Software Requirements Specification

Thomas Bui, Justin Carter, Patrick Flaherty, Wesley Miller, Philip Seitz

Change Log

Version Number	Date	Description of the Change	Team Member
1	October 3, 2019	Initial Version	All Members

Contents

1 Introduction

This section of the Software Requirements Specification (SRS) document provides the purpose of the SRS, scope of the SRS, acronyms and abbreviations used in the SRS, list of references, and an overview of the SRS document's structure.

1.1 Purpose

The purpose of this project is to develop a fully operational demonstration of the control center, communications, train, and track control system with simulator for the transit system for review by the Port Authority of Allegheny County (PAAC) procurement committee. The intended use for this project will be for the North Shore Extension of the Light Rail Transit system.

1.2 Scope

The purpose of this project is to design and implement a new Centralized Traffic Control Center and Signaling System for Light Rail Transit system. The final system will be an executable that will demonstrate a fully operational control center, communications, train and track control system simulator for the transit system. This system would include a specification and implementation of a Centralized Traffic Control Center, a Track Controller, a Track Model, a Train Controller, and a Train Model. These modules would all be interface-able with a user. Above all, we hope to provide a comfortable user experience along with the best pricing available.

1.3 Definitions, Acronyms, and Abbreviations

The following are a list of terms, each followed by synonyms and abbreviations. A definition of each of the terms then follows:

- Train Control Signaling System - TCSS
  - The Train Signaling System is the complete deliverable which this SRS shall specify.
- Centralized Traffic Control - CTC
  - The Central Office is one of the major modules of the TCSS. The CTC is able to track the locations of the trains in the railway, as well as communicate to these trains important and vital information.
- Operation Control Center - OCC
  - See *Centralized Traffic Control*
- Central Office

  - See *Centralized Traffic Control*

- Track Controller
  - This is a major module of the TCSS. The Track Controller is responsible for sending important information to both the CTC and the Train Controller (Through the Track and Train Models). It also is responsible for controlling the state of the tracks, such as track switching and railway crossing. This runs a PLC and is configurable on a per-Track Controller basis.

- Train Controller
  - This is a vital *(See Vital)* component of the TCSS. It is responsible for regulating the speed of the train. It is also responsible for controlling various other features of the Train Model. It is directly interface-able by the train driver, and will be capable of being controlled in both automatic and manual modes.

- Track Model
  - This is a major component of the TCSS. It is a digital representation of the track with which the Train Model rides over, and the model which the Track Controller is responsible for. This is configurable via a formatted csv file that will allow the user to simulate and control over any track diagram.

- Train Model
  - This is a major component of the TCSS. It is a digital representation of the Train with which the Train Controller is responsible for. Furthermore it is a representation of the effect of Newtonian physics on the TCSS decision making.

- Port Authority of Allegheny County - PAAC
  - A public transit agency within Pennsylvania.

- Programmable Logic Controller - PLC
  - This is a logic controller that is specially programmable on a per-device basis. It is often adapted for the control of systems and devices that require a high reliability and ease of programming.

- Vital
  - Safety-Critical

- Java Runtime Environment - JRE
  - The JRE is a virtual machine that enables a host computer to execute programs written in Java.

- Office Open XML Workbook - XLSX
  - Office Open XML is a zipped, XML-based file format developed by Microsoft for representing spreadsheets, charts, presentations and word processing documents.
- Graphic User Interface - GUI
  - An electronic user interface displayed through a screen so that a user may interact with software.

1.4 References

- Blackpool Flexity Tram Specifications, Bombardier Inc., 2009. Contained in professor-supplied Project Information directory.

1.5 Overview

The following sections of this document go into more of the details of the TCSS. Section 2 is customer/user focused, discussing the system's interfaces, and requirements for the computer running the program. Section 3 will be of more interest to developers; it outlines requirements that must be met, both functional and structural, in order to consider the system satisfactorily operational. Sections 4 and 5 are more administrative in nature, outlining the process by which future changes to this document and the overall system must be approved. Section 6 includes a list of figures and an index.

2 Overall Description

The TCSS is a software program, comprised of multiple modules, that will be capable of controlling and simulating a train transit system. The several modules will be used by different types of users, such as Train Drivers, Track Builders, and Transit Dispatchers. The system must be capable of operating safely and the simulator must be capable of running for at least a minimum of 24 hours without fail or stop. The client shall be capable of installing a new track layout and the software must be capable of adapting to this new track layout. The specifics of this product are described within this section.

2.1 Product Perspective

The TCSS shall interface with multiple different portions of an already present light rail system.

- The Track Controller will be installed to Wayside Controllers throughout the existing railway.
- The Track Model will be specified according to the existing track, beacons, stations, and crossings.

- The Train Model will be specified according to the physical requirements of the existing rail system's tram. See references(1).
- The Train Controller will be installed within the existing rail system's tram.

Additionally, the TCSS, CTC and Simulator modules will be installed within an personal computer running Windows 10.

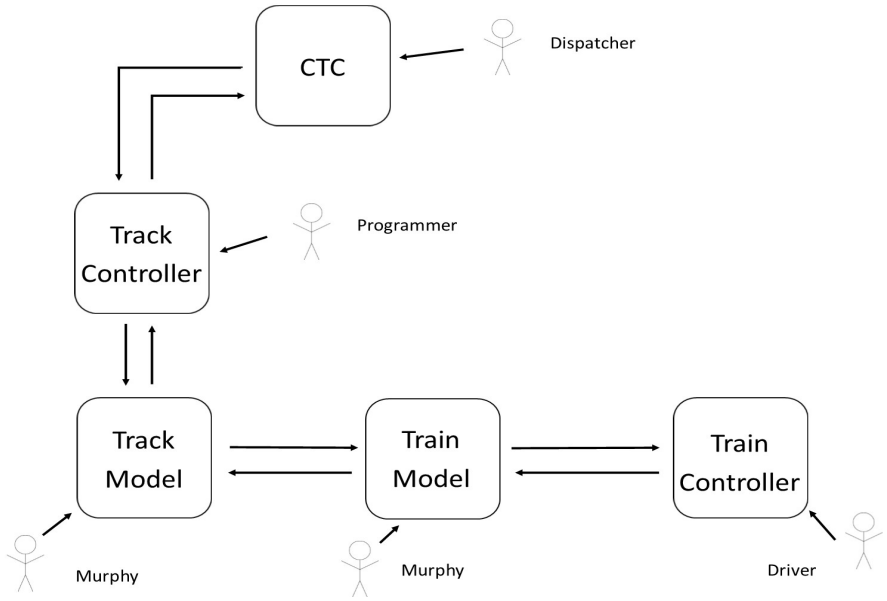

Figure 1: Block Diagram of TCSS

The following subsections describe how the software operates inside various constraints.

2.1.1 System Interfaces

- The PLC must be compatible with the Wayside Controllers of the existing Transit System. In order to accommodate this, the PLC shall be written in one of the following formats:
  - Ladder Diagram

  - Functional Block Diagram
  - Structured Text
  - Instruction List
  - Sequential Function Charts
  - Specially Developed Boolean Based Language

- The CTC must be compatible with the existing format of train schedules. In order to accommodate this, the user shall upload a schedule file, in the form of a xlsx, which will then be parsed for known station names and time requirements. *See 3.1 External Interfaces.*

- The Track Model module must be compatible with the specifications of the existing track layout. The user shall upload a track layout file, in the form a xlsx, which will then be parsed for block properties, station locations, switch locations, speed limit, railroad crossings. *See 3.1 External Interfaces.*

- The TCSS shall utilize and interface with the Java Runtime Environment.

2.1.2 Interfaces

- The CTC shall contain a GUI through which the Dispatcher will be able to view the current state of the railway system. There the Dispatcher can view all of the information about track blocks and active trains in the system and can also dispatch new trains to a location. Additionally the Dispatcher shall be capable of closing and opening specific blocks of the track for maintenance.

- The Train Controller module shall contain a GUI through which the Train Driver will be capable of controlling various aspects of the train's operation, as well as observe other characteristics of the controller's behavior. Additionally, the Train Controller shall be installed on a per tram basis so that a Train Engineer shall specify individual characteristics of the Train Controller.

- The Track Controller module shall contain a GUI where the track engineer shall upload a PLC file containing Boolean logic to each Track Controller . After upload, the track engineer shall see all outputs of the Track Controller GUI.

- The Track Model shall contain a GUI through which the Track Builder will be able to upload a desired track layout file, which the TCSS will use to initialize the track. After the setup, the Track Model will include a GUI that allows a user to observe the current status of all components of the track architecture for the purposes of validation. This includes failure status of track blocks, static information in beacons, locations and orientations of track switches, locations of railroad crossings, train locations, and locations of stations.

