**Software Requirements Specification**

**for**

**Autonomous Train Control System**

**(Version 1)**

**Prepared by Orange Juice and the Cowboy Hats**

Table of Contents

[**1. Introduction 3**](#_hf9c7yhd1x7x)

[1.1 Purpose 3](#_pwluy88xh3r4)

[1.2 Scope 3](#_gfewyqkmrhe0)

[1.3 Definitions, Acronyms, & Abbreviations 3](#_u3jzgad2plyv)

[1.4 References 4](#_2m3me1phiqxs)

[1.5 Overview 4](#_jtxq97537qt6)

[**2. Overall Description 5**](#_cw8dv06jbxp9)

[2.1 Product Perspective 5](#_4ytdya26vgdx)

[2.2 Product Functions 5](#_c4g12zmsg0lh)

[2.3 User Characteristics 6](#_pv9ljdc2d8dt)

[2.4 Assumptions & Dependencies 10](#_wp4gzry1day9)

[2.5 Apportioning of Requirements 10](#_jrph1sliz76h)

[**3. Specific Requirements 11**](#_nujahq9hlurw)

[3.1 Specific Requirements 11](#_aevs616vr7wy)

[3.2 Functional Requirements 14](#_f3b95icwjxpp)

[3.3 Non-Functional Requirements 15](#_ms73bpm6meiz)

[3.3.1 Performance 15](#_9y7ekc7jqw03)

[3.3.2 Reliability 16](#_o7t1v9degr02)

[3.3.3 Availability 17](#_566aolpxlj2p)

[3.3.4 Safety 18](#_vqrbgjy7pty0)

[3.3.5 Security 19](#_szb9nfjaa9sa)

[3.3.6 Maintainability 19](#_2nwthka995tn)

[3.3.7 Portability 20](#_gg6c6kksxy9u)

[3.4 Design Constraints 20](#_9jtktj54nr1c)

[3.5 Other Requirements 20](#_v64fwgk8eam8)

# 

# 1. Introduction

## 1.1 Purpose

This document will detail the software requirements for the train control system. It will cover the overall description of the final product, as well as list the specific functional and non-functional requirements.

This document will serve as a guide for the engineers creating the project. It will also allow for clarification of requirements with the users/stakeholder.

## 1.2 Scope

The scope of this Software Requirements Specifications (SRS) outlines the Train Control System (TCS), a comprehensive solution designed to manage and control train operations within railway networks. The TCS consists of various interconnected modules, including the Centralized Traffic Control (CTC), Track Controller (Wayside Controller), Track Model, Train Model, and Train Controller. The main objective of the TCS is to ensure safety of the passengers, and the efficient and reliable movement of trains as well as making sure to address the needs and expectations of the users and stakeholders while ensuring compliance with relevant regulations and standards. Key users include the dispatcher, track builder, Murphy, passenger, train driver, train engineer, and PLC programmer.

## 1.3 Definitions, Acronyms, & Abbreviations

1. SRS: Software Requirements Specifications
2. TCS: Train Control System
3. Vital: Processes that are critical for the safe operation of the train control system
4. CTC: Centralized Traffic Control
5. PLC: Programmable Logic Controllers
6. UI: User Interface
7. GUI: Graphical User Interface
8. COTS: Commercial-off-the-shelf
9. Track Control is synonymous with Wayside Controller
10. The main purpose of the user Murphy is to sabotage and break train/track.
11. KI: Integral Gain
12. KP: Proportional Gain
13. CSV File: a plain text file format where data values are separated by commas, commonly used for storing tabular data.

## 1.4 References

*There are no references at this time*

## 1.5 Overview

This SRS is broken into two major sections: Overall Description and Specific Requirements.

The Overall Description section will discuss the train control system as a whole, specifying the major functions and components of the system and how the software will operate at a higher level. It will also discuss the users and outside constraints/requirements of the system. To clearly discuss these things, the Overall Description section will be divided into five subsections: Product Perspective, Product Functions, User Characteristics, Assumptions & Dependencies, and Apportioning of Requirements.

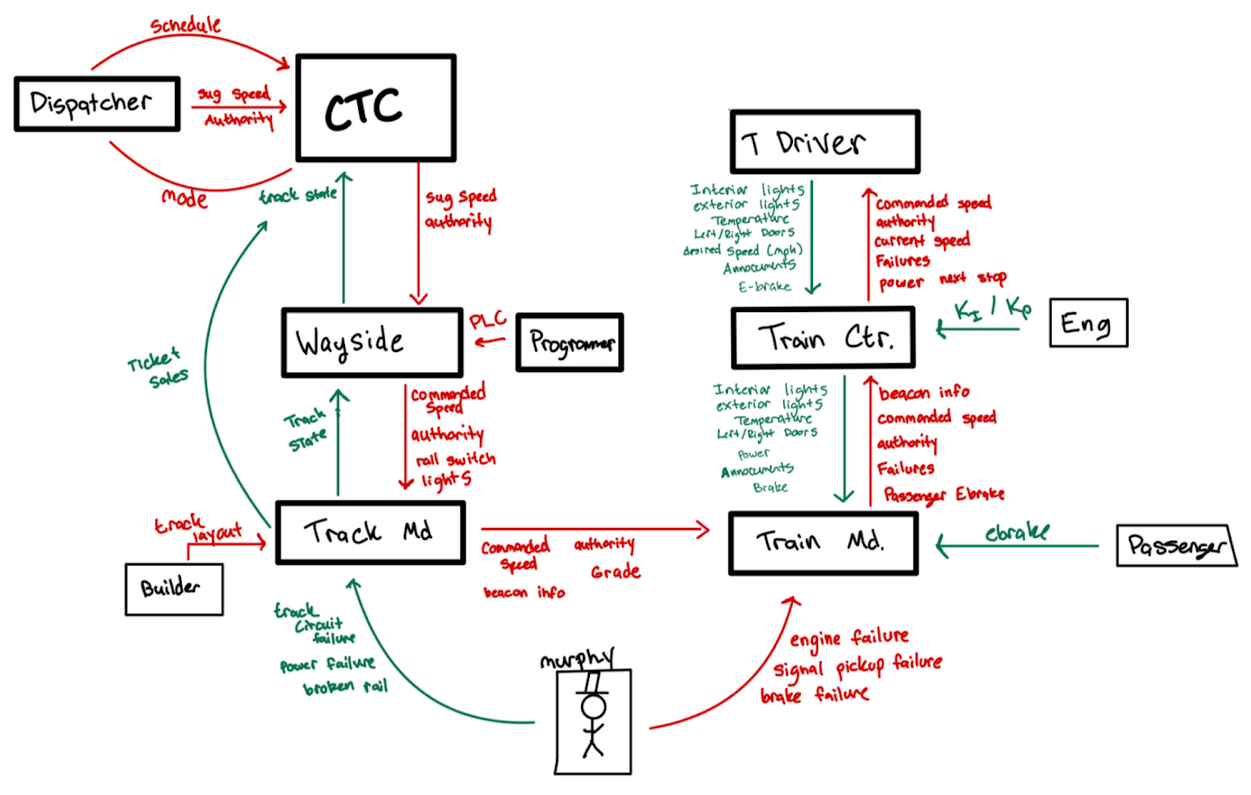
The Specific Requirements section will deep dive into the specific functions of the system, both function and non-functional. It will list all the intended functions of the system clearly, in precise and easily testable statements. The Specific Requirements Section will also be divided into five subsections: Specific Requirements, Functional Requirements, Non-Functional Requirements, Design Constraints, and Other Requirements.

# 

# 2. Overall Description

## 2.1 Product Perspective

This product is a Train Control System (TCS) that will successfully transport passengers to the correct destination within a timely manner. Each of the previously specified modules will be integrated and work cooperatively to perform said task.



The software follows the same format and operation as many existing passenger train models. Each user interface within the TCS has a similar format to these existing systems.

## 2.2 Product Functions

The software will be able to safely and autonomously guide trains along the red and green lines from an uploaded schedule.

