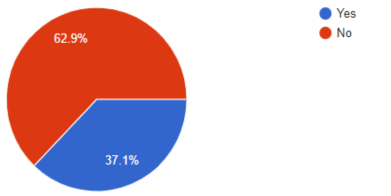
The results of the data collected through the various requirement elicitation techniques are presented below.

4.6.1 Literature Review

Finding	Citation
Location awareness is becoming highly important in daily lives of humans (Rainer, 2012). From exact location of one's children, to the exact coordinates of an Unmanned Aerial Vehicle, these have become integral for smooth and efficient performance of highly important daily tasks. GPS has revolutionized outdoor localization and has ensured the possibility of self-navigating outdoor vehicles.	Fawwad, A., Shabbir, Shyan, L. and Thangasamy, V. (2017). Real-Time Indoor Tracking. *Journal of Applied Technology and Innovation*, [online] 1(2), pp.49–57. Available at: https://dif7uuh3zqcps.cloudfront.net/wp-content/uploads/sites/11/2018/07/17035741/2017_Issue2_Paper5.pdf [Accessed 18 Oct. 2022].
A small-scale RC Car platform was successfully developed that allows for future students to continue work on the car as well was documented with the possibility to create multiple platforms if desired.	Ashton, J., Hunt, S. and Spencer, M. (n.d.). *WORCESTER POLYTECHNIC INSTITUTE MAJOR QUALIFYING PROJECT Autonomous RC Car Platform*. [online] Available at: https://web.wpi.edu/Pubs/E-project/Available/E-project-032219-094723/unrestricted/MQP_Final_Paper.pdf [Accessed 2 Nov. 2022].

Table 6: Literature survey

4.6.2 Survey

Question	Do you have RC cars?
Aim of the question	To see how many people play RC cars for as a hobby or profession
Finding and Conclusion 37.1% of people have RC cars and 62.9% majority of people don't have them.	

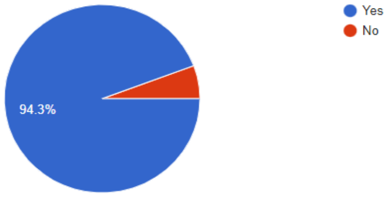
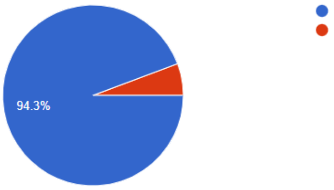
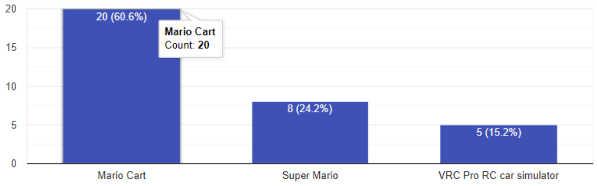
Question	Have you ever played Computer car games?
Aim of the question	To find how many people play car games in computer
Finding and Conclusion 94.3% people play car games in computer and 5.7% people are doesn't.so that people will find an interesting in play Rc cars online.	
Question	Have you ever heard a platform that can play RC cars in online? If yes, please mention them below
Aim of the question	To see how many games have that can play RC cars online
Finding and Conclusion	 
The result of the questions shows 94.3% people know RC car games such as Mario cart, Super mario (same as mario cart) and VRC Pro RC car simulator. 60.2% people are suggest mario cart but VRC PRO RC car simulater can be helpful for the project.	

Table 7: Survey

4.6.3 Prototype

There developed many requirements and challenges through iterative prototyping. First, there were no projects like this, and it was difficult to find the requirements. The main challenge of this project is to connect the connection between Unreal and Arduino. Incorporating social trends data provided a fresh and valuable perspective that could be utilized for making recommendations. Please refer APPENDIX B – Prototype

4.6.4 Summary of Findings

Findings	Literature Review	Survey	Prototype
The proposed system would be useful for professional and Hobby for users.	✓		
The new platform can be used for creating the simulation.		✓	
Can used simple device for the demo part.			✓
Get Location awareness for setting up the track.	✓		
Find about the people who are interesting in computer games.		✓	✓
For future continue work on autonomous cars.	✓	✓	✓

Table 8: Summary of Findings

4.7 Context Diagram

The system's boundaries and interactions should be established before development.

The figure below shows the context of the system.

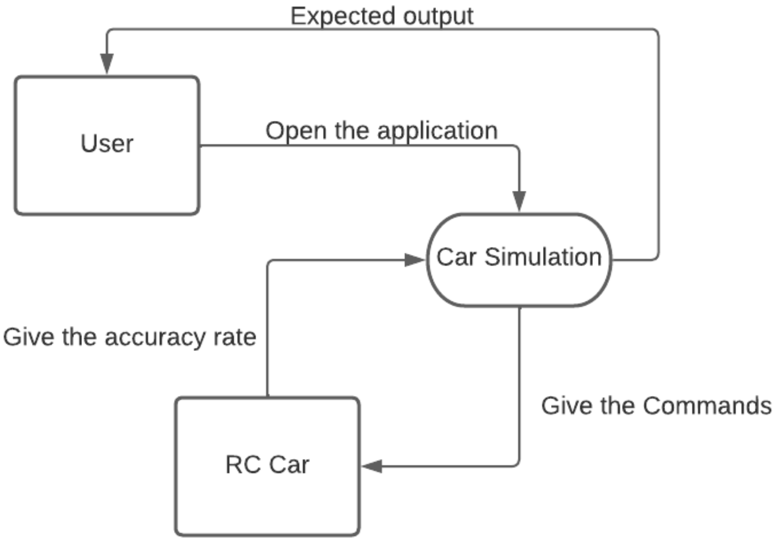


Figure 3: Context Diagram

4.8 Use case diagram.

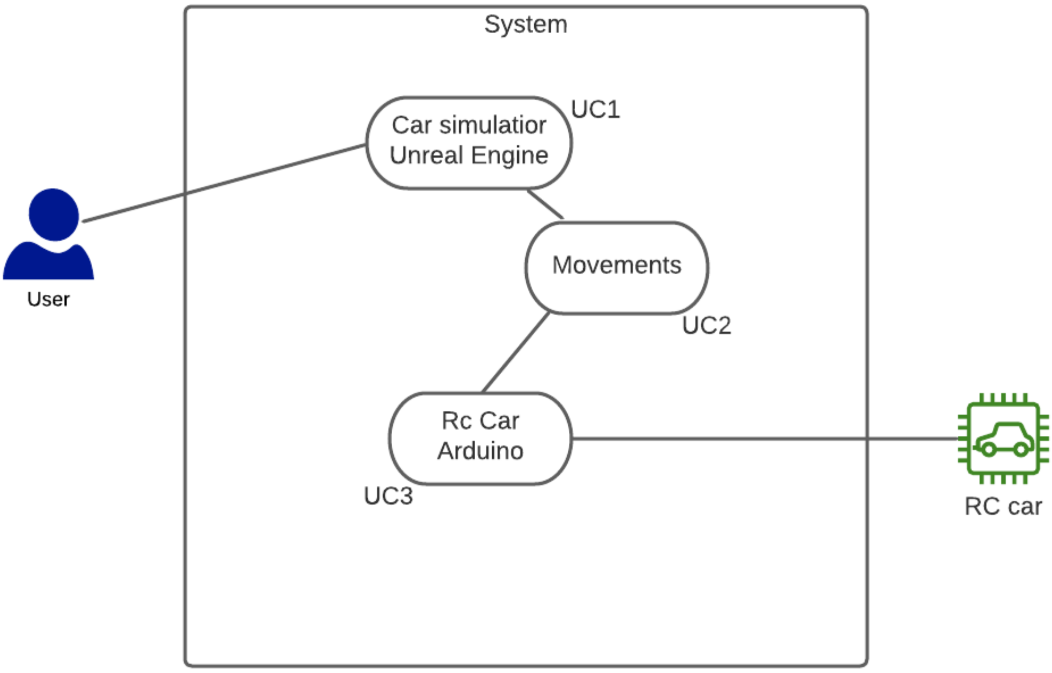


Figure 4: Use case Diagram

4.9 Use case descriptions.

User case	Car simulation (Unreal engine)
ID	UC 1
Description	Open the application to play the simulation.
Primary actor	User
Stakeholders (If any)	none
Triggers	A user wishes to play the RC car through the application.
Main success scenario	• System gives the output to the device. • Can get accuracy rate from device.
Variation	A user can be transferring the input to the device through the application to get the output.

Table 9: Use case Description

User case	RC car (Arduino)
ID	UC 3
Description	Give the commands to the RC car.
Primary actor	none
Supporting actor (If any)	RC car
Stakeholders (If any)	none
Triggers	The Arduino gives the command to the device to make movements.
Main success scenario	• The device gives the output to the application. • Can give the accuracy rate to the device.
Variation	The RC car can be controlled by give the output through the Application by user.

Table 10: Use case Description

4.10 Requirements

4.10.1 Functional Requirements

Based on their significance, the MoSCoW technique was used to rank the system demands in order of priority.

Priority Level	Description
Must have (M)	The requirement at this level is the fundamental functional requirement for a prototype, and it must be carried out.
Should have (S)	Although not strictly required for the anticipated prototype to function, important criteria do provide a lot of value.
Could have (C)	Optional, non-essential desirable needs are crucial to the project's scope.
Will not have (W)	The objectives that the system might not provide right now and that aren't given high priority.

Table 11: MoSCoW

Requirement	Priority level	Use case
User should be giving the exact output to the application to run the device.	S	UC1
Application must have followed the right movement to run the simulation.	M	UC3
RC car Could have given the accuracy rate as expected to the application	C	UC3
RC car Must give the correct input to the application to run the device.	M	UC3
Application will not have speed rate to get the accuracy by the device.	W	UC2
Movement will not have given the opposite command to the device.	W	UC2
User should be careful with the device while it is controlling by the application.	S	UC1

Table 12: Functional Requirement

4.10.2 Non-Functional Requirements

Requirement	Description	Priority Level
Performance	While recommendations are based on user input, the recommendations matrix and opinion-mining data can be pre-processed and stored in memory for use. Real-time processing is not a requirement, as the data can be processed in advance.	Desirable
Security	To ensure the security of user inputs, the application should prevent any attempts by attackers to manipulate results or extract user information. This can be achieved through testing and ensuring that proper security measures are in place.	Desirable
Quality of data	Given that this is a new project, it is imperative that the output be of the highest quality possible.	Important
Scalability	Due to the high level of interest in the project, the prototype may be opened for testing by many users, potentially leading to many concurrent user requests. The system must be designed to handle such a high volume of traffic	Desirable

Usability	Since the purpose of this project is to play RC cars in online the usability of this system must be easy for users of all the level of expertise.	Important

Table 13: Non-Functional Requirement

4.11 Chapter Summary

In this chapter, a Rich Picture Diagram was created to demonstrate the connection between the system and society, helping to identify the stakeholders involved. The Stakeholder Onion Model was used to show the stakeholders and their level of influence. To gather all necessary information and opinions from potential stakeholders, requirement elicitation techniques were employed. Finally, the use cases, functional requirements, and non-functional requirements for the system were determined based on the insights gathered through these elicitation methods.

Chapter 5: Social, Legal, Ethical and Professional issues (SLEP)

The BCS code of conduct must be reviewed and detailed if any of the mentioned SLEP issues have been faced and how they were mitigated. Must be very specific to your project. Putting anyone’s name (ex: the ones who you interviewed) requires written consent.

5.1 Chapter overview

In this chapter, the author has deeply described how he has followed the guideline of BCS (British Computer Society) for adhering to the ethics of computing when conducting the research. When conducting research on this topic, it is unavoidable to use sensitive data, systems, and technology. The author avoided any potential problems by considering the potential for them to be raised and the potential for them to breach the social, legal, ethical, and professional identities of other people. The researcher has followed the methods listed below for each stage to secure sensitive data by limiting the access of unauthorized people.

5.2SLEP issues and mitigation

5.2.1 Social issues

RC simulation with the Arduino and Unreal may bring up a variety of societal themes, such as:

- Safety: RC simulators may be used to teach people how to drive actual RC cars, but there is a chance that they might also make individuals overconfident in their skills and encourage them to take risks they wouldn't otherwise take. Injury and accident risks might result from this.
- Privacy: Data regarding driving behaviors may be gathered through RC simulations and used to monitor people's whereabouts or target them with advertisements. Concerns concerning data security and privacy are raised by this.
- Cyberbullying: Realistic simulations of violence may be made using RC simulations and then used to harass or bully others. The victim's mental health can suffer as a result.
- Stereotypes about gender: In RC simulations, men drivers are frequently shown to be more skillful and aggressive than female drivers. This may make girls and women less likely to participate in RC activities and promote gender stereotypes.

5.2.2 Legal issues

When using the Arduino to build an RC simulation based on Unreal, there are a few legal considerations to be mindful of.

- Copyright violation: You may be held accountable for copyright infringement if your simulation contains any pictures, audio, or code that are protected by a copyright. Before using another person's work, you should get their permission.
- Trademark infringement: You may be held accountable for trademark infringement if you utilize trademarked phrases in your simulation, such as the names of real firms or items. Unless you have the trademark holder's consent, you should refrain from utilizing trademarked terminology.
- Privacy rules: You must abide by privacy laws if your simulation gathers user personal data such names or email addresses. A privacy policy should be in place outlining how you will gather, utilize, and keep user data.

- Export restrictions: You must abide by export control rules if your simulation uses technology that is restricted from export. If you want to know whether your simulation is subject to export restrictions, you should check with the appropriate government body.

5.2.3 Ethical issues

Multiple ethical issues could develop in an Arduino-based RC simulation that is based on Unreal. These consist of:

- Privacy: Information on people's movements and activities may be gathered through RC simulation. This information may be used to identify individuals, follow their whereabouts, or even forecast their behavior. Concerns regarding privacy and the possibility of surveillance are raised by this.
- Safety: Realistic simulations of risky or unlawful actions may be made using RC simulation. If anyone tries to replicate these behaviors in real life, it might result in pain or injury. For instance, a person may utilize RC simulation to learn how to fly a drone before attempting to do it in real life without the necessary preparation.
- Misinformation: With the help of RC simulation, bogus movies and pictures that mimic actual occurrences may be produced. This may be used to disseminate propaganda or false information. As an illustration, someone may produce a phony protest film that appears to be in progress. This may be used to cause discontent or strife.
- Intellectual property: RC simulation can be used to replicate items that are protected by copyright, such as vehicles, structures, or people. The proprietors of such works' intellectual property may be infringed upon by this.

5.2.4 Professional issues

There are some technical challenges with Arduino-based RC simulation using Unreal:

- Unreal Engine, one of the programs used for RC simulation, is subject to a range of licenses, including commercial, academic, and personal usage. Make sure you have the appropriate license for the program you're using.

- Copyright or other intellectual property rights may be utilized to safeguard the models and other assets used in RC simulation. To use these models and materials, you must be sure that you have the necessary authorizations.
- Professionalism: It's critical to uphold a high standard of quality when constructing RC simulations. This entails employing high-quality models and resources and producing error-free simulations.
- Credibility: RC simulations may be used to show prospective clients or investors how items or concepts work. You must make sure that your simulations are realistic and believable.
- Safety: People may learn how to drive RC cars by using simulators. Make sure your simulations are secure and do not put users in danger.

5.3 Chapter summary

In this paragraph, the SLEP (Social, Legal, Ethical, and Professional) difficulties associated with the creation and application of an automobile simulation are highlighted. The social issues are described the importance of safety and the privacy of the model. Legal issues are described the privacy violation, trademark infringement and privacy rules too. Ethical issues are described in the misinformation and the Intellectual property of the design prototype of the project. Licensing, safety, copyright, and credibility are described in professional issues in the project.

Chapter 06: DESIGN

6.1 Chapter overview

This chapter outlines the design choices made to develop an appropriate architecture for the implementation, considering the requirements that were collected. The design process includes high-level and low-level design, design diagrams, and UI wireframes to show how the design objectives will be accomplished. The chapter also explains the rationale behind the design decisions that were made.

6.2 Design Goals

- Performance – Since this is a new project the performance of the application must be clear. The connection of the Application and the RC car had to be made correctly to run the project otherwise it will give the wrong command to the device.
- Usability – The purpose of this project is to play online in RC cars, the usability of this project must be easy for users of all the levels of expertise.
- Adaptability – The recommendation models utilized in the system should be flexible enough to allow for changes and upgrades in the future. The ability to swap out old models for new ones without causing any disruption to the system should be a priority, ensuring minimum modifications are needed during the upgrading process.
- Scalability – In a user suction situation, the system might have to support several concurrent user requests. The Engine should be able to handle this. The system should be easily expandable to accommodate new data.

