IEEE Software Requirements Specification Version (1.0)

Original Issue Date: 09/20/2024
Document Number: PAAC-01

Version Number	Author	Date	Reviewed By	Signature
1.0	Karthik Raja	9/22/2024	Ricky Zhao	
1.0	Zachary McPherson	9/22/2024	Karthik Raja	
1.0	Ricky Zhao	9/22/2024	Zachary McPherson	
1.0	Thomas Eckrich	9/22/2024	Jiahao Li	
1.0	Jiahao Li	9/22/2024	Manelin John Rajan	
1.0	Manelin John Rajan	9/22/2024	Thomas Eckrich	

System Requirements Specification 1.1

Final Approval

Name	Date	Signature
Dr. Joseph Profeta	**09/26/2024**	
Mr. Anuj Lee	**09/26/2024**	
Mr. Thomas Gallagher-Teske	**09/26/2024**	

This document represents the current requirements (Version 1.0) of the railway control system. As is normal in the development process, the requirements are not complete on the first writing. The Requirements Analysis phase that follows the SRS will assess the consistency, completeness and feasibility of the requirements. In addition, during the RA phase evaluation of third-party products will be performed to determine their fit and usability. After the RA phase, the SRS will be updated to reflect any necessary changes.

Table of Contents

1. Overview

This Software Requirements Specification (SRS) will detail the process of ensuring a systematic approach to creating a train control system. Clause 1 will provide an introduction and outline of the SRS document. Clause 2 will list the references made to relevant, external documents. Clause 3 will define terms to be used throughout the entirety of this document. Clause 4 will discuss the overall structure of the project, including software and hardware interfaces, users within the system, and the constraints bounding the project. Lastly, Clause 5 will detail the system's specific requirements, including the functional and non-functional requirements, design constraints, and other tertiary requirements.

1.1 Purpose

The purpose of the SRS is to present a detailed description of each requirement necessary for the development of the train transit system that will serve the people of Pittsburgh, facilitating their movement around Allegheny County. This document is created for the stakeholders and the developers of the system as it will be proposed to the Port Authority of Allegheny County (PAAC) for approval upon its completion.

1.2 Scope

This document will outline the specifications of the software requirements for the train transit system project as it includes all the necessary elements of a complete SRS specific to the aforementioned project. The document will discuss the purpose, interfaces, requirements, constraints, users, future additions, and overall functionality of the system. The information entailed in this document will be a useful resource for stakeholders and developers to fully understand the objective of the train transit control project.

2. References

[1] "Flexity 2 Tram Datasheet," Part datasheet, November 2009

[2] IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

[3] J. Profeta, "System and Project Engineering," ECE1140, University of Pittsburgh, [Document], September 2024. Available: https://canvas.pitt.edu/courses/272320/files/folder/Grading%20Rubrics

[4] J. Profeta, "System and Project Engineering," ECE1140, University of Pittsburgh, [PowerPoint slides], September 2024. Available: https://canvas.pitt.edu/courses/272320/files/folder/Lectures

[5] J. Profeta, "System and Project Engineering," ECE1140, University of Pittsburgh, [PowerPoint slides], September 2024. Available: https://canvas.pitt.edu/courses/272320/files/folder/Labs

3. Definitions, Acronyms, and Abbreviations

3.1 CTC – Centralized Traffic Control

A railway signaling system responsible for managing decision for train routing.

3.2 PLC – Programmable Logic Controller

Industrial-grade computers designed to complete tasks that involve the usage of logic functions.

3.3 PAAC – Port Authority of Allegheny County

The transit agency that operates public transportation in the Allegheny County area.

3.4 COTS – Commercial-Off-The-Shelf

Refers to ready-made hardware or software products that are available to the public.

3.5 Suggested

The values are determined by non-vital systems in accordance with the schedule.

3.6 Commanded

The values determined by vital systems in accordance with safety limitations.

3.7 Setpoint

A target value set as input by the user.

3.8 Beacon

A device located at both ends of each train station that transmits static information to the train via the train's antenna.

3.9 Block

A length of track consisting of various components.

3.10 Vital

Refers to the systems responsible for maintaining the safety of the system.

3.11 Shall

Expresses certainty about an action that will occur in the future.

3.12 Train Yard

Refers to the origin point where trains begin and end their route. It stores all the trains that run on the railway system.

4. Overall Description

4.1 Product Perspective

The overall product perspective was described in this document's introduction. We have is that it will demonstrate a complete train transit system consisting of hardware and software module.

4.1.1 System Interfaces

The various system interfaces in this project are the Central Traffic Control (CTC), Wayside Controller, Track Model, Train Model, and Train Controller. They function as follows:

4.1.1.1 CTC

The CTC is where the Dispatcher is responsible for routing trains along the railway system through determining the suggested speed and authority in accordance with the schedule. Moreover, they determine the failure status of the track and perform maintenance when necessary.

4.1.1.2 Wayside Controller

A Programmer is responsible for the Wayside Controller where vital information such as commanded speed/authority, crossing, and switch/signal commands are calculated or processed. The Wayside controller will pass along information from the CTC to the track model and vice versa.

4.1.1.3 Track Model

The Track Model is a software representation of the physical railway and relays information between the Wayside Controller and the physical train. The physical track is comprised of various components that enable signals to be transmitted through different mediums.

4.1.1.4 Train Model

The Train Model is a software representation of the physical train and relays information between the track model and the train controller. The train model is intended to mimic the critical functionality of a train as demonstrated in a real-world environment.

4.1.1.5 Train Controller

The Train Controller is where vital information is processed, calculated and then commanded to the train model so that it functions as intended. The Driver of the train is responsible for overseeing these operations whereas the train Engineer is responsible for determining the initial conditions for each train.

4.1.2 User Interfaces

Each system component requires their own user interface in which their designated users can interact with. As per the system requirements, each of the user interfaces shall be legible and easily accessible. They shall include separate areas for different information relevant to the various users and dynamically display information as they are altered.

4.1.2.1 CTC

4.1.2.1.1 Train Scheduling

The user interface shall enable the dispatcher to upload a train schedule via a data file which details the arrival and departure times for each station.

4.1.2.1.2 Triggers Maintenance Mode

The user interface shall provide a platform for the dispatcher to view blocks in failure, and to toggle blocks in and out of maintenance mode

4.1.2.1.3 Simulation

The user interface shall provide a method for the user to control the speed of the simulation via a slider.

4.1.2.2 Wayside Controller

4.1.2.2.1 PLC Upload

The user interface shall enable the Programmer to upload a PLC program to the system which handles the logical processes in the controller.

4.1.2.2.2 Controls

The user interface shall house areas to view the information about each block in the system as well as the individual switches, railway crossings, and traffic signals. The user shall be able to comfortably navigate this interface with 1-2 hours of training.

4.1.2.3 Track Model

4.1.2.3.1 Track Layout

The user interface shall provide an area for the Track Builder to import a track layout via a data file, providing the necessary information regarding the track model.

4.1.2.3.2 Visual Representation

The user interface shall provide a visual representation of the track model which allows the user to select individual blocks to view related information, track current block occupancies, zoom in and out for different perspectives, view switch orientations, view crossing statuses and traffic signals, and recognize failures. This user interface is designed in a comprehensive manner – suggesting that the Track Builder can comfortably utilize this portion of the user interface after 1-2 hours of training.

4.1.2.3.3 Triggering Failures

The user interface shall provide a method for Murphy, the secondary user of the track model, to trigger failure modes on selected blocks. This user interface is designed in a simple, minimalistic manner that leaves no margin for user input error. Thus, Murphy shall be able to trigger failure modes with no prior experience.

4.1.2.4 Train Model

4.1.2.4.1 Necessary Information

The user interface shall provide Passengers with information regarding the train's specifications, statuses reflecting operational components, and relevant media. The state of operation, states of the external and internal lights, operation of the doors, and current properties of the train shall be clearly displayed on the interface.

4.1.2.4.2 Emergency Brake

The user interface shall include a separate emergency brake for the passengers of the train to activate when necessary. This feature will be implemented as a large and visually captivating button. This is the only function that the Passenger can control. Thus, the Passenger shall be able to utilize this interface without prior training.

4.1.2.4.3 Triggering Failures

The user interface shall provide a method for Murphy, the secondary user of the train model, to set components of the train to fail. This user interface is designed in a simple, minimalistic manner that leaves no margin for user input error. Thus, Murphy shall be able to trigger failure modes with no prior experience.

4.1.2.5 Train Controller

4.1.2.5.1 Train Operation

The user interface shall enable the Driver of the train model to activate brakes, input the setpoint speed, command the power output, and operate various physical components of the train. Other autonomous functionalities shall be handled within the train controller, as well. The train Driver shall be able to command the train through this user interface after 1-2 hours of training.

4.1.2.5.2 Relevant Information

The user interface shall display important information regarding the train's operation – namely, the commanded speed, authority, and failure statuses within the train.

4.1.2.5.3 Critical Inputs

The user interface shall provide a method for the Engineer to assign individual Kp and Ki values for separate trains as they enter the system. The minimal design of the user interface leaves little room for user input error; thus, the Engineer shall be able to set initial conditions to the system with less than an hour of training.

4.1.3 Hardware Interfaces

For the hardware component of this system, a Raspberry Pi will be utilized to display the critical output of the Train Controller, the power command, on an LCD display. This display will be updated properly to reflect the changes made to the power command in the Train Controller.

4.1.4 Software Interfaces

The software product that is used across all the interfaces is Python version 3.12.6 which can be accessed from their website, https://python.org/. Python is one the most popular programming languages worldwide due to its comprehensive nature; therefore, in a collaborative project such as this, Python is ideal. The documentation defining this interface can be found at https://docs.python.org/3/.

4.2 Product Functions

4.2.1 Manage Routing of Trains

The train transit system functions through the creation and management of train routes, ensuring trains can reach their intended destinations while avoiding collisions and maintaining schedules.

4.2.2 Determines Failures

4.2.2.1 Track Failures

The track model has three modes of failure:

- Broken Rail Failure – This refers to when the rails physically break in a manner that the train wheels can no longer roll over them.
- Track Circuit Failure – This refers to when there exists a short in the track circuit caused by an external presence or internal malfunction.
- Power Failure – This refers to when the power source for the track circuit malfunctions and power can no longer be provided to the track.

4.2.2.2 Train Failures

The track model has three modes of failure:

- Train Engine Failure – This refers to when the engine cannot produce enough power to meet the required speed.
- Signal Pickup Failure – This refers to when the train fails to receive the 10 baud per second signal from the track via its antennae.

- Brake Failure – This refers to when the service brakes fail to activate.

4.2.3 Controls Train Speed

Vital values are calculated and used within the system to control train speeds in accordance with safety limitations and time constraints.

4.2.4 Operates Track Components

The railway system houses various operational components that control the orientation of track switches, railway crossing gates, and signal statuses along the track.

4.2.5 Operates Train Components

The train operation includes functional components – namely the lighting, braking temperature adjustment, door operation, engine output, and announcements.

4.3 User Characteristics

The users of each modules have different levels of education, experience, and technical expertise; thus, each user interface is designed to cater towards the individual needs of the user.

4.3.1 Dispatcher

As the user of the CTC, the dispatcher's highest level of education is a high school diploma. They possess a basic level of proficiency with technology. Some familiarity with train transit systems is necessary to succeed in this position.