Features include:

1. Dispatching and returning trains to the yard
2. Following a given schedule imputed to CTC
3. Controlling all rail switches and crossbars via PLC
4. Controlling rail occupancy lights
5. Accelerate and decelerate the train
6. Monitor and act accordingly for failures such as:
   1. Power failure
   2. Broken rails
   3. Track circuit failure
   4. Engine failure
   5. Signal pickup failure
   6. Brake failure
7. Open and close doors at appropriate
8. Control Interior Features of the Train
9. Report ticket sales to CTC
10. Advertise to passengers
11. Train and Track Models to test the system in simulation

## 2.3 User Characteristics

1. **Dispatcher:** Susan Johnson

**Age:** 44

**Education:** Bachelor’s Degree (Business)

**Salary Range:** $66,481 - $89,623

**Description:** Susan is a seasoned train dispatcher with a wealth of experience. Her journey in this field began after completing her Bachelor's degree. She plays a crucial role in coordinating and managing train schedules to ensure smooth operations. Susan is known for her sharp organizational skills and ability to handle the complexities of coordinating train movements efficiently.

**Technical Comfort Level:** Susan is adept at working with computers, a necessary skill for her role as a train dispatcher. Her technical proficiency allows her to navigate the software and systems crucial for managing train schedules and ensuring the safety and efficiency of train operations.

**Hobbies:** Susan’s hobbies are reading, hiking and gardening. She looks forward to doing the NYT Wordle, Mini-Crossword and Connections everyday.

**Goals:** Susan’s goals are to prioritize safety, preventing accidents and responding swiftly to emergencies. She would like to optimize train schedules for efficiency and minimize delays. She aims to provide reliable, timely services, and stay adaptable to technological advancements.

1. **Track Builder:** Tom Miller

**Age:** 29

**Education:** High School Diploma/GED

**Salary Range:** $45,000 - $60,000

**Description:** Born and raised in a blue-collar neighborhood of Pittsburgh, Tom has always been fascinated by the vast network of railways that crisscross America. From a young age, he was drawn to the raw power and intricate engineering of trains. Tom is a rugged, hands-on worker with a deep respect for the history and importance of the railroad industry. He's known for his strong work ethic, practical skills, and a no-nonsense attitude. Tom's life has been shaped by the railroads, not just professionally but also personally, as it's where he’s met his closest friends and where he finds his greatest sense of purpose.

**Technology Comfort Level:** Moderate. Tom is comfortable with the technology required for his job, including machinery used in track laying and maintenance. However, he prefers physical labor and hands-on work over computer-based tasks.

**Hobbies:** Model railroading, hiking in the Pennsylvanian wilderness, and working on his vintage car collection. He also enjoys local history and often volunteers for community projects, especially those related to preserving Pittsburgh's industrial heritage.

**Goals:** Becoming a supervisor or manager in the railroad construction sector.

1. **Murphy:** Lucas S. Malone

**Age:** 35

**Occupation:** Cargo Inspector

**Education:** High School Diploma/GED

**Salary Range:** $64,000 - $79,800

**Description:** Lucas, a skilled cargo inspector in the rail industry. Despite his initial commitment to upholding industry standards, he orchestrated the theft of valuable cargo and disrupted train operations. Lucas's illicit activities were eventually exposed, highlighting the importance of maintaining trust and integrity in the field. His journey into this deceptive path began when he initially signed up for the job, concealing his true intentions with a veneer of professionalism.

**Technology Comfort Level:** Lucas grapples with technology, relying on a dated computer and a basic cell phone for work. While slow to adopt new advancements, he perseveres through the challenges, demonstrating a persistent effort to navigate the digital landscape despite his inherent discomfort with modern tech tools.

**Hobbies:** Underground poker, graffiti art, playing board games and puzzles, and fishing

**Goals:** Lucas has many goals, including professional advancement in cargo inspection, financial independence through illicit activities, personal fulfillment through hobbies, avoiding detection for criminal pursuits, building a positive reputation as a respected cargo inspector.

