Module 5- Computer Systems (2021-22)
Project

UNIVERSITY OF TWENTE.

Requirement Analysis Document

Project Name: RC Racing System	**Team Members:** Rick Pluimers, Kağan Gülsüm, Kristiyan Velikov, Laurens Neinders, Aleksandar Petrov, Liran Neta
Team ID: 22	**Mentor(s):** Mohammed Assaad and Alex Mo

1. Introduction

1.1. **Purpose**:

The project, namely RC Racing, was selected for the following reasons:

- *Implementation of a system using hardware (sensors connected to RPi)*
- *Many young people are fans of RC cars*
- *High-Quality timing-recording*
- *An opportunity to make the RC racing competitive among friends in the same place*
- *Our love to car races*
- *Probably possible to finish before the deadline*

Our Web Application enables management of racing, also among friends. The timings of every lap and sector will be recorded and shown to the users. We aim to give the users data that will drive them even more to play and help them to figure out what are their pros and cons in their racing sessions.
The Web Application can be a great entertainment for individual RC lovers, friends, and even as a family activity.

1.2. **Limitations of the current system**:

The current limitations of already existing rc racing systems are:

- *The system does not give detail about sector times throughout the lap.*
- *The system does not provide the option to have challenges between users.*
- *The UI is not user-friendly.*

1.3. **Intended Audience:**

1. *Alex Mo and Mohammed Assad (Mentors)*
2. *Eric Tews (Module Coordinator)*
3. *Dipti Sarmah (Project Coordinator)*
4. *Rick Pluimers, Kristiyan Velikov, Laurens Neinders, Liran Neta, Aleksandar Petrov, Kağan Gülsüm (Project Members)*
5. *People who like/want to try out rc car racing.*

1.4. **Define SMART Goals**:

Specific (What)	Measurable (Up to)	Attainable (How)	Relevant (Why)	Time-bound (when)
*1. To improve the **productivity** of the system by using motion sensors to detect the rc car going past and save the current time.*	*To evaluate the success of the motion detection and the time saved.*	*To test the system with different lap times.*	*To ensure the relevant data is strictly correct for the current lap.*	*Week 6-7*
*2. To improve the **functionality of the UI** by providing a leaderboard with all recorded times.*	*To evaluate the correctness of the times displayed from fastest to slowest.*	*To test the system with different recorded times.*	*To ensure no errors in the leaderboard are introduced.*	*Week 6-7*
*3. To improve the **system functionality** by providing an option to challenge other users to a race.*	*To evaluate the success of issuing a challenge and recording the result.*	*To test the system with many challenge pairs.*	*To ensure fair and correct results are displayed after 1V1 sessions.*	*Week 7-8*
*4. To improve the **Sector's timing comparison functionality** by providing sector by*	*To evaluate the comparison's correctness of different times which represent the*	*To test if the best timing among the given timings of a sector is actually the best time.*	*To ensure the truthness of the sectors' competition. To ensure that the best driver of the sector is*	*Week 7-8*

sector times and conclude the best time for the current sector.	*sector’s timings.*		*the one who made it in the shortest time span.*	
5. To improve the ***user-friendliness*** *of the system by having different accounts and seeing race data about them.*	*To evaluate the correctness of the data of race times about a particular user and the easy access to it.*	*To test the system with many accounts and many recorded lap times on them.*	*To ensure the accounts do not show incorrect data*	*Week 7-8*
6. To improve the ***correctness*** *of the motion sensors by ensuring time detection of every sensor in the right order.*	*To evaluate the time measurement of the different sensors in the same race*	*To test the time measurement of all the motion sensors on the Race Track and check if each of them was recording in the right order*	*To ensure the decency of the RC race*	*Week 7-8*
7. To improve the ***system-recognizability****, by asking the users if they understand what to do based on the UI.*	*To evaluate how clear is the system and instructions / button displayed to the user*	*To test if the user understands what every button leads to.* *Using User Testing.*	*To ensure the convenient usage of the system, so that the user will be able to operate by himself without confusions*	*Week 5-6*
8. To improve the ***sectors’ time measurement*** *by measuring the time it takes to drive the RC one sector*	*To evaluate the accuracy of the timing measurement it takes to drive the RC between 2 motion sensors*	*To test how much time delay there is in the sector’s time measurement*	*To ensure that the timings of the RC driver are trustworthy*	*Week 7-8*