6.3 High level Design

6.3.1 Architecture Diagram

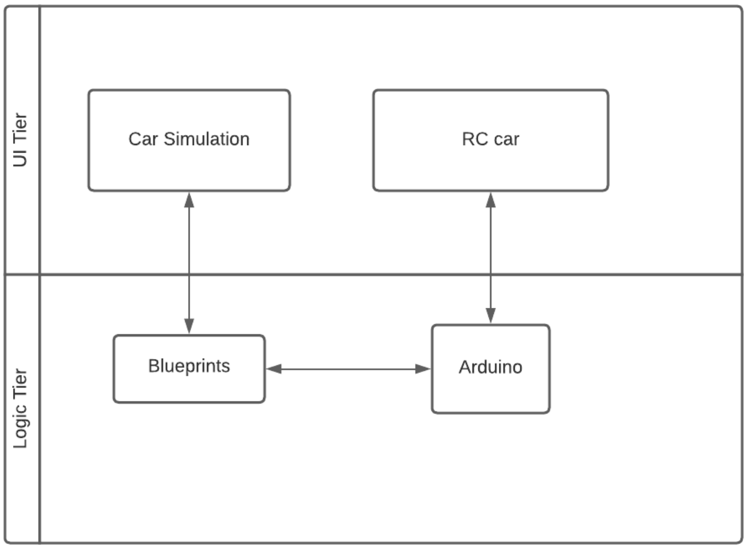


Figure 5: Architecture diagram

6.3.2 Discussion of tiers/ layers of the Architecture

UI Tier

- Car simulation – The application that creates to play the simulation in Unreal Engine with a small Environment.
- RC car – The device that is created by using Arduino to play the car by using the simulation via online.

Logic Tier

- Blueprints – The coding version that is used in the unreal engine to make the design and the movement in simulation.
- Arduino – Tool that helps to transfer the data between Unreal engine and the RC car.

6.4 System Design

6.4.1 Choice of design paradigm

The previse research provides that it was easy by Using OOAD (Object oriented Analysis and design) to build the prototype due to the ease of expendability and further development. The decision was made to use **SSADM** (**Structured System Analysis and Design method**).

6.5 Design Diagrams

6.5.1 Data Flow Diagrams

A more thorough description of the elements of the Context Diagram that was shown in the SRS is provided in the Level 1 Data Flow Diagram that is shown below.

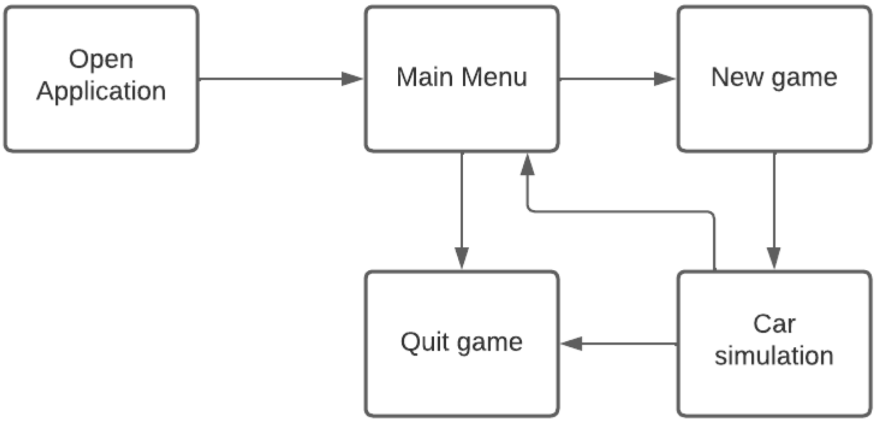


Figure 6: Data Flow Diagram (Level 1)

The Level 2 Data Flow Diagram shown below offers a more thorough description of the elements of the Level 1 Data Flow Diagram mentioned before.

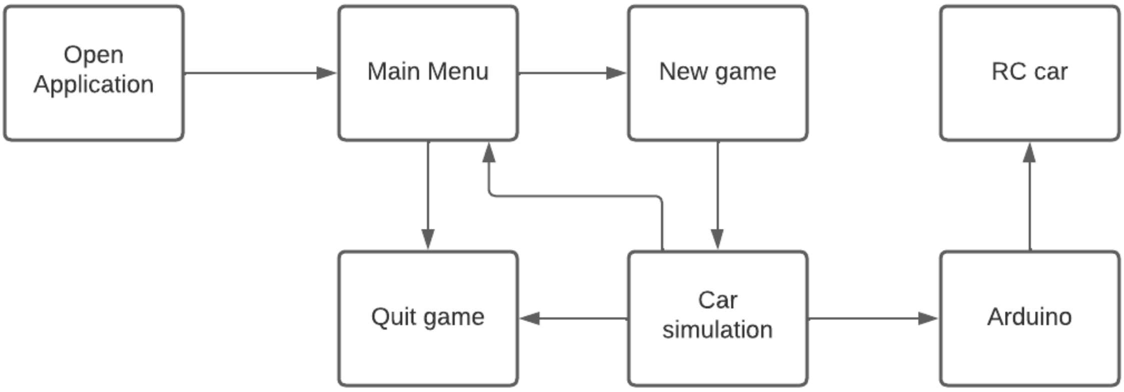


Figure 7: Data Flow Diagram (Level 2)

6.6 Algorithmic Design

The algorithm for this project is used for finding the accuracy rate of the car and passing it to the Simulation. Therefore, another algorithm is also used for cam detection to get the view to the application. The Algorithm will be finalized over the following weeks.

6.6.1 System Process Flow Chart

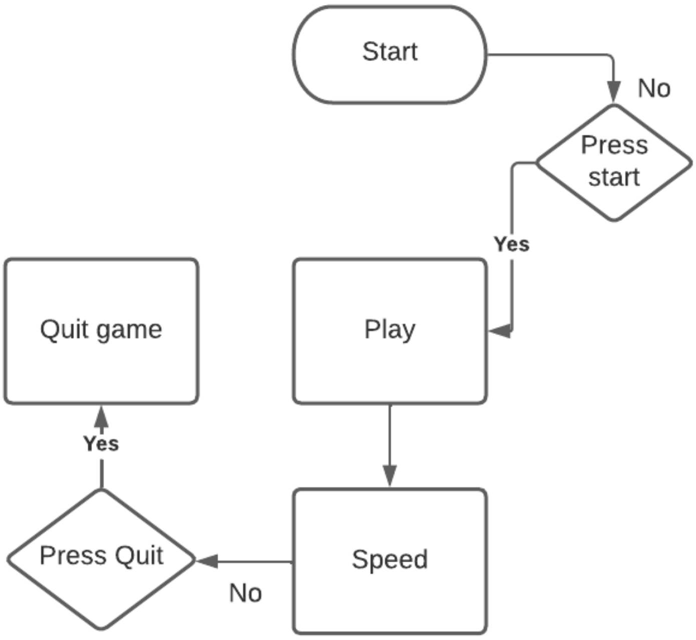


Figure 8: System process flow chart

6.6.2 User Interface Design

UI wireframe was done by the Unreal engine 5 creating a car simulation with an environment to play the device.

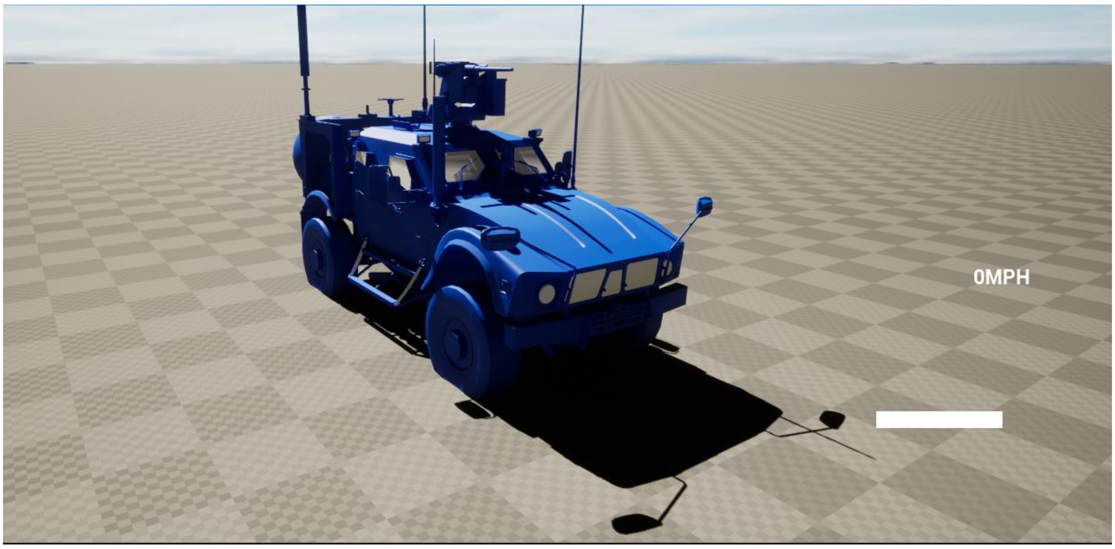


Figure 9: User Interface Design

6.7 Chapter summary

This chapter documents the design and architecture of the project, including the flow and any unique algorithms created by the author. It also includes the planned user interface wireframes for the end-user's interaction with the system.

Chapter 07: IMPLEMENTATION

7.1 Chapter Overview

This chapter will explain the main implementation of the prototype with the technologies and Languages and Supporting tools used for development of the prototype, with the reasoning to the choice of each selection.

7.2 Technology Selection

7.2.1 Technology Stack

Windows will be the default choice for the development since Unreal Engine is available for itself and it may take a high ram and graphic to show the UI. The following layers show the technologies that are used for the prototype.

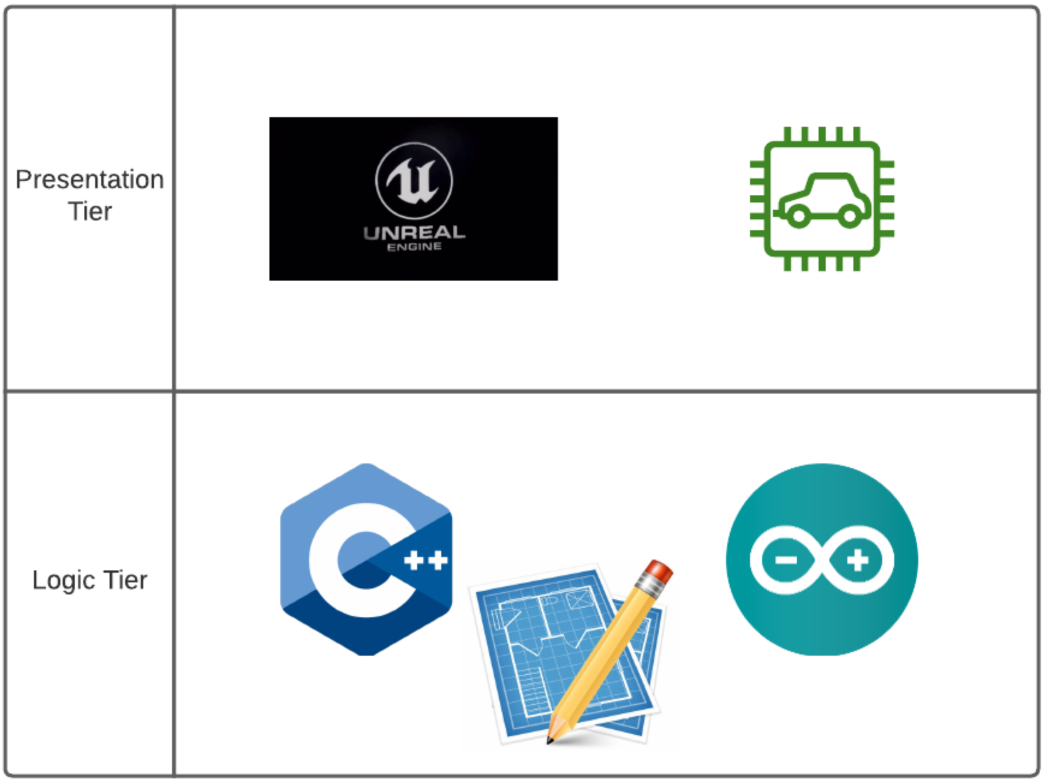


Figure 10: Technology Stack

7.2.2 Development Frameworks

Unreal Engine is the Latest Engine that Created games. It will be easy to create a simulation with the help of this engine. Since this is a new project, running the IOT part small RC car will be helping to do the simulation.

7.2.3 Programming Languages

C++ is used to build the connection between Unreal and Arduino to Play the device through the Application. For the Simulation **Blueprint** will be used for creating the movement and speed in the simulation.

7.2.4 Summary of Technology Selection

Component	**Tools**
Programming Languages	C++, Blueprint
Development Framework	Arduino
UI Framework	Unreal Engine 5

Table 14: Summary of Technology Stack

7.3 Implementation of the Core Functionality

Since this is a unique project, the code of the movement UI part of the car simulation can be shown and described. The movements are coded with the help of blueprint.

7.4 Chapter Summary

In this chapter, the technologies, programming languages, and tools used to build the research prototype are outlined. The author provides explanations and code samples of the key functions, as well as their personal reflection on the completed prototype.

Chapter 8: Testing

8.1 Chapter Overview

In this chapter, the discussion will focus on the testing process of the Unreal RC simulation and how it will be connected to the Arduino using serial COM. The primary focus will be on evaluating the model's performance with building the connection with Arduino and Unreal engine.

8.2 Objectives and Goals of Testing

The RC simulation testing using Unreal, and the Arduino aims and objectives are to:

- Verify the RC simulation's functioning. This involves assessing the simulation's ability to be correctly controlled by the Arduino, its ability to faithfully reproduce the behavior of the real-world RC car, and its use in the testing of various control schemes.
- Find and repair any RC simulation problems. Testing for unexpected behavior, such as the simulation crashing or failing to react to user input, is part of this process.
- Evaluate the RC simulation's performance. This involves calculating the simulation's running duration, CPU efficiency, and memory consumption.
- Improve the user experience of the RC simulation. This includes testing for usability issues, such as the simulation being difficult to control or the simulation being visually unappealing.

The RC simulation used, and its intended use will determine the tests that are carried out in detail. However, a few frequent examinations are as follows:

- Functional testing: These tests ensure that the Arduino can control the RC simulation as intended and that the simulation can faithfully mimic the behavior of the actual RC vehicle.
- These tests ensure that the RC simulation operates appropriately at the input space's borders. A boundary-value test, for instance, may confirm that if the Arduino is detached, the simulation ceases to respond to human input.
- Performance evaluations: These evaluations track the RC simulation's execution time, CPU consumption, and memory usage.
- User evaluations of the usefulness of the RC simulation are conducted through usability tests. For instance, in a usability test, users could be asked to complete a series of activities in the simulation before rating the tasks' complexity.

8.3 Testing Criteria

Following are some evaluation standards for RC simulation using Unreal and Arduino platform:

- Tests of the motors utilized in the RC simulation's power and torque are included in this. By evaluating how rapidly the motors can accelerate the Arduino, you may, for instance, gauge their strength.
- Testing of communication: This includes examining how well the Arduino and the Unreal engine communicate. As an instance, you may send a command from the Unreal engine to the Arduino to test the communication, and then confirm that the Arduino accurately received the instruction.
- Collision testing entails evaluating the RC simulation's collision detection and reaction. By smashing the Arduino into a simulated object, for instance, you may test the collision detection and ensure that it stops traveling as predicted.
- Performance testing: This includes evaluating the RC simulation's performance in various scenarios. By adding more items to the simulation, for instance, and checking that the simulation does not noticeably slow down, you may assess the simulation's performance.

8.4 Functional Testing

Test case	FR ID	User action	Expected result	Actual result	Result status
1	FR1	The car movements in the Unreal engine and the speedometer.	Movements are done with the given command and the speedometer works with the speed that car goes.	Player able to move the car when give command key.	Passed
2	FR2	Open a port to Arduino	A port connection must be open when	Player be able to open a connection	Passed

			the unreal engine is run.	to the Arduino when unreal engine is running.	
3	FR3	RC car movements	When the port connection has running RC car has to be move.	When the player gives the commands in unreal engine the RC car must be moved though Arduino.	Passed

Table 15: Functional testing

8.5 Module and Integration Testing

Module	Input	Expected output	Actual output	Status
Motor testing	Power accelerating	Get power from batteries to run the car.	12V must give to power up the car.	passed
Communication testing	Communication between unreal and Arduino.	Get a port connection from Unreal to Arduino	When the Unreal engine is running a serial connection	passed

			has been opened.	
Performance testing	Test the performance in the RC simulation.	The car must be moved by given command in unreal engine.	When the player gives command to move forward the car is going forward.	passed

Table 16: Integration testing

8.6Non-Functional Testing

8.6.1 Accuracy Testing

The pictures show how to build the connection between Unreal engine and Arduino and passed the data.

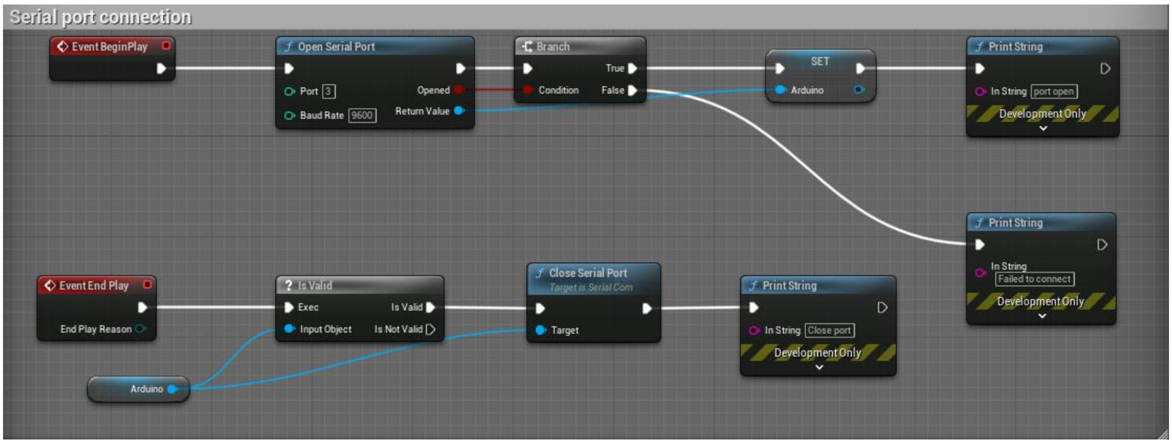


Figure 11: Accuracy testing

8.6.2 Performance Testing

The below picture shows how performance works when the player gives commands in Unreal engine.

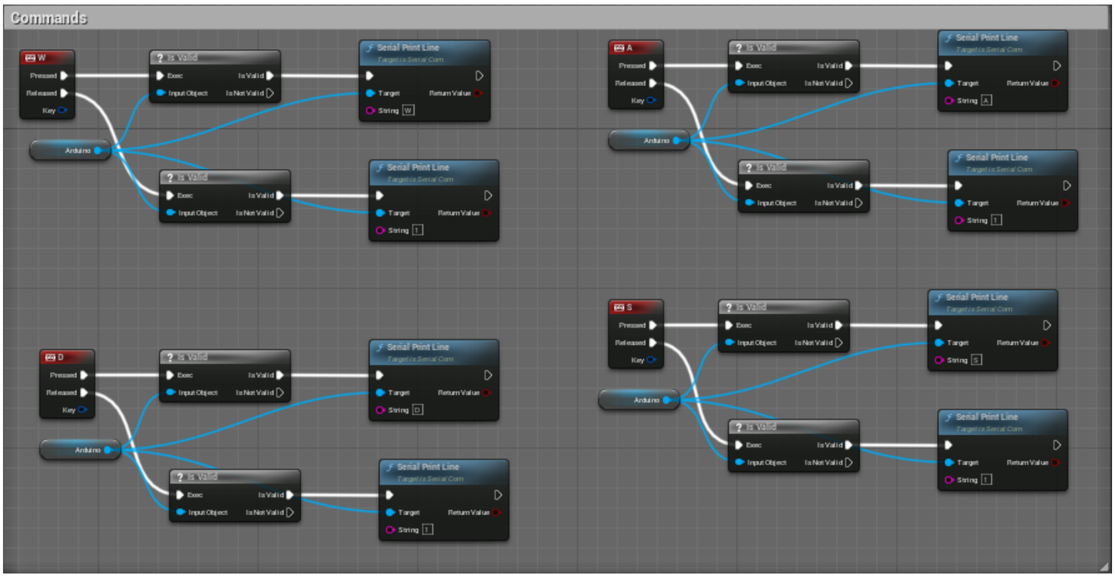
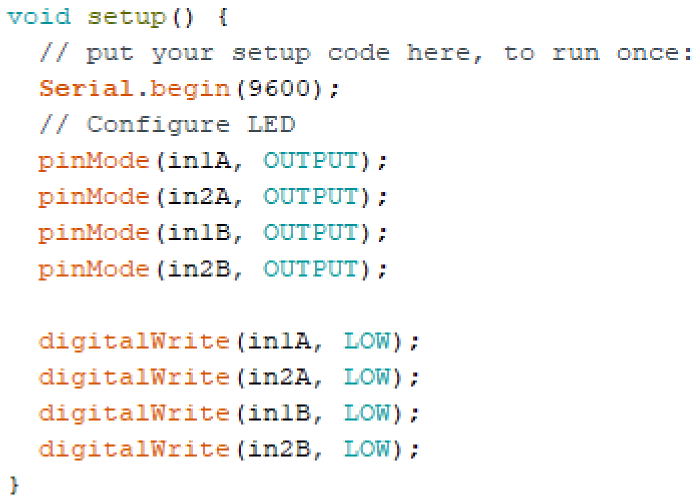


Figure 12: Performance testing

8.6.3 Security Testing

Since the connection applied only for one car at a time there are no securities, but the serial number has been given to find the correct connection to pass the data between Arduino and Unreal



```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  // Configure LED
  pinMode(in1A, OUTPUT);
  pinMode(in2A, OUTPUT);
  pinMode(in1B, OUTPUT);
  pinMode(in2B, OUTPUT);

  digitalWrite(in1A, LOW);
  digitalWrite(in2A, LOW);
  digitalWrite(in1B, LOW);
  digitalWrite(in2B, LOW);
}
```

Figure 13: Security

8.7 Limitations of the testing process

The testing procedure for RC simulation using Unreal and Arduino has a few restrictions. These restrictions consist of:

- The simulation's complexity: The harder it is to completely test, the more complicated the simulation. This is since more potential outcomes must be considered.
- The necessary time and materials are: An RC simulation's testing process can be time-consuming. This is particularly true for extensive and intricate simulations.
- Bug detection challenges: Even in thoroughly tested models, bugs might be hard to uncover. This is since several circumstances, some of which are quite subtle, can lead to defects.
- The simulation's dynamic nature: As new features are introduced or as the simulation is updated, the simulation may vary over time. To verify that the simulation continues to function as predicted, the testing procedure must be repeated.
- Despite these restrictions, testing is still a crucial step in creating an RC simulation. You may assist in guaranteeing that your simulation is of the greatest quality and that it will perform as intended by adhering to appropriate testing procedures and employing a range of testing techniques.

Despite these restrictions, testing is still a crucial step in creating an RC simulation. You may assist in guaranteeing that your simulation is of the greatest quality and that it will perform as intended by adhering to appropriate testing procedures and employing a range of testing techniques.

8.8 Chapter Summary

In this chapter, the connection between Unreal and Arduino, the accuracy rate of RC cars and testing process on the car motor was discussed. The primary focus of this project was how strong connection can get from unreal to Arduino to passing the data using Serial Communication. Overall, the connection was successfully evaluated the performance in the car proving that not only remotely also can control by running a simulation in the PC too.

Chapter 9: Evaluation

9.1 Chapter Overview

This chapter presents the RC simulation based on Unreal through Arduino, a car that is controlled by a PC using car simulation. The evaluation includes methodology and approach, criteria, self-evaluation, selection of evaluators, evaluation results, focus group testing, limitations, and assessment of functional and non-functional requirements. This chapter also discusses the in-scope and out-of-scope aspects of the project.

9.2 Evaluation Methodology and Approach

The following phases would make up the assessment technique and strategy for RC simulation based on Unreal through the Arduino:

- Define the measures for evaluation. This would entail figuring out the key performance indicators (KPIs) that would be used for the simulation's assessment. Several potential KPIs include:

1. The simulation's ability to accurately capture the behavior of the RC vehicle in the actual world.
2. the simulation's computational efficiency.
3. how simple the simulation is to use.

- Create the assessment experiment designs. To measure the KPIs, a series of experiments would need to be created. As an illustration, one experiment would entail operating the RC vehicle in a virtual setting and contrasting its behavior there with its behavior in the actual world.
- Put the assessment experiments into action. Running the assessment trials and gathering the data would be required.
- Examine the evaluation results. To do this, statistical techniques would be used to evaluate the data and spot any trends or patterns.
- Analyze the evaluation's findings. This would entail analyzing the evaluation's findings and making judgments on the simulation's efficacy.

9.3 Evaluation Criteria

The assessment standards for an Arduino-based RC simulation that uses Unreal will vary depending on how the simulation is used. However, a few potential evaluation standards are as follows:

- Accuracy: The simulation's ability to accurately capture the behavior of the RC vehicle in the actual world. This might be assessed by contrasting the simulation's representation of the vehicle's behavior with the vehicle's actual behavior.
- computational performance: The simulation's computational performance, including the memory use and frame rate. By running the simulation on several hardware platforms and evaluating the results, this might be determined.
- Usability: The simulation's usability, including the documentation and user interface. Users might be asked to test out the simulation and provide comments to help determine this.
- Realism: The simulation's realism, including the visuals and physics. Users' ratings of the simulation's realism might be used to determine how realistic it is.
- Immersion: The degree of immersion offered by the simulation, including a sensation of control and presence. Users' evaluations of the simulation's immersion might be used to gauge this.

9.4 Self-Evaluation

I believe that my evaluation of the RC simulation running on Unreal via Arduino was successful. I've determined the important assessment measures, developed, and performed evaluation experiments, examined evaluation data, and evaluated evaluation outcomes. I have also included the extra factors for the evaluation strategy and technique consideration.