1. **Passenger:** Brad Jones

**Age:** 27

**Occupation:** Business Analyst

**Education:** Master’s Degree (Business Analytics)

**Salary Range:** $100,000 - $200,000

**Description:** Brad is a Pittsburgh native who loves nothing more than to spend a weekend watching the Pittsburgh Pirates play at PNC park. He lives Downtown, so he loves how close the stadium is. Brad is single and lives alone. He enjoys Pittsburgh because of its accessibility and great sports teams.

**Technical Comfort Level:** Brad is really comfortable with modern technology. Though, he does want to put the effort in to learn something new, as he's lived in Pittsburgh his whole life.

**Hobbies:** Getting drinks with his friends, watching the Stillers, and golfing

**Goals:** He wants the experience to be so easy that it is almost thoughtless. Because Brad goes to games a lot, he doesn't want a lot of hassle with boarding/deboarding the train, and cares about the train being on time.

1. **Train Driver:** Frank Smith

**Age:**  37

**Education:** High School Diploma/GED

**Salary Range:** $64,210 - $77,000  
**Description:** Frank has been driving trains for 13 years and went to school at the Pennsylvania College of Technology for automotive technology. From there, he found a newspaper advertisement for train drivers and decided to apply. Frank has two kids, ages two and five.

**Technical Comfort Level:** Frank owns an iPhone and a personal computer. Through his work as a Train Driver, he has gotten used to working with technology. It takes him a while to learn new technology, but he's pretty savvy once he is familiar with it.

**Hobbies:** Golfing, hanging out with my young children, playing with my cats

**Goals:** Frank has many user-friendly goals related to his job. Some of these include the brake being very bright and distinguishable from other mechanisms, current speed being displayed very large, clear reporting of any faults that are happening, and a touchscreen for non-vital procedures (cabin control and doors, etc.)

1. **Train Engineer:** Gregory Mcaugh

**Age:** 45

**Education:** Bachelor’s Degree (Mechanical Engineering) Salary Range: $80,000 - $100,000

**Description**: Gregory has loved trains since he was a kid, and decided that after college, he would pursue his passion of making sure trains were functioning correctly. He grew up in the Washington area but came to Pittsburgh to follow his wife's new job. He is deeply disappointed with the lack of trains in Pittsburgh.

**Technical Comfort Level**: While Frank is comfortable with the technology used on trains due to his job, he has a hard time learning other newer technology and sometimes struggles to navigate his iPhone. He is more adapted than most people his age to modern technology, but doesn't like new change.

**Hobbies:** Watching Sunday football, Sudoku, walking his dog

**Goals:** Easy understandable input for KI and KP and can very clearly see the outputted affect on the power of the train

1. **PLC Programmer:** Jason B. Morris

**Age:** 31

**Education:** Bachelor’s Degree (Computer Engineering)

**Salary Range:** $98,600 - $133,100

**Description:** As told by Jason’s age, he does not have as much industry experience as some of the other users of the system. He worked for one other company as a programmer prior to joining this project, and has currently been employed here for ~3 years. Although Jason is not as seasoned as his colleagues, he earned his undergraduate degree from a rigorous institution and had exceptional grades. He is known for his excellent problem-solving abilities, as well as his dedication and persistence in regards to solving difficult problems.

**Technology Comfort Level:** Jason is quick to adapt to new technology, as implied by his area of study. He is interested in and familiar with most of the latest technology.

**Hobbies:** Jason enjoys creating small, personal projects related to electronics. He also enjoys playing pick-up soccer with his friends, as well as snowboarding during the Pennsylvania winter.

**Goals:** Jason’s goal as the PLC programmer is to allow the wayside system to run in automatic mode (autonomously) as smoothly as possible, allowing passengers to arrive at the correct destination quickly without any rail-occupancy conflicts. He wants it to be easy to upload and run the PLC file, save and re-use previous PLC files, and wants the interface to be simple. He also would like some clear distinction as to what mode the wayside system is in (manual vs. automatic), as well as clear stating of which wayside is currently being viewed.

## 2.4 Assumptions & Dependencies

Ability to Integrate

* The functionality of the software and build of the architecture may change based on the modules ability to integrate with one another
* Failure to integrate may require modules to create other internal dependencies in order to function properly

Change in Customer Requirements

* As we consult with the customer, it's important to remember that customers may change their mind about the original design of the system and may require the SRS to change appropriately
* In addition, as we interview the user, our understanding of the systems may change and require appropriate pivots of software implementation

Input Dependencies

* As stated in this SRS, we have very specific layouts on how users will input information into the system and how this will affect the flow of information. However, if design changes require us to change the way users input information, it will require us to adjust the SRS

Murphy’s Law

* There will be changes to the system that we as designers may not be able to predict.Thus, forcing us to adjust documentation. We assume under Murphy's Law that any chaos situation could affect this project

## 2.5 Apportioning of Requirements

Change in Track Layout

* An altered track layout is a possibility. We will make sure that the track is easily adjustable and not “hard-coded” if any issue arises in the layout.

Change in Train Functions

* In case certain train functions need altered, such as the brake operation being only one constant value opposed to a gradient of choices, we will make sure we generalize our code to not lock us into any of these possible configurations.

Change in PLC Files

* In case characteristics of the PLC file inserted by the programmer are changed, we are prepared by having generalized code to parse this.This is currently expected to come as a .csv file. If this is changed however, we will ensure parsing the information is still possible.

# 3. Specific Requirements

## 3.1 Specific Requirements

**CTC**

1. The dispatcher shall be able to choose between manual or automatic mode of dispatching trains
2. In manual mode, dispatcher shall be able to manually input information to dispatch trains one at a time
3. Dispatched trains shall be added to the schedule
4. In automatic mode, dispatcher shall be able to upload an excel sheet of the schedule and have it populate the train schedule on the UI
5. The system shall display the current schedule to the dispatcher
6. Dispatched trains shall be given the correct authority based on starting stations and destinations
7. Currently dispatched trains shall be given a suggested speed based on their location on the track
8. The dispatcher shall be able to close track blocks/sections down for maintenance.
9. The dispatcher shall be able to modify switch position while for maintenance
10. The CTC system shall calculate and display throughput metrics for all lines in the system

**Track Controller**

1. The Programmer shall be able to insert a PLC file of purely boolean values.
   1. The PLC file shall be interpreted by a chosen encoding for the values it stores.
   2. In automatic mode, these values shall determine things like train crossings, switch positions, and lights.
   3. In manual mode, those values shall be overwritten.
2. In manual and automatic mode, the user should be able to have a general overview of everything going on in any given wayside area.
   1. A line is chosen first.
   2. A wayside is then chosen. Once chosen, a static map of the area the wayside covers is shown. This map will also have the id’s of the following displayed: switches, rails, crossings, and wayside.
   3. In automatic mode, nothing else can be altered.
   4. In manual mode, the user can select a block that occupies a switch, railroad, or crossing to change the current behavior.
      1. No matter what, dangerous conditions can never be created, even in manual mode. If a dangerous choice is chosen, the system will simply ignore it.

**Track Model**

1. The Track Builder and Murphy shall be able to select a specific block.
   1. Once a block is chosen, block data shall be presented to the users.
   2. Light and switch status shall also be presented to the user.
2. The users shall be able to indicate block failures.
   1. If the broken rail switch is on, a failure signal shall be sent to the Wayside Controller system.
   2. If the track circuit failure switch is on, a failure signal shall be sent to the Wayside Controller system.
   3. If the power failure switch is on, a failure signal shall be sent to the Wayside Controller system.
3. The user Murphy shall be able to upload a track layout file.
   1. Once a file is uploaded, live block data shall be presented to the user.

**Train Model**

1. The train shall have acceleration and velocity limits keeping account of terrain
2. The train model shall display length, height, width, mass, crew count, and passenger count.
3. Murphy shall be able to deliver train engine failure, signal pickup failure, brake failure.
4. Passengers should be able to pull the emergency stop brake.