4.3.2 Programmer

As the user of the Wayside Controller, the Programmer has a bachelor's degree in electrical engineering. They are proficient with technology, and they understand multiple programming languages.

4.3.3 Track Builder

As one of the users of the track model, the track builder shall have prior experience in civil engineering. Consequently, they have a bachelor's degree in civil engineering and a basic level of proficiency with technology.

4.3.4 Passenger

To represent the average train passenger, the user of the train model, they have a bachelor's degree in criminal justice; however, they are not very proficient with technology.

4.3.5 Murphy

As a user of the track and train models, Murphy is notorious for testing the failure responses of each model. As a troublemaker, Murphy's highest level of education is a high school diploma, hence their poor understanding of technology.

4.3.6 Driver

The driver of the train received a certificate from a commercial driver's licensing program which qualifies them to drive commercial vehicles. They also possess a basic level of proficiency with technology and a sufficient understanding of operating vehicles. Some experience in the railway industry and familiarity with train operation is required for train drivers.

4.3.7 Engineer

As another user of the train controller, the Engineer has a bachelor's degree in mechanical engineering with a proficient understanding of technology. A background in mechanical engineering and some experience involving calibrating machines is ideal for this position.

4.4 Constraints

This section will describe the different constraints that the developers were bound by while implementing the train transit system.

4.4.1 Regulatory Policies

The system shall follow the requirements provided by the various users of the product. These requirements include detailed functional and non-functional specifications that must be met to ensure the final project functions as desired.

4.4.2 Interfaces to Other Applications

The system shall be successfully integrated with all of the different interfaces included in this project, namely the CTC, Wayside Controller, Track Model, Train Model, and Train Controller. The changes on one module should successfully affect the other module(s).

4.4.3 Higher-Order Language Requirements

The project must be implemented using high-level programming languages that support external libraries, faster development, and automation. Preferred languages include Python or C++.

4.4.4 Reliability Requirements

The project shall demonstrate a reliable train transit system which constitutes the train being a reliable form of transportation. This suggests that the train shall always be on schedule no matter the situation. For example, the track shall be able to handle multiple trains on the railway system by properly routing trains and controlling the track. Moreover, the system should handle failures appropriately and in a timely manner. No matter the configuration of the track, the train shall handle various cases and function as intended, regardless.

4.4.5 Criticality of the Application

The transit system should be focused on prioritizing the passengers' safety and the condition of the train. Any failures that occur on the train put the passengers, as well as the train, at risk of injury.

4.4.6 Safety and Security Considerations

The system should be implemented in a manner to prevent cyber-attacks that might compromise the network. The system should implement an access control mechanism as well as encryption of its data to prevent it from being hacked.

4.5 Assumptions and Dependencies

The SRS reflects the known design requirements. Due to the design process and the communication process between the system designers and shareholders, the system's specific requirements may change over time. The SRS should be written as completely as possible with the most updated information available. Changes to the requirements should be documented within the SRS to provide an audit trail. This should allow both current and past versions of design requirements to be reviewed, and the evolution of the document to be tracked.

4.6 Apportioning of Requirements

The train system is continuously changing implying that requirements may change accordingly. Considering this factor, some potential requirements for future iterations of the project include:

4.6.1 Additional Hardware Requirements

As technological innovations occur in the future, new types of sensors or communication devices may be invented, changing the way information is transmitted and translated through each component.

4.6.2 Additional Track Requirements

The track shall be configured and adapted for different terrains and environments. This transit system may be adopted by different cities which will require the consideration of different constraints specific to the area.

4.6.3 Additional UI Requirements

With the addition of new features and constraints, additional buttons and tables may be necessary. Moreover, accessibility standards may change, requiring the user interface to change in compliance with the revisions.

5. Specific Requirements

This section will discuss the requirements of each module in the entire system.

5.1 Functional Requirements

Each module in the train system shall have a function behind it. The CTC and Train Controller UIs shall have an automatic mode with preset scenarios to demo the system. The system shall be capable of running at least 10 times faster than wall clock time and shall be able to pause the simulation. The final project shall use at least one or more architectural and design patterns covered during the term. These shall be identified in the architecture and design documentation. Any commercial-off-the-shelf components used in the system shall be identified. The vital aspects of the system and how it affects the architecture and design of the system shall be described.

5.1.1 CTC Requirements

The CTC is responsible for the high-level goals of a train control system.

5.1.1.1 Train Scheduling

The CTC shall maintain a schedule of trains to be dispatched. The information associated with each train shall include a block number for its current location; a departure station and time; a destination station and time; and the train line the train is associated with. The CTC shall manage the distribution of new trains from the train yard to meet the specified schedule.

5.1.1.2 Train Routing

The CTC shall be responsible for pathing trains to their destination and avoiding deadlock states where trains block each other. The trains shall be on time while the track is in working condition.

5.1.1.3 Track Status

The CTC shall monitor and display the current track status. The features of the current track status include the signal, switch, and crossing states of each block, the occupancy values of each block, and the maintenance values of each block.

5.1.1.4 Track Failure Detection

The CTC shall monitor block occupancies and train locations to determine where track failures have occurred.

5.1.1.5 Track Maintenance Management

The CTC shall enforce the maintenance of blocks. The system shall route trains around blocks that are currently under maintenance.

5.1.1.6 Suggested Speed

The CTC shall calculate a suggested speed for each train in accordance with the schedule and speed limits, to be communicated through the wayside controller.

5.1.1.8 Suggested Authority

The CTC shall calculate a suggested authority (distance to destination) value for each train. This value will be communicated through the wayside controller.

5.1.1.9 Throughput

The CTC shall calculate the throughput of the system in units of trains per line per hour. This value shall be communicated to the dispatcher.

5.1.2 Wayside Controller Requirements

The Wayside Controller is a vital system responsible for controlling various components of the Track Model and communicating information between the CTC and Track Model.

5.1.2.1 Vital Decision Making

The Wayside Controller shall make vital changes to the incoming suggested speed and suggested authority from the CTC Office. If any changes are made, it will send the new commanded speed and authority to the Track Model; otherwise, it will send the suggested speed and authority. The Wayside Controller will make these changes based on the block occupancy information it receives from the Track Model.

5.1.2.2 Control Track Switches

The Wayside Controller shall control the track switches on the Track Model. It will output binary commands to the Track Model to toggle the direction of the switches from section to section. It will toggle switches to route trains to their destinations and avoid accidents.

5.1.2.3 Control Traffic Lights

The Wayside Controller shall control the traffic lights on the Track Model. It will output binary commands to the Track Model to change light statuses. It will use the traffic lights to signify trains to either continue on their way, slow down, or come to a full stop in order to avoid accidents.

5.1.2.4 Communicate Block Occupancy

The Wayside Controller shall communicate block occupancies from the Track Model to the CTC Office. The Wayside Controller will receive and output block occupancies in the form of Boolean values (block is occupied or not).

5.1.3 Track Model Requirements

5.1.3.1 Communication

The Track Model serves as a medium of communication for the Wayside Controller and the train model.

5.1.3.1.1 Commanded Speed/Authority

The track model shall communicate commanded speed and authority from the Wayside Controller to the train model via the 10 baud limit.

5.1.3.1.2 Beacon Information

The track model shall communicate static information to the train model via antennae, limited by 128-bytes of data per beacon.

5.1.3.1.3 Light Level Status

The track model shall communicate whether the terrain is underground, triggering the external lights of the train.

5.1.3.1.4 Number of Passengers Boarding/Vacancy

The track model shall keep track of the number of people boarding at a station and shall compare this number with vacancies on the train in order to maintain the safe occupational capacity.

5.1.3.1.5 Block Occupancy

The track model comprises of a set of steel rails, track circuits, and battery components. The track model shall detect presence on each block through the shortening of the track circuit.

5.1.3.1.6 Switch, Signal, Crossing Commands and Statuses

The track model shall receive commands from the Wayside Controller regarding the orientation of switches, railway crossing, and traffic light signals and changes to these inputs shall be reflected on the track model, accordingly.

5.1.3.1.7 Failure Modes

The track model shall detect and report any failures in the model to the Wayside Controller.

5.1.3.2 Visual Representation

The Track Model is a software implementation of the physical railway system and provides a visual representation for the users.

5.1.3.3 Flexibility

The track model shall be stored in a database and shall be configurable. As the user of the track model, the track builder is responsible for importing a track layout to the system and determining its external temperature. This feature allows future changes to the system to be implemented without reconstruction of the component.

5.1.4 Train Model Requirements

As a software implementation of a physical object, the train model accurately simulates a train's movement.

5.1.4.1 Adheres to Newton's Law to Calculate Train Parameters

The train model shall be able to use the different Newton law equations, under the assumption that the train is a rigid body point mass, to find the force, acceleration, and the current velocity of the train. The equations used to find the parameters mentioned above includes $F = ma$, $P = Fv$, and $v = \int a(t)\,dt$.

5.1.4.2 Passenger Brake Command

The train model shall be able to allow the user to activate a passenger emergency brake, which then automatically applies the emergency brake for the train and alerts the driver that the passenger brake command has been pressed.

5.1.4.3 Number of Passengers

The train model shall be able to calculate how many passengers are allowed to board a train when arriving at a station. The train model shall then show the passengers how many more passengers are allowed to board the train. This happens by the train model sending the track model how many open seats there are in the train the track model would then determine how many people are boarding the train, which cannot be more than the open seats in the train.

5.1.4.4 Advertisements

The train model shall be able to display multiple advertisements to the passengers of the train.

5.1.4.5 Current Velocity

The train model shall be able to calculate the current velocity of the train by the power command that is given by the train controller.

5.1.4.6 Displays all information regarding the train

The train model shall be able to display any information that pertains to the train and the passengers inside of the train.

5.1.4.6.1 Current Speed

The current speed that the train is traveling.

5.1.4.6.2 Current Acceleration

The current acceleration of the train, which is calculated in the train model.

5.1.4.6.3 Commanded Speed

Calculated in the Wayside Controller, the commanded speed determines the safe speed for the train to travel at in accordance with the terrain and condition of the track. This is vital information that needs to be relayed to the train to ensure safe operation.

5.1.4.6.4 Commanded Authority

The authority, determined by the CTC, communicates to the train how far it is allowed to travel before it must stop. This is another piece of vital information that must be sent through the track model and into the train model.

5.1.4.6.5 Brake Status

The status of the brake is shown as either on or off. On meaning that the brakes are being applied and off meaning that the brakes are not being applied.

5.1.4.6.6 Power Command

The power command that is given by the train controller is displayed in the train model.

5.1.4.6.7 Train Length

The length of each car attached to the train.

5.1.4.6.8 Train Width

The width of each car attached to the train.

5.1.4.6.9 Train Height

The height of each car attached to the train.

5.1.4.6.10 Weight of Empty Train

The weight of the train when it is empty with no passengers.

5.1.4.6.11 Weight of Train With Passengers

The weight of the train considering all of the passengers' weight.

5.1.4.6.12 Crew Count

The number of staff members that are on the train.

5.1.4.6.13 Passenger Count

The number of passengers that are on the train.

5.1.4.6.14 Left Door Status

The status of the left door is shown in the train model's interface. When the left door is open and passengers are getting out, the status is shown as open, and when the door is closed, then the status is shown as closed.

5.1.4.6.15 Right Door Status

The status of the left door is shown in the train model's interface. When the left door is open and passengers are getting out, the status is shown as open, and when the door is closed, then the status is shown as closed.

5.1.4.6.16 Interior Light Status

The status of the interior lights is shown in the train model's interface, When the interior lights are on, then the status is shown as on, and when the interior lights are off, then the status is shown as off.

5.1.4.6.17 Exterior Light Status

The status of the exterior lights is shown in the train model's interface, When the exterior lights are on, then the status is shown as on, and when the exterior lights are off, then the status is shown as off.

5.1.4.6.18 Train Temperature

The train model shall also display the current temperature of the train, which is calculated in the train model via a first-degree equation.

5.1.4.7 Triggering Failure Modes

The train model shall be able to trigger all three failure modes that affect the train and make it go under maintenance mode. There are three failure modes for the train: brake failure, train engine failure, and signal pickup failure.

5.1.4.8 Physical Characteristics

The train model shall be able to obtain static information from beacons that are right before and right after each station. These beacons hold information such as the name of the next station, the grade of the current block, and the elevation of the current block.

5.1.4.8 Control and Commands

The train model shall be able to display any information that pertains to the train, such as the current velocity, acceleration,

5.1.5 Train Controller Requirements

The train controller is a vital component of the railway system that processes information and commands the train's operation.

5.1.5.1 Setpoint Speed

The train driver shall be able to regulate the speed at which the train can travel in respect to the bounds of the speed limit and commanded speed, which is inputted as the setpoint speed. This input is only required when the train is running in manual mode. When it is in automatic mode, the setpoint speed input shall be disabled, so that the user cannot enter in a desired speed value.

5.1.5.2 Power Command

The train controller shall calculate the power command, which is the calculated power specifying the power to be supplied to the train's motors to achieve the desired speed, with the current velocity that is given by the train model. In our hardware interface, we are displaying the power command output from the train controller to an LCD Display via a raspberry pi.

5.1.5.3 Toggle Train Lights

The train controller shall also be able to toggle the interior and exterior lights at appropriate times. For example, when the train is in a dark environment, such as a tunnel or underground, the interior and exterior lights would have to be turned on. In manual mode, the exterior lights are automatically turned on, but the interior lights have to be manually turned on by the driver.

5.1.5.4 Announcements

The train controller shall provide the passengers of the train with messages at appropriate times, such as station arrivals and when the emergency brake has been pressed by either the passenger or driver. It reaches the driver by outputting the announcement to the train model. The name of the station that the train arrived at is passed on from the beacon, which is an input from the train model to the train controller. There is a beacon right before a station and right after a station. The station name will be passed onto the train controller in the beacon that comes right before the station.

5.1.5.5 Handle Failures

The train controller shall be able to handle three different failure modes: Brake Failure, Train Engine Failure, and Signal Pickup Failure. These failures shall be supervised by the driver at all times to ensure that the train is able to identify the failure and handle it in a safe manner. In the case where a signal pickup failure occurs, the CTC is unable to communicate with the train controller and cannot update the train's current authority. When any of the failures are triggered, the train automatically stops, and the driver would get out of the train and remain stopped until the issue is fixed.

5.1.5.6 Train Brakes

The train driver shall be able to operate the driver's emergency brake and driver's service brake in order to slow down or stop the train at appropriate times. The train controller shall also know when the passenger emergency brake has been triggered. When the passenger brake command is activated,

5.1.5.7 Door Controls

The train controller shall be able to open and close the corresponding doors depending on which side the station is located on. When the train is in manual mode, the doors are opened and closed by the driver, but when the train is in automatic mode, the doors are opened and closed automatically. There is also a separate button for the left door and the right door for the driver.

5.1.5.8 Train Temperature Control

The train driver shall be able to set the desired temperature that is inside of the train. The desired temperature set by the driver is then sent to the train model, where a first order equation is used to determine the current train temperature as the temperature increases gradually. Each iteration of the temperature increase is then sent to the train controller for a live value for the current temperature.