I think my selection of assessment criteria is pertinent to the simulation's intended usage. They can be measured and quantified, and they are constant across hardware systems and users. Sure, the evaluation's findings will help to make the simulation better. I'll utilize the results to determine how to enhance the simulation's accuracy, computing efficiency, and usability. Though I am still growing, I learn something new every day. In the future, I'll be able to sharpen my evaluating abilities.

9.5 Selection of the Evaluators

The following factors should be taken into consideration while choosing the assessors for RC simulation using Unreal and an Arduino:

- Knowledge: The assessors should be knowledgeable in Arduino, Unreal Engine, and RC simulation.
- Experience: The assessors should have previous knowledge of utilizing simulations and RC vehicles.
- Bias: The simulation's success or failure should not be influenced by the assessors' personal interests.
- Communication abilities: The assessors must be able to convey their conclusions in a clear and straightforward manner.
- The unique application of the simulation will determine the exact criteria that are utilized to choose the evaluators. The above-mentioned characteristics, however, offer a decent place to start when choosing assessors for RC simulation using Unreal and Arduino.

9.6Evaluation Result

9.6.1 Expert Opinion

To evaluate the proposed system, the opinions of 3 technical experts and 3 domain experts were obtained. These experts provided their insights on the concept and the solution.

9.6.1.1 Domain Experts

Expert	Background	Opinion on concept	Opinion on solution
Dakshina wijeykulathilaka	Senior game developer – Mogo games	Believes that concept has the potential to revolutionize the Unreal industry.	Recommends the addition of more customization options like gear systems.

Dimuth Karunarathne	Game developer – Mogo games	Thinks that the concept will help professionals.	Appreciates the use of advanced technologies like Unreal engine for create simulation.
Kasun Kavishka	4th year IOT student in SLTC campus	Believes the concept will be popular among RC players and Kids.	Suggest for implement a camera for get a real view.

Table 17: Domain experts

9.6.1.2 Technical Experts

Expert	Background	Opinion on scope	Opinion on the implementation of the solution
Dakshina wijeykulathilake	Senior game developer – Mogo games	Very impressive use case that has the potential to be integrated and monetized for RC players	Very good use of off-the-shelf components. Demonstrated understanding of underlying technologies by creating an application that can be used by the consumer market.

Dimuth Karunarathne	Game developer – Mogo games	The project has a significant scope in the field of personalized grooming and can be extended to other domains as well.	implementation is well-executed, with attention to detail in aspects

Table 18: Technical expert

9.7 Limitations of Evaluation

The Arduino and Unreal-based RC simulation has the following drawbacks:

- Sometimes the simulation environment is off. The simulation's physics engine might not be able to accurately reflect the RC car's actual properties. This might result in simulation mistakes, which could make it challenging to assess the RC car's performance.
- Complex simulations cannot be done on the Arduino platform since it is underpowered. Since the Arduino is a microcontroller, its processing speed and memory are constrained. The complexity of the simulations that may be done on the Arduino may be constrained as a result.
- Simulators may be tricky to troubleshoot. It may be challenging to identify the simulation's error's root cause if there are errors. This is due to the lack of a graphical user interface on the Arduino, which makes it challenging to see the simulation in action.
- Despite these drawbacks, an Arduino-based RC simulation that uses Unreal Engine might be a valuable tool for assessing the performance of RC vehicles. Users can be aware of potential mistakes and take action to reduce them by being aware of the simulation's limits.

9.8 Evaluation on Functional Requirements

The following criteria may be used to assess the functional needs of an RC simulation based on Unreal through the Arduino:

- Speed: By keeping track of the RC car's distance traveled over a predetermined amount of time, the average speed of the vehicle may be calculated.
- Acceleration: By keeping track of the RC car's change in speed over a certain amount of time, the average acceleration of the vehicle can be calculated.
- Turning radius: The RC car's typical turning radius may be calculated by keeping track of the distance it travels when turning around a given point.
- Success rate: The number of times an RC car completes a job successfully may be used to calculate the success rate of the vehicle.

9.9 Evaluation on Non-Functional Requirements

Non-functional requirements (NFRs) are specifications that specify how a system ought to behave rather than what it ought to perform. Although they are sometimes disregarded, they are just as crucial as functional needs. Some of the significant NFRs for RC simulation include:

- Performance: For a fluid and responsive experience, the simulation should be able to operate at a high frame rate.
- Accuracy: The simulation should reflect real-world physics as closely as feasible.
- Scalability refers to the simulation's capacity to expand to accommodate more users and vehicles.
- Security: The simulation must be protected from tampering or illegal access.

9.10 Chapter Summary

This chapter covered the evaluation process, methodology, criteria, and results for RC simulation Based on Unreal through Arduino. Expert judgments, focus group testing, functional and non-functional needs, and self-evaluation were all included in the evaluation. The outcomes showed that the approach was successful in achieving its goals while also pointing up possible areas for development and improvement in subsequent iterations.

Chapter 10: Conclusion

10.1 Chapter Overview

In this chapter, the project's accomplishments, difficulties, and overall addition to the corpus of knowledge are summarized. It covers how to use what you learned in class, how to use what you already know, and how to pick up new skills as you go along. It also describes the research's limits, any departures from the original goal, and prospective future improvements.

10.2 Achievements of Research Aims & Objectives

Research Objectives	Description	Learning Outcomes	status
Literature Review	To achieve the following objectives, a review of the body of literature is applied. • To specify evaluation standards and metrics. • To Identify the research gaps on the project. • To systematically analyze the existing system in the Arduino and Unreal.	L01, L06	passed

Requirement Elicitation	There was a thorough review of the prerequisites to, • To use surveys to specify the requirements for hardware and software. • To compile the demands of researchers and domain professionals to comprehend what they anticipate the system to achieve.	L02, L03, L04, L05, L06	passed
Design	The proposed framework for cognitive networking's design intends to, • To create a driving environment for an unreal world.	L03, L04, L06	passed
Implementation	creation of the conceptual model in accordance with the architectures and models created during the design stage using the determined hardware and software components. • To develop the task switching mechanism.	L03, L04, L06	passed
Evaluation	Putting the prototype to the test and having researchers and specialists in the field examine it, • To evaluate how the system performs in the real world and simulations. • To test individual components of the prototype	L04, L06	passed

Table 19: Achievements of Research Aims & Objectives

10.3 Utilization of Knowledge from the Course

- C ++ - Implement to build the connection between Unreal and Arduino.
- Unreal Blueprint - Helped to create the Car movements in the Unreal engine.
- Unreal Animation – To help to activate the wheels physics in the car.
- Blender 3D – To create a 3D model car to use to simulation.

10.4 Use of Existing Skills

Throughout the project, a wide range of skills acquired from the course were effectively applied, including Unreal Blueprint, Arduino IDE, and C++. existing skills were crucial in achieving the project objectives and demonstrating a strong foundation in various computer science disciplines.

10.5 Use of New Skills

During the project, these new skills were applied:

- Serial Com plugin – This plug is helpful for opening a port between Unreal engine and the Arduino to start a connection to pass the data.
- ESP32 – This platform will be helpful for collecting the data from Arduino IDE and pass the data to the vehicle to activate the car.

The application of these newly acquired skills not only enhanced the project's outcome but also broadened the researcher's technical expertise, enabling further research and development opportunities.

10.6 Achievement of Learning Outcomes

The relative completion statuses of the study goals listed in the INTRODUCTION reflect whether they have been met. The Achievement of Learning Outcomes table has these statuses.

10.7 Problems and Challenges Faced

The Unreal-based RC simulation using the Arduino can be a difficult undertaking. The following are some issues and difficulties that could be encountered:

- Hardware restrictions: Since the Arduino is a microcontroller, its processing speed and memory are constrained. Because of this, simulating complicated RC vehicles or settings may be challenging.
- Software restrictions: Unreal is a strong gaming engine but using it for RC simulation might be challenging. The learning curve can be severe, and there aren't many resources accessible.
- Communication problems: Data interchange requires Arduino and Unreal to be able to interact with one another. Given that the two systems make use of distinct protocols, this can be difficult.
- Real-time performance is necessary for RC simulation to deliver a fluid and responsive experience. It can be challenging to accomplish this, especially with weak hardware.

RC simulation using Unreal through the Arduino may be beneficial despite these difficulties. Realistic and captivating simulations that may be used for training, education, or entertainment can be made with proper preparation and execution.