**Train Controller**

1. The Train Driver shall be able to select two modes of operation: Manual or Automatic
   1. Automatic Mode
      1. In Automatic Mode, the Train Controller system shall determine appropriate speed the train should be traveling via authority,commanded speed, and speed limit.
2. Thus, the Train Controller system shall determine appropriate power output to Train Model
   1. In Automatic Mode, the Train Controller system shall utilize authority to determine appropriate braking operations for the train
   2. In Automatic Mode, the Train Controller system shall announce train stations as they approach and open door appropriately
   3. In Automatic Mode, the Train Controller system shall use default values for KI and KP
   4. In Automatic Mode, the Train Controller system shall set interior/exterior light states appropriately
   5. Manual Mode
      1. In Manual Mode, The Train Controller system shall take input for the driver for both power output and brake input
         1. The Train Controller shall monitor the inputs from the Train Driver to determine appropriate brake and power outputs to the Train Model
         2. The Train Controller shall not allow the driver to put the Train in hazardous conditions
            1. The Train Controller shall not allow the train to exceed the speed limit
            2. The Train Controller shall not allow the train to be driving past its Authority Designation
            3. The Train Controller shall not allow the train to apply brakes and increase power output at the same time
      2. In Manual Mode, The Train Controller shall allow inputs for announcements from the Train Driver
      3. In Manual Mode, The Train Controller shall allow inputs for interior/exterior lights setting from the Train Driver
      4. In Manual Mode, The Train Controller shall allow inputs for opening the left/right doors from the driver
      5. In Manual Mode, The Train Controller shall allow inputs for KI and KP metrics from the Train Engineer
         1. The Train Controller shall monitor KI and KP inputs and evaluate for the creation of unsafe conditions
         2. The Train Controller shall not pass KI and KP metrics from the Train Engineer if they create unsafe conditions
            1. The Train Controller shall pass default KI and KP values in this case
3. The Train Controller shall have an emergency braking
   * 1. The Train Controller shall have an emergency ‘service’ brake input from the driver
     2. The Train Controller shall take input if a passenger pulls the emergency brake
     3. The Train Controller shall turn power output to Train Model to 0 in the case of an emergency brake
     4. The Train Controller shall calculate safe braking output to Train Model in case of an emergency braking
4. The Train Controller at all costs shall choose operations that keep the train and its passengers safe

## 

## 3.2 Functional Requirements

**CTC**

1. The dispatcher shall be able to dispatch trains and have them added to the schedule, in manual or automation mode
2. The current schedule shall always be displayed
3. Authority and suggested speed shall be given to each train based on the schedule

**Track Controller**

1. The track controller shall be able to operate lights, railroad crossings, and switches while in manual mode.
2. The track controller shall be able to take a PLC file as an input.
3. The track controller shall be able to interpret this PLC file and set initial conditions of the track based on the PLC file for the automatic mode.
4. Dangerous conditions shall not be allowed even if the system is currently in manual mode.

**Track Model**

1. The track builder shall be responsible for keeping live updates of the block data with/without the user uploading the track layout:
   1. Grade
   2. Elevation
   3. Length
   4. Speed Limit
   5. Direction of Travel
   6. Railway Crossing
   7. Track heaters
   8. Becon
   9. Ticket Sales
   10. Number of Passengers Boarding
2. The track builder shall be responsible for keeping the live status of the lights and switches of the track.
3. Murphy shall be able to deliver the track circuit failure, power failure, broken rail.

**Train Model**

1. The train shall be able to take in a power command input and calculate a velocity output.
2. Murphy shall be able to deliver train engine failure, signal pickup failure, brake failure. Passengers should be able to pull the emergency stop brake.

**Train Controller**

1. The train controller shall regulate the speed of the train to the setpoint while not exceeding the speed limit or authority allowed by the system, regardless of mode.
2. The train controller shall use the track signal as input and decode the information to determine speed limit and authority.
3. The train controller shall take as input the command setpoint from a Transit Operator.
4. The train controller shall open and close doors at appropriate times.
5. The train controller shall turn on and off lights at the appropriate times.
6. The train controller shall announce stations and stops at the appropriate times.