1.5. **Scope:**

- *System boundaries:*

- *Hardware: Raspberry Pi 4, motion/distance sensors (to be decided), power supply.*
- *Software: Python language, Java language, Java Jersey, Rest API, Javascript language, OAuth, Python DB Connectivity, Java DB Connectivity, PostgreSQL, Python Sockets, Java Sockets, HTML/CSS.*

- *Interfaces:*
  - *WiFi, 4G, etc. for internet connectivity*
  - *Sockets for connectivity between java back-end and raspberry pi*
  - *Python db connectivity for Raspberry Pi & DB communication*
  - *Java db connectivity for back-end & DB communication.*
- *Limitations:*
  - *A maximum of one person can race at any given time.*
  - *Timing won't be represented in real-time, but only after the race is done.*
  - *Timing's accuracy is in the order of tenths due to the use of motion sensors (if we choose that option eventually).*

2. **Product features:**

2.1 **Functional requirements:**

The functional requirements of a ***RC Racing System*** *are:*

- *The user should be able to sign-up.*
- *The user should be able to login.*
- *The user should be able to start a race.*
- *The user should be able to add friends.*
- *The user should be able to remove friends.*
- *The user should be able to challenge a friend.*
- *The user should see his/her times.*
- *The user should see his/her times in sectors.*
- *The user should be able to see a global leaderboard.*
- *The user should be able to see a leaderboard of friends.*
- *The user should be able to see the ongoing time, while he/she is racing.*

2.2 **Nonfunctional requirements:**

The non-functional requirements of the ***RC Racing System*** *are:*

- *The system should perform user authentication in order to log in* ***(Security requirement)****.*
- *As a user, I can quickly and securely log in to the system using a 3rd party account (for example Google)* ***(Security requirement)****.*
- *As a user, I can access the UI I have logged in* ***(Security requirement)****.*
- *As an admin, I want to check the log files to check what happened if something is out of the ordinary* ***(Security requirement)****.*
- *In case my files are deleted by hackers, I want to be able to recover the system's data* ***(Security requirement).***
- *As a user, I don't want my personal information to be known by unknown parties* ***(Security requirement).***
- *The delay of time recording should not exceed 0.5 sec.*
- *The systems should be user-friendly.*
- *The system should not limit the number of users who can sign-up.*
- *The system should not limit the number of users who can be online at the same time.*

3. **Conclusion:**

By brainstorming about different ideas for the project, we decided that the RC Racing System would be a good choice, because of the reasons mentioned in part 1.1. We managed to recognize the upcoming challenges that we would probably encounter, such as: physical connections of cables, sending signals and dealing with delay times. The next phase will include the design, therefore we need to be in the consistent state of mind of "how to make the system secure enough". To better design the system, we need to understand the pros and cons of any decision, such as what sensor will be used, how the data will be delivered from the sensor to the RPi, from the RPi to the DB, and from the DB to the user at the end.

4. **Reference**:

- *Similar Projects Websites*
  - *https://www.livetimescoring.com/*
  - *https://www.mylaps.com/rc-drone-racing/*
- *Java Book*
  - *https://math.hws.edu/javanotes/*
- *Java Jersey Documentation*
  - *https://eclipse-ee4j.github.io/jersey.github.io/documentation/latest/index.html*
- *Javascript Documentation*
  - *https://developer.mozilla.org/en-US/docs/Web/JavaScript?retiredLocale=tr*
- *Python Documentation*
  - *https://docs.python.org/3/*
- *RPi Documentation*
  - *https://www.raspberrypi.org/documentation/*
- *Sensor Documentation*
  - *https://tutorials-raspberrypi.com/raspberry-pi-ultrasonic-sensor-hc-sr04/*
  - *https://maker.pro/raspberry-pi/tutorial/how-to-interface-a-pir-motion-sensor-with-raspberry-pi-gpio*
- *PostgreSQL Documentation*
  - *https://www.postgresql.org/docs/*