5.1.5.9 PI Controller Gain Values

The train engineer shall be able to set the Kp (Proportional Gain) and Ki (Integral Gain) values corresponding to each train that is leaving from the train yard. These two values help make the train stable and safe for the passengers. These two values are used to calculate the power command in the train controller using the formula given to us and output that power to the train model. As stated above, the train model would then use that power and find the current speed of the train.

5.1.5.10 Display Current State of Train

The train controller shall display critical values to the driver, such as the current speed, commanded speed, commanded authority, failure modes, actual train temperature, and passenger brake command signals, which are all inputs to train controller from the train model.

5.2 Non-Functional Requirements

The non-functional requirements refer to requirements regarding the software aspect and the human interaction which contribute a crucial role in ensuring the entire system operates properly and efficiently.

The railway system shall be executable on the Windows 10 operating system.

5.2.1 Compatibility Requirements

5.2.2 Vital Interface Requirements

The two modules in the architecture that are considered vital are the Track Controller and the Train Controller. These are the only two modules where safety measures can be taken. For example, these are the only two modules where the speed and authority can be changed, since they are the two commands that are risking the safety of the system.

5.2.3 System Requirements

Each module shall have a user interface and shall be submitted as an installable executable and the whole system shall be submitted as a runnable executable.

5.2.4 Performance

5.2.4.1 Simulation Efficiency

The system shall enable the developers to efficiently run the simulation 10 times faster than wall-clock time.

5.2.5 Security

5.2.5.1 Username and Password

The system shall not use a username and password.

5.2.5.2 Data Storage

Data shall be contained within the system.

5.2.6 Reliability

5.2.6.1 Failure Handling

The system shall handle failures in a graceful manner.

5.2.6.2 Integrity of Communications

The communication between modules shall not result in a crash or loss of valuable information.

5.2.7 Availability

The system shall be available to all the related stockholders and developers to view and use.

5.2.8 Maintainability

5.2.8.1 Versioning

All components shall have a version number. A version compatibility policy shall be created, and all modules shall adhere to that policy.

5.2.8.2 Compatibility

The compatibility between modules shall be maintained by providing a conversion utility to move from one version to another.

5.2.8.3 Runtime Maintenance

The system shall be easily upgradable and installable and configuration files shall be capable of being re-used.

5.2.9 Portability

The system is not intended to be used outside of transit systems.

5.3 Design Constraints

The software design is limited by the physical infrastructure of the North Shore Extension and the education of the users.

5.3.1 Physical Infrastructure

Since the railway operates on DC signals, the communication from the CTC Office to the Wayside Controller is done through Boolean logic. Once information has exited the Wayside Controller and enters the track, it can only be transmitted through a 10 baud (bit) per second channel on the rails which is one-way only, meaning the trains cannot send information back to the CTC Office. Additionally, the beacons placed at the start and end of each station are limited to 128 bytes of static information.

5.3.2 Software Constraints

Since the users of the system come from various educational backgrounds, specific user interfaces must be tailored to someone with no degree of higher education. Furthermore, none of the hardware hosting the user interfaces have any kind of file management system, so the software must be able to detect files on a local drive.

5.4 Other Requirements

5.4.1 Submission

All group members must submit all assignments by the due date.

5.4.2 Teamwork

Throughout the semester, groups should work together and try to divide and conquer the workload.

5.4.3 Know Your User

The most important consideration is to know the users of the system. The best systems are catered around the people who use them; therefore, it is essential to understand what capabilities the users of the system have, and the workflows that they would like to use to complete their tasks. Developers must have constant communication with the stakeholders to fully understand the requirements and make sure none of the requirements change in the change of developing.

5.4.4 UI Demonstration Layout

During the final demonstration, the UIs of each module should not take up the entire computer screen, instead, they should provide sufficient space for the other UIs to be displayed as well.