## 3.3 Non-Functional Requirements

### 3.3.1 Performance

**CTC**

1. The CTC shall have one terminal
2. The CTC shall support one dispatcher user at a time
3. The CTC shall be able to take in and handle the block occupancies of the entire system
4. The CTC shall be able to take in and handle the ticket sales from each station in the system
5. The system shall be able to load and run one schedule at a time

**Track Controller**

1. Wayside controller shall support one terminal
   1. Multiple wayside points shall be selected within the terminal, specifying where the change to the track can be made if in manual operation
2. Wayside controller shall only support one user at a time (PLC programmer)
3. Wayside controller shall receive the suggested speed (Km/hr)
4. Wayside controller shall receive train authority (m)
5. Wayside controller shall receive the contents of a single PLC file (bool)

**Track Model**

1. Static numerical requirements
   1. The track model shall have one terminal
   2. The track model shall only have two users (Murphy and Track Builder)
   3. The track model shall take the commanded speed in km/hr
   4. The track model shall take authority in meters
   5. The track model shall take the status of the switch position of the track as a boolean value
   6. The track model shall take the status of the lights as a boolean value (on/off)
   7. The track model shall take the status of failures as a boolean value (on/off)

**Train Model**

1. Train Model shall support one terminal
2. Train Model shall only cater to two users- Murphy and Passengers
3. The train model shall take a power command and use Newtonian formulae to calculate a velocity output.
4. The train model shall execute internal calculations in metric units and display output in imperial units.

**Train Controller**

1. Static Numerical Requirements
   1. The controller shall have one terminal
   2. The controller shall only support 2 users at a time (*Train Engineer* and *Train Driver*)
   3. The controller shall take power and brake inputs as percentage
   4. The controller shall take take Station Announcements as string inputs
   5. The controller shall take Commanded Speed as an integer in km/hr
   6. The controller shall take Authority as an integer in meters
   7. The controller shall take exterior lights, emergency brake commands, failures and ‘Mode’ as boolean values
   8. The controller shall take interior lights and door status as integer values (from 0 to 2)
   9. The controller shall input braking and engine power as watts

### 

### 3.3.2 Reliability

**CTC**

1. The current schedule shall always be visible to the dispatcher
2. The current state of the systems, occupied blocks, ect., shall always be visible to the dispatcher.

**Track Controller**

1. The wayside controller shall correctly command the speed of the train.
2. The wayside controller shall always accept a valid PLC file input
3. The wayside controller shall accurately make any changes to the track as necessary (crossings, rail switches, etc.)
4. The wayside controller will always accept user input in manual operation

**Track Model**

1. The system shall correctly report the status of failures of the track to the necessary module(s).

**Train Model**

1. The system shall develop strong and tested algorithms for train control to minimize unexpected failures.
2. The system shall make sure the train model accurately reflects real-world dynamics to reduce surprises.

**Train Controller**

1. The system shall correctly report the current speed of the train.
2. The system shall be at a full stop when approaching its authority.
3. The system shall accelerate and decelerate the train when prompted by the driver if not exceeding limits or authority
4. The system shall correctly open and close the correct doors at the appropriate times.

### 3.3.3 Availability

**CTC**

1. The CTC shall allow the dispatcher to start the system in manual or automatic mode upon startup
2. The CTC shall NOT allow the dispatcher to switch modes while the system is running

**Track Controller**

1. The wayside controller UI shall allow the user to download the current PLC file for future availability.
2. The wayside controller shall NOT be able to continue in automatic mode after being switched to manual operation.
3. The wayside controller shall have the appropriate UI available for the user of the module (PLC programmer) to manually access the system in manual operation.
4. Upon restarting the entire system, the PLC file shall begin reading at the beginning of the file.