10.8 Deviations

Deviations in RC simulation based on Unreal through the Arduino can be caused by a few factors, including:

- Imperfect modeling of the physical world. The Unreal engine is a powerful tool, but it is not perfect. The physics engine that it uses is based on several assumptions, and these assumptions may not always be accurate. This can lead to deviations in the simulation, especially when the simulation is dealing with complex physical phenomena.
- Imperfect communication between the Arduino and the Unreal engine. The Arduino is a microcontroller, and it is not as powerful as a computer. This means that it can be difficult to send and receive data between the Arduino and the Unreal engine in real time. This can lead to delays and other errors, which can cause deviations in the simulation.
- Imperfect sensors and actuators. The Arduino is connected to the real world through sensors and actuators. These sensors and actuators are not perfect, and

they can introduce errors into the simulation. For example, a sensor may not be able to measure the position of a motor accurately, or an actuator may not be able to apply the correct amount of force.

10.9 Limitations of the Research

There are several restrictions on the study being done in RC Simulation Based on Unreal using the Arduino.

- The simulation falls short of actual RC racing in terms of realism. The simulation might feel a little fake because the physics engine in Unreal Engine is not as advanced as the mechanics of a genuine RC car. The simulation may also be sluggish and jerky because the Arduino is not as powerful as a specialist RC controller.
- The investigation is still ongoing. Some parts of the simulation have not yet been included since the research team is still trying to improve it. For instance, multiplayer racing is not currently supported by the simulation.
- The study is still not readily accessible. It is not yet feasible to use the simulation for yourself because the study team has not yet made it available to the public.

Despite these drawbacks, research into RC Simulation Based on Unreal using the Arduino is a promising first step in the direction of producing more immersive and realistic RC racing experiences. The simulation is anticipated to improve in realism and accessibility as the study goes on.

10.10 Future Enhancements

A strong gaming engine, Unreal Engine may be used for RC simulation among other things. By incorporating real-world physics and sensor data into your RC simulations using Arduino, you may further improve their realism.

Future improvements to Arduino-based RC simulation with Unreal might include the following:

- Real-time physics: Arduino may be used to gather real-time sensor data from devices like magnetometers, accelerometers, and gyroscopes. The physical forces operating on the RC vehicle may then be simulated in Unreal Engine using this data. The simulations would become more immersive and lifelike as a result.

- Arduino may also be used to combine sensor data from different sensors. This would enable you to produce simulations of the environment that the RC car is operating in that are more accurate. For instance, you might combine information from an accelerometer, gyroscope, and magnetometer to provide a simulation of the vehicle's orientation that is more precise.
- AI-powered control: Control algorithms for the RC vehicle may be implemented using Arduino. This would enable the car to make judgments based on information from its sensors and the surrounding area. You could, for instance, use an AI system to enable the car to avoid impediments.

10.11 Achievement of the contribution to body of knowledge

A strong gaming engine, Unreal Engine may be used for RC simulation among other things. Realistic and engaging RC simulations that may be used for teaching, research, and enjoyment can be made using Unreal Engine and Arduino.

The ability to create incredibly complex and realistic surroundings is one of the key advantages of utilizing Unreal Engine for RC simulation. This is crucial for training reasons since it gives users the chance to practice driving in real-world scenarios in a secure setting. Additionally, for entertainment reasons, immersive and captivating RC simulators may be made using the realistic visuals of Unreal Engine.

The fact that Unreal Engine is quite flexible is another advantage of adopting it for RC simulation. To meet their unique demands, users may simply design their own RC models and surroundings or alter already-existing ones. Unreal Engine is a valuable tool for research because of its amount of customization, which enables users to test various RC models and surroundings in a range of scenarios.

Overall, the Unreal Engine and Arduino combo are potent tools for RC simulation. It enables the development of interactive, realistic, and immersive RC simulations that may be used for education, research, and entertainment.

10.12 Concluding Remarks

An innovative new technique that has the potential to completely change how we engage with RC cars is RC simulation powered by Unreal through the Arduino. We can develop RC simulations that are both realistic and engaging by fusing the flexibility of the Arduino platform with the power of Unreal graphics engine.

There are several possible uses for this technology, including:

- Training and education: RC mechanics and pilots may be trained via simulators. They may also be used to teach others the fundamentals of radio-controlled flight.
- Entertainment: New and fascinating RC games and experiences may be made with RC simulators.
- Research: RC simulations may be used to examine how RC vehicles behave in various settings.

The potential advantages of this technology are obvious, but the development of RC simulation based on Unreal through the Arduino is still in its early phases. This technology has the potential to improve how we interact with RC vehicles with further development.

References

Alcalá, E., Puig, V., Quevedo, J. and Rosolia, U. (2020). Autonomous racing using Linear Parameter Varying-Model Predictive Control (LPV-MPC). *Control Engineering Practice*, 95, p.104270. doi:10.1016/j.conengprac.2019.104270.

Pagot, E., Piccinini, M. and Biral, F. (2020). *Real-time optimal control of an autonomous RC car with minimum-time maneuvers and a novel kineto-dynamical model*. [online] IEEE Xplore. doi:10.1109/IROS45743.2020.9340640.

Ashton, J., Hunt, S. and Spencer, M. (n.d.). *WORCESTER POLYTECHNIC INSTITUTE MAJOR QUALIFYING PROJECT Autonomous RC Car Platform*. [online]

Available at: https://web.wpi.edu/Pubs/E-project/Available/E-project-032219-094723/unrestricted/MQP_Final_Paper.pdf.

Unreal Engine Forums. (2021). *[NEW-Free] Arduino Serial Communication Plugin 'Serial COM v3'-(Fork from UE4Duino)*. [online] Available at: https://forums.unrealengine.com/t/new-free-arduino-serial-communication-plugin-serial-com-v3-fork-from-ue4duino/265486 [Accessed 2 Nov. 2022].

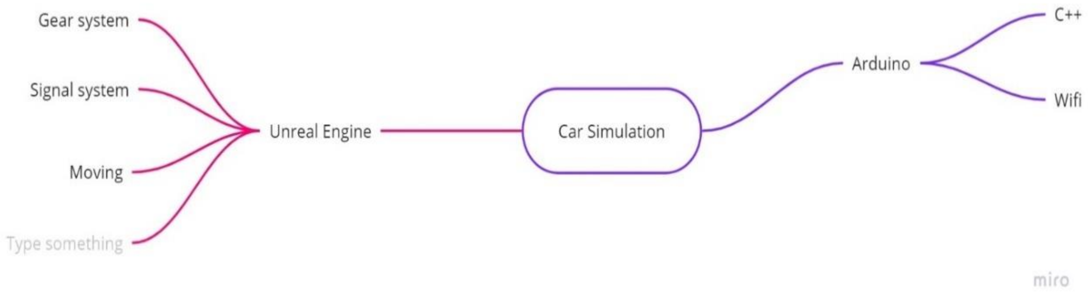
mlab-upenn.github.io. (n.d.). *F110 Autonomous Race Cars*. [online] Available at: https://mlab-upenn.github.io/f110/ [Accessed 2 Nov. 2022].

control.ee.ethz.ch. (n.d.). *Autonomous RC car racing*. [online] Available at: https://control.ee.ethz.ch/research/team-projects/autonomous-rc-car-racing.html [Accessed 2 Nov. 2022].

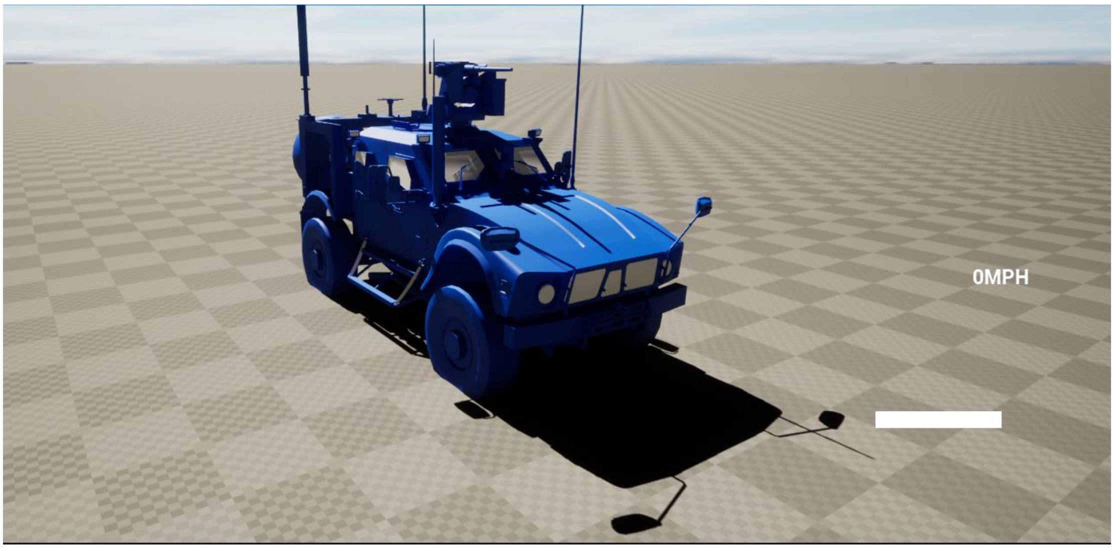
Brown, M., Funke, J., Erlien, S. and Gerdes, J.C. (2017). Safe driving envelopes for path tracking in autonomous vehicles. *Control Engineering Practice*, 61, pp.307–316. doi:https://doi.org/10.1016/j.conengprac.2016.04.013.

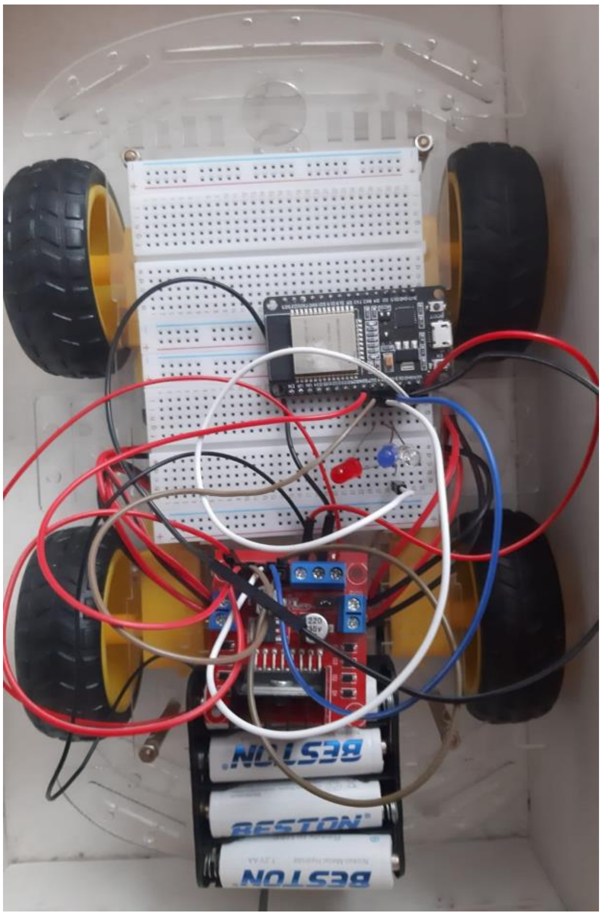
Caporale, D., Fagiolini, A., Pallottino, L., Settimi, A., Biondo, A., Amerotti, F., Massa, F., De Caro, S., Corti, A. and Venturini, L. (2018). *A Planning and Control System for Self-Driving Racing Vehicles*. [online] IEEE Xplore. doi:https://doi.org/10.1109/RTSI.2018.8548444.

Appendix A – Concept map

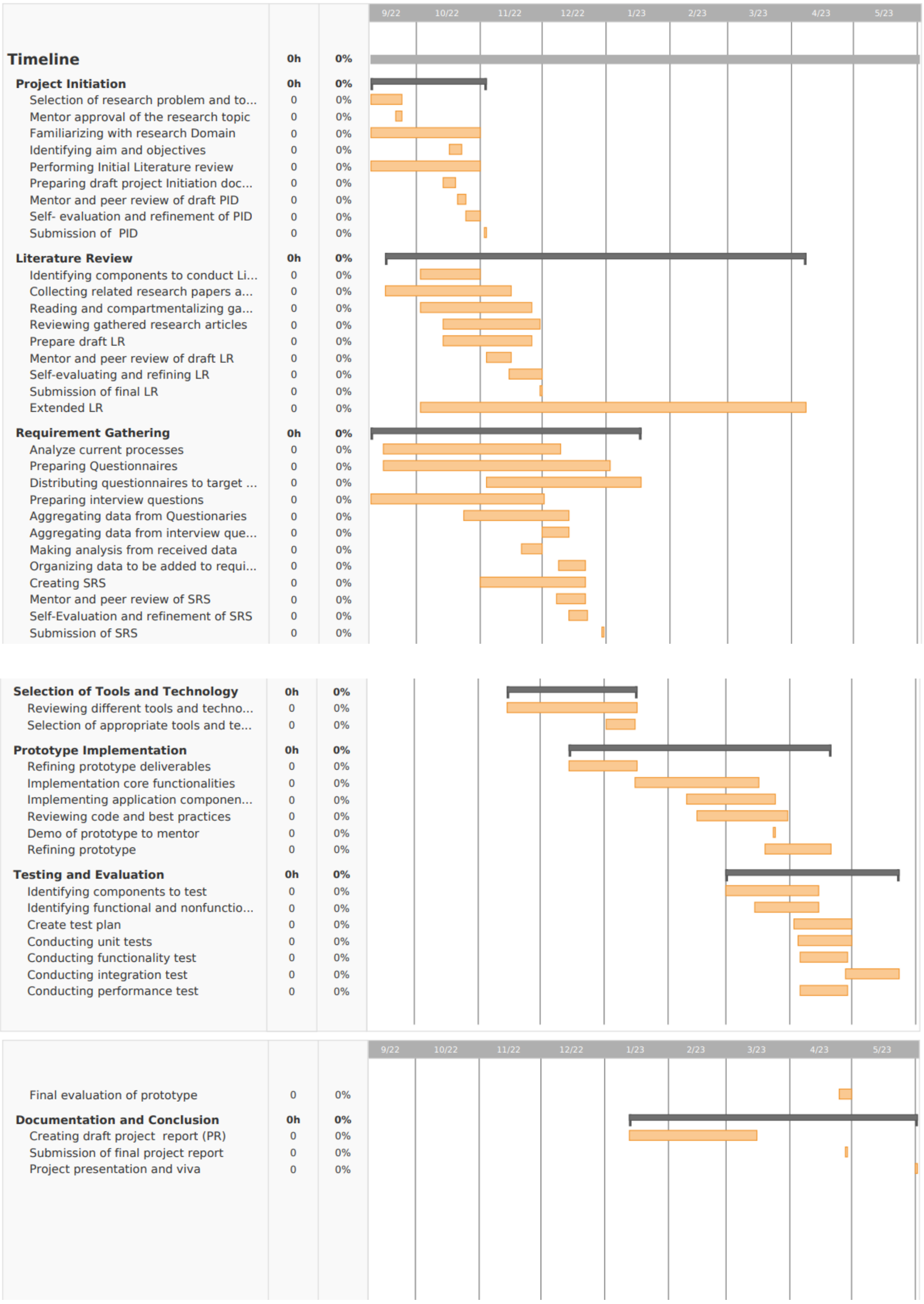


Appendix B – Prototype





Appendix C – Gantt chart



Timeline	**0h**	**0%**
Project Initiation	**0h**	**0%**
Selection of research problem and to...	0	0%
Mentor approval of the research topic	0	0%
Familiarizing with research Domain	0	0%
Identifying aim and objectives	0	0%
Performing Initial Literature review	0	0%
Preparing draft project Initiation doc...	0	0%
Mentor and peer review of draft PID	0	0%
Self- evaluation and refinement of PID	0	0%
Submission of PID	0	0%
Literature Review	**0h**	**0%**
Identifying components to conduct Li...	0	0%
Collecting related research papers a...	0	0%
Reading and compartmentalizing ga...	0	0%
Reviewing gathered research articles	0	0%
Prepare draft LR	0	0%
Mentor and peer review of draft LR	0	0%
Self-evaluating and refining LR	0	0%
Submission of final LR	0	0%
Extended LR	0	0%
Requirement Gathering	**0h**	**0%**
Analyze current processes	0	0%
Preparing Questionnaires	0	0%
Distributing questionnaires to target ...	0	0%
Preparing interview questions	0	0%
Aggregating data from Questionaries	0	0%
Aggregating data from interview que...	0	0%
Making analysis from received data	0	0%
Organizing data to be added to requi...	0	0%
Creating SRS	0	0%
Mentor and peer review of SRS	0	0%
Self-Evaluation and refinement of SRS	0	0%
Submission of SRS	0	0%
Selection of Tools and Technology	**0h**	**0%**
Reviewing different tools and techno...	0	0%
Selection of appropriate tools and te...	0	0%
Prototype Implementation	**0h**	**0%**
Refining prototype deliverables	0	0%
Implementation core functionalities	0	0%
Implementing application componen...	0	0%
Reviewing code and best practices	0	0%
Demo of prototype to mentor	0	0%
Refining prototype	0	0%
Testing and Evaluation	**0h**	**0%**
Identifying components to test	0	0%
Identifying functional and nonfunctio...	0	0%
Create test plan	0	0%
Conducting unit tests	0	0%
Conducting functionality test	0	0%
Conducting integration test	0	0%
Conducting performance test	0	0%
Final evaluation of prototype	0	0%
Documentation and Conclusion	**0h**	**0%**
Creating draft project report (PR)	0	0%
Submission of final project report	0	0%
Project presentation and viva	0	0%