Software Requirements Specification

for

Port Authority North Shore Extension

Version 1.0 approved

Prepared by Sarah Bunke, Nicholas Schnur, Meyling Taing, Keith Payne, Brandon Bock, Catherine Nalesnik

On Track Trainwreck

9/20/13

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Product Scope 1

1.3 Definitions, acronyms, and abbreviations 1

1.4 References 1

1.5 Overview 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 General Constraints 2

2.5 Assumptions and Dependencies 2

3. Specific Requirements 3

3.1 External interface requirements 4

3.2 Functional requirements 5

3.3 Performance requirements 5

3.4 Design constraints 5

3.5 Software system attributes 3

3.6 Other requirements……………………………………………………………………………… ..4

4. Appendices 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Index 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

This SRS document describes details and requirements for release 1.0 of the Centralized Traffic Control Center and Signaling System for the North Shore Extension Transit System. The documents primary audience is Port Authority of Allegheny County which is the client. This SRS was created based upon the specifications and requirements set forth by the client. The secondary audience is the members of the On Track Trainwreck team that are developing the system. The developers will use the document as design guidelines to implement and also to verify that the system functions correctly.

## Product Scope

The scope is to develop a fully operational prototype of a Centralized Traffic Control Center and Signaling System for the North Shore Extension Light Rail Transit system will be developed for the Port Authority of Allegheny County. The primary goal of the software is to create a safe and reliable system for transportation. This product will include a train and track model to simulate and control the transit system. It will also include a control center and communications between the different parts of the system.

## Definitions, acronyms, and abbreviations

### Centralized Traffic Control (CTC) – The main office in which the dispatcher monitors and controls the transit system. This interacts with the train and track controllers.

### Moving Block Overlay (MBO) – Controls speed limit and authority for the train in addition to scheduling the trains.

### Wayside Controller/Track Controller –

### Authority – The distance the train is allowed to travel.

### Setpoint – The target velocity for the train to travel

## References

### IEEE Standard 830-1993 - Recommended practice for software requirements specification

## Overview

In section two the CTC software is detailed.

# Overall Description

## Product Perspective

The North Shore Extension is an addition to the current transportation system for Port Authority of Allegheny County. The product includes six modules: Train Model, Track Model, Train Controller, Track Controller, Moving Block Overlay (MBO), and the Centralized Traffic Control (CTC). The software is designed to have safety systems in place to ensure that the CTC system runs reliably and transports passengers without harm.

## Product Functions

### To simulate the flow of passengers and trains through the system

### To simulate the scheduling of trains and personnel (or employees)

### Ensuring the safety of the passengers and personnel

### To have an intuitive user interface

### Model the track factors

### Model the train and forces acting upon it.

### Set authority and setpoint speed

## User Classes and Characteristics

### Track Creator

#### Will be able to create the track using a track layout file

### Dispatcher

#### Can close tracks down for maintenance

#### Can route trains

### Conductor

#### Can use the emergency break

#### Gives reassurance to passengers

### Passenger

#### