

CS CAPSTONE REQUIREMENTS DOCUMENT

NOVEMBER 28, 2018

HYPERRAIL APP

PREPARED FOR

OPENS LAB

CHET UDELL

PREPARED BY

GROUP 25

VINCENT NGUYEN

ADAM RUARK

YIHONG LIU

Abstract

CONTENTS

1	Introduction	2
1.1	Purpose	2
1.2	Scope	2
2	Product Overview	2
2.1	Product Description	2
2.2	User Characteristics	2
2.3	Limitations	2
3	Specific Requirements	3
3.1	Usability Requirements	3
3.2	Database Requirements	3
3.3	Code Quality	3
3.4	Reliability	3
3.5	Efficiency	3
3.6	Integrity	3
3.7	Flexibility	3
3.8	Portability	3
3.9	Safety	4
3.10	Performance Requirements	4
4	Stretch Goals	4
5	Gantt Chart	4

1 INTRODUCTION

1.1 Purpose

The purpose of this document is to provide an overview of the product to be delivered to the client by the end of Spring 2019. The client has listed a number of features, or project requirements, that they wish the capstone team to implement.

1.2 Scope

The hardware for the HyperRail project has already been decided upon and its code has been mostly developed. For this project, the hardware code needs to be optimized and a web-based graphical user interface and central server are needed to complete the software portion of the HyperRail application. The specified features are meant to allow a user to remotely communicate with a specified HyperRail system, which is handled and abstracted through the central server. The features have two levels of priority: required and convenient. Specifications that are required are needed for the system to be fully developed and deployed. Those that are convenient are stretch goals that are not required, but would be nice to have to make the system more robust.

2 PRODUCT OVERVIEW

2.1 Product Description

The HyperRail is a small railway where an automated environmental sensor package, which contains a variety of different sensors, can traverse through a space and collect information as it travels along the railway. Using the HyperRail application, the sensor package's settings can be customized and monitored. The application currently allows the user to specify the speed of the package, current length of the rail, and the size of the spool used for the motors. It also monitors the position of the sensor package along the railway by calculating the number of motor steps it has taken and updates the position display in real-time. However, it currently requires a direct connection, wired or wireless, to the sensor package itself, limiting the HyperRail's use.

2.2 User Characteristics

Users of the HyperRail will be using it to monitor environmental changes. They can monitor levels of carbon dioxide, moisture, light levels, and other variables in the environment depending on the sensors used in the package. No programming experience is required, but the user will need to know how to interact with computers as the application abstracts the programming required to automate the HyperRail system. The user can interpret the data collected in any manner they would like.

2.3 Limitations

The use of the HyperRail is limited to areas with low interference. Because a computer communicates with the sensor package using a radio or Bluetooth connection, there cannot be strong interference disrupting the connection or the data collected by the sensor package will be lost. The HyperRail is also limited to safe or contained environments. Because the HyperRail system utilizes a physical setup, the railway or the sensor package can be damaged by entities in the environment. The computer and sensor package also requires electricity to function. Therefore, the HyperRail is limited to areas with power.

3 SPECIFIC REQUIREMENTS

3.1 Usability Requirements

This application must have a user interface that is intuitive to use. This means that the primary functionality of the app such as defining the parameters of the system's journey and uploading them to the robot should never be more than one click away. Secondary functionality such as saving and loading configurations can be a few clicks away. This is to provide a readable interface that is not cluttered.

The application must also provide feedback to the user when an action succeeds or fails. This can be done with a pop-up containing a status message for the last completed action.

3.2 Database Requirements

There are some attributes that must be stored in a database. First, users must be allowed to log in, therefore users and their encrypted passwords must be stored in a database. Second, we potentially want to allow saved configurations and these must be saved and loaded; this will also require a database.

3.3 Code Quality

The underlying code that will be running this application must be maintainable for teams in the future to improve upon. This means that the code must be homogeneous and well documented. These qualities will allow the any portion of code to be understandable just from looking at its source and to have the reason for its existence documented.

3.4 Reliability

The application needs to allow the user to operate the HyperRail system remotely, using the custom-developed web interface to finish their specific request without directly connecting to the micro-controller.

3.5 Efficiency

The application needs to be able to handle the different commands from multiple users and handle those commands accurately and quickly. The server must be able to handle multiple users connected at once and ensure that their user experience is not affected by any other connected users. User actions must be handled in a timely fashion.

3.6 Integrity

Integrity for this system is defined by how well it can track those commands from different users then handle those commands successfully. This means that user actions must be accountable by either displaying what user triggered each action on a status check, or by storing a log of previous actions performed on an individual system.

3.7 Flexibility

The application should account for future changes. If functions are modified or added to the application, version control should be used to ensure that there is working code.

3.8 Portability

This application should be able to interact and configure new sensor packages without major changes to the software. The software itself should be portable in the sense that mobile devices and personal computers can access the web application and interact with a specified HyperRail system.

3.9 Safety

This application will need to utilize a secure connection to ensure that all content is controlled and private. The system will also only allow registered users to log in and interact with the system.

3.10 Performance Requirements

The essential measurement standard for the performance requirement is if the application is able to satisfy all the requirements and include all of the functionality needed to be deployed. Although incorporating the stretch goals is not mandatory for the application’s release, it will make the application more robust.

4 STRETCH GOALS

One stretch goal is to allow the sensor package to interact with existing actuators in the space. Because actuators are machines that perform simple movements on objects, such as valves or switches, this interaction will give the application additional capabilities depending on the application of the actuators.

Another stretch goal is the capability to configure a fleet of sensor systems to use the same configuration. This allows for quick configuration of a group of sensor packages that are intended for the same purpose.

Another feature that we would like to include is the capability to save and load configurations to allow users to share their settings across devices. This would make the application even more portable for users and allow the sensor system to be truly configured from anywhere. It would also allow the user to experiment with different configurations while preserving the old settings.

5 GANTT CHART