**Track Model**

1. The system shall always have availability to block information as it should update with live information.
2. The system shall always have status of the track’s light and switch status as it should update according to the track layout inputs.
3. The system shall always have live checkpoints to guarantee the status of the track using testbenches.
4. The system shall always relay back information in case of failures.

**Train Model**

1. We shall implement regular checkpointing using the testbenches to save the current state of the train model.
2. Using our checkpoints, we shall facilitate recovery to a known good state in case of failures.

**Train Controller**

1. The system shall always have availability of emergency brake action
2. The system shall always have availability of power output, current power and brake output
3. The system shall always have availability of outputs such as Light Status, Door Status, Internal Temperature
4. The system shall always have availability of current KI and KP

### 

### 3.3.4 Safety

**CTC**

1. The CTC shall not dispatch two trains at the same time
2. The CTC shall only give authority to any given part of the track to one train at a time

**Track Controller**

1. The wayside system shall not permit the user of the wayside UI (PLC programmer) to command a speed deemed unsafe to the passengers aboard the train.
2. The wayside system shall not permit the user of the wayside UI (PLC programmer) to command an unsafe authority to an active train.
3. The wayside system shall prevent any unsafe decisions from being made that are commanded within the PLC file.
4. The wayside system shall report any track failures to the CTC office for further action.
5. The wayside controller, as a whole, shall prohibit any action from occurring that will result in the collision of multiple trains, or any action that will endanger the lives of passengers.

**Track Model**

1. The track model shall have live information of the block and the track layout ensures ultimate safety.
2. The track model shall have signals to be sent to the necessary module(s) to ensure appropriate safety measures can be executed.

**Train Model**

1. The train model shall facilitate the train controller’s emergency brake functionality to ensure ultimate safety.
2. The system shall constantly follow a speed limit to avoid collisions.
3. All failure signals shall be sent to the necessary modules immediately so that appropriate safety measures can be executed.

**Train Controller**

1. The Train Controller shall have an emergency brake within the controller
2. The Train Controller shall apply emergency braking when the emergency brake is pulled by a passenger and override any driving commands of the Train Driver until the emergency brake is reset.
3. The Train Controller shall have an override system, such that it overrides actions of Train Driver input if they have the potential to put the train in harm's way.
   1. The override system shall determine speed such that it does not go over the speed limit for that block.
   2. The override system shall apply brakes appropriately based on authority proximity.
   3. The overridden system shall control the power output to the train model such that it does not damage the train.
4. The train shall react to the failure state of brake failureby setting engine power to zero and apply the emergency brakes.
5. The Train Controller shall react to the failure state of power failure by setting engine power to zero and applying the service brakes.
6. The Train Controller shall react to the failure state of signal pickup failure by setting engine power to zero and applying the service brakes.

### 3.3.5 Security

1. We shall only provide code/ github access to verified users such as our teammates and the Professor and TAs
2. We shall only provide Jira access to verified users to maintain security
3. Information shall only be passed between modules as necessary to meet the requirements of the system.
4. Train control shall only be accessible by password

### 3.3.6 Maintainability

1. This system shall be executed in terms of modules.
2. The failure of one module shall not affect the functionality of other modules
3. Proper github commit and push practices shall be followed.
4. Regular scrum meetings shall be conducted to maintain cohesiveness.

### 3.3.7 Portability

1. The entirety of the software shall be launchable for a single executable compatible with Windows 10. Other operating systems may be compatible but not guaranteed.
2. Implementations to be run on embedded hardware shall be programmable via standard practice.

## 3.4 Design Constraints

1. The simulation must run on Windows 10 computer
2. The UI shall not be full screen to accommodate for other module UIs.
3. The system cannot be tested in the real world and can only be tested on the train and track model.

## 3.5 Other Requirements

1. Has an automatic mode with preset scenarios to demo the system
2. The system shall be capable of running at least 10 times faster than wall clock time and shall be able to pause the system.
3. The final project must use at least one or more architectural and design patterns covered during the term.
   1. These must be identified in the architecture and design documentation.
4. Identify any COTS components used in the system.
5. Describe the vital aspects of the system and how it affected the architecture and design of the system.