- The Train Model shall contain a GUI through which the behavior of a train can be emulated for purposes of validation. It shall also interface directly with other modules to simulate the physical behavior of a real train.

2.1.3 Hardware Interfaces

The system has no hardware interface requirements.

2.1.4 Software Interfaces

- Windows Operating System® by Microsoft®, version 10, from Microsoft®
  - The host system of the TCSS must operate within a Windows operating system for the purpose of file navigation and management.
- Java™ Runtime Environment (JRE) by Oracle®, version 8, from Oracle®.
  - The host system of the TCSS must have the specified installation of the JRE to execute the program.

2.1.5 Communications Interfaces

There are no communication interfaces or protocols for this system.

2.1.6 Memory Constraints

The program is required to run on a Microsoft Surface Pro Tablet, therefore it must not exceed 8GB of RAM. Additionally, the implementation of the TCSS shall utilize as little memory as possible to minimize resource consumption on the host system.

2.1.7 Operations

No business operations affect the design of the TCSS.

2.1.8 Site Adaptation Requirements

- The user can enter Error Simulation mode from the menu screen to access a simulation of the train system where the user can manipulate the system to check failure methods.
- In order to run the program correctly, a track layout excel file (.xlsx) shall be uploaded to the system to produce the track diagram.
- Should the automatic scheduler be used, a schedule excel file (.xlsx) shall be uploaded to the system for reference.
- The PLC files shall be imported on a per-Track Controller basis.

No modification of the user's host system shall be necessary as long as it satisfies the requirements listed within this document.

2.2 Product Functions

The product will provide functions for scheduling, dispatching, monitoring, and tracking trains. Users shall be able to input a track layout and input schedules into the program to simulate a route. The program will notify the users of emergencies and failures and ensure the safety of the passengers onboard.

2.3 User Characteristics

- The Dispatcher – This user oversees the CTC. They need to be able to understand the UI in the system so that they may safely dispatch trains and oversee the system. Additionally, they shall be able to close and open portions of the track.
- The Train Driver – This user is on the train itself and shall be capable of manipulating the state of the train in case of emergencies.
- The Train Engineer – This user is responsible for installing and adjusting the kinetic motion constants of the Train Controller.
- The Passenger – This user is anybody riding the trains of the TCSS, and is capable of manipulating the system via the Train Model's emergency brake.
- The Track Engineer – This user is responsible for manipulating the wayside controller. This user imports the PLC program.
- The Track Builder – This user is responsible for designing and importing the track layout for the Track Model.
- The Validation Engineer – This user is in charge of causing errors to verify that the system works correctly.

2.4 Constraints

- The system shall be easily understood by the user.
- The system shall be written in Java.
- All sub-modules shall integrate with each other.
- The sub-modules shall all contain their own GUI.
- The system shall operate reliably in the environment in which it is operating.

- The system shall be safety critical. In particular the Train Controller and Track Controller shall be vital.
- The Train Controller module is only capable of communicating with the Train Model
- The Train Model module is only capable of communicating with the Train Controller and Track Model.
- The Track Model module is only capable of communicating with the Track Controller and the Train Model.
- The Track Control module is only capable of communicating with the Track Model and the CTC.
- The CTC module is only capable of communicating with the Track Controller
- The beacons shall contain a static 128 characters of information, this can not be changed during run-time of the system or simulation.
- The track shall not be reconfigured during run-time. Neither shall the PLCs be manipulated during run-time.
- The Train Model will only operate at the constraints required in the Blackpool Flexity tram specifications document. *See 1.4 References*
- The track circuit shall pass information with no time or bandwidth constraints. However, the track circuit shall only be used to send the speed limit and authority to the Train Model.

2.5 Assumptions and Dependencies

- The host system for the TCSS shall utilize the Windows 10 operating system.
- The host system for the TCSS shall have the JRE installed.

2.6 Apportioning of Requirements

A primary UX shall be presented to the client on September 26, 2019. Updates and demonstrations shall be presented to the client on October 17, 2019, as well as November 21, 2019. Additionally, the totality of the requirements listed within this SRS shall be completed and delivered to the client on December 12, 2019.

3 Specific Requirements

This section contains all the software requirements of the SRS. This section includes sufficient enough detail to allow designers to satisfy the requirements listed within them.

3.1 External Interfaces

The following are a list of inputs into and outputs from the TCSS. These interfaces interact specifically with the user.

Track Layout Import

- These new track layout files must be properly formatted .xlsx files. The TCSS will perform verification that that imported file is valid
- Units of measure must be made clear according to the file description
- The Track Layout files must be properly formatted according to the following:
  - Each row represents a track block
  - Columns should be ordered as follows:
    * Line
    * Section
    * Block Number (int)
    * Block Length (m)
    * Block Grade (%)
    * Speed Limit (km/hr)
    * Infrastructure
    * Block Elevation (m)
    * Cumulative Elevation (m)
    * Speed Limit (m/s)
    * Time to Travel Block (s)
    * Accel. and Decel. (m/s^2)
    * Constant Speed Time (s)
    * Total Time to Station (s)
    * Dwell Time (s)
    * Total Time to Station w/ Dwell (min)
    * Stop in this Block
    * Stop in Two Blocks
    * Decel. Rate (m/s^2)
- The track distances must be included in meters, and speeds must be reported in meters/second

- This layout is required for the TCSS to operate properly, although it does not directly interact with other interfaces, it certainly does impact the user experience in other interfaces.
- These files shall be entered into the TCSS through a setup screen upon installation.
- There are no window formatting requirements for these files
- There are no special formatting requirements for commands with these files.

Train Schedule Import

- The interface between the Train Schedule Input and the TCSS is responsible for allowing the user to input a new Train Schedule to the TCSS.
- The additional train schedules shall be located within the folder named "Train Schedules"
- All train schedules must be properly formatted. Each row shall represent a stop's information. The column information will be formatted according to the following:
  - Line
  - Type of Stop:Identifier:Underground(If Applicable)
  - Total Time to Station With Dwell
- All time measurements must be included in units of minutes.
- These files will be input and entered into the TCSS through the "New Dispatch" screen by the Dispatcher.
- These schedules have no interactions with other inputs/outputs
- There are no window formatting requirements for these files
- The schedule file should be in the form of an excel file (.xlsx)
- There are no special formatting for commands with these schedules.

Train Controller Kinetic Constants Import

- This interface between the Track Controller and the user shall allow the Train Engineer to adjust, the Ki and Kp values of any one Train Controller.
- These new Ki and Kp values must be decimal point values.
- Ki and Kp do not have units.
- There are no timing requirements.

- This interface has no interactions with other input/output interfaces.
- There are no screen formatting requirements for these files.
- There are no window formatting requirements for these files.
- This data will be entered through a GUI within the TCSS within the Track Model.
- There are no command formatting requirements for these files.

Track Controller PLC Import

- This interface of the Track Controller will allow the Track Engineer to import a PLC program.
- The PLC import file must be a txt file
- The PLC file shall be written using Boolean algebra
- There are no timing requirements.
- There are no screen formatting requirements for these files.
- There are no window formatting requirements for these files.
- There are no command formatting requirements for these files.
- The data shall be entered by the Track Engineer into the physical module.
- This file shall interact with inputs from the Track Controller s and have outputs to the Track Controller .

3.2 Functional Requirements

1. The TCSS shall operate with the following primary workflow:
    - The system shall first process a track layout file and convert that into a working logical data structure to represent the virtual track.
    - The system shall next query the user to enter their relevant module in the TCSS.
    - The system shall then display the information and allow the user to manipulate the relevant portions of the TCSS.
    - All information that the user may not necessarily need to operate successfully will still be presented to the user.
2. The system shall process a train schedule file and accurately display it to the Dispatcher.
3. The system shall allow the Dispatcher to view a map of the current state of the rail system.

4. The system shall allow the Dispatcher to assign a destination, suggested speed, and authority for a train to be dispatched.

5. The system shall allow the Dispatcher to assign a schedule to a specific train, either through an automatically set schedule or by adding individual stops manually. Additionally, the TCSS shall allow a Dispatcher to save a manually created schedule as a preset for other trains to be dispatched.

6. The system shall allow a train to repeat its schedule until instructed to stop.

7. The system shall allow the Dispatcher to open and close blocks of the track for maintenance.

8. The system shall allow the track engineer to input a PLC file to the Track Controller.

9. The system shall accurately display the outputs of the PLC file to the Track Controller interface.

10. The system shall allow for input of a file to configure track layout. Track layout may be reconfigured by inputting a different file. This layout will include all static information with regards to the transit system layout.

11. The system shall simulate the physics of trains, approximating it as a point mass.

12. The system shall use the power command from the Train Controller to adjust the velocity of the train, accounting for mass, load, and grade of the track.

13. The system shall allow the train passenger and train driver to utilize the emergency brake, bringing the train to a complete stop at a predetermined rate of deceleration.

14. The system shall allow the train driver to input a desired speed and automatically calculate the appropriate force command.

15. The system shall allow the train driver to control the state of their train only within safe parameters set by the system (i.e., speed limits).

16. The system shall display current properties such as speed to the train driver.

17. The system shall accurately notify the user of any errors in input.

18. The system shall not allow the user to enter any illegal information, i.e. "forty-five" when the system requires an integer.

19. The system shall update all output values in real-time.

20. The system shall safely respond to commands from the user that may cause the system to fail (i.e., causing two trains to collide).
21. The system shall check the validity of all inputs.

3.3 Non-Functional Requirements

1. The TCSS shall be executable on a Windows OS
2. The Train Controller and Track Controller modules shall have vital architectures.
3. Each module of the TCSS shall have a User Interface
4. Each module of the TCSS shall be submitted as an installable executable.
5. The TCSS shall be submitted as an installable executable.
6. The inputs and outputs of each module of the TCSS shall be visible through the GUI.
7. The design and architecture of the TCSS shall have used at least one of the architecture and design patterns covered in lecture.
8. Whenever relevant, the TCSS shall display information in Customary units instead of Metric.

3.4 Performance Requirements

- The TCSS shall support one terminal.
- The TCSS shall support one active user.
- The TCSS shall support one track layout which may be exchanged.
- The TCSS shall support train schedules.
- The TCSS shall be capable of running at least ten times normal wall-clock speed.

3.5 Logical Database Requirements

The TCSS system does not have any database requirements. All external data storage needs of the system are satisfied through reading from local files.

3.6 Design Constraints

Design constrains for the TCSS include the following:

- The system must run in an operable manner (i.e., should not display large amounts of lag) on a standard, mid-range laptop with an Intel® i5 processor or equivalent.
- Individual modules of the system must also be independently executable, and each must have its own Graphic User Interface.
- The system must be self-contained, i.e., it should be delivered via an executable file or similar archive that includes all files necessary for the system to run.

3.7 Software System Attributes

3.7.1 Reliability

The TCSS shall be capable of operating for at least one simulated day without failure.

3.7.2 Availability

The TCSS shall be used on-demand. There are no recovery requirements in the case of a program failure or program crash.

3.7.3 Security

There are no security constraints for this application.

3.7.4 Maintainability

- Each module of the TCSS shall be capable of executing on its own.
- Each module of the TCSS shall contain its own GUI.
- Each module of the TCSS shall be compatible with the testing scenarios.

Specify attributes of software that relate to the ease of maintenance of the software itself. There may be some requirement for certain modularity, interfaces , complexity, etc. Requirements should not be placed here just because they are thought to be good design practices. If someone else will maintain the system

3.7.5 Portability

This system is to be as portable as possible. To accomplish this, the system will be written using Java; the system will be able to be compiled and run on any system with the JRE installed.

It would be ideal, however, for the system's deliverable to be structured in such a way that the JRE is not a requirement. If this goal can be met, focus will be placed on formatting the system to run on any Windows 10 machine, as specified in section 3.6 Design Constraints on page 17.

4 Change Management Process

Changes to the SRS must be done following a strict process in order to maintain the accuracy of the document, the functionality and consistency of the product, and to keep all team members informed and up to date.

- All changes to the SRS or TCSS must be proposed to the entire team before implementing. Such proposals should be done in writing via the team's designated channels or verbally in meetings with a majority of the team present.
- All changes must be agreed upon by a majority of the team. Minor changes can be discussed and approved in writing via communication channels; major changes must be discussed in one of the team's weekly meetings.
- Any proposed changes discussed in weekly meetings, whether approved or discarded, should be recorded in the meeting notes and distributed to all members of the team for reference and the record.
- If a proposed change will disproportionately affect one team member (i.e., a change specifically to a single module versus a change in overall architecture), then that team member must approve of the change.

5 Document Approvals

Approved: _____
Thomas Bui

Approved: _____
Justin Carter

Approved: _____
Patrick Flaherty

Approved: _____
Wesley Miller

Approved: _____
Philip Seitz

6 Supporting Information

Figures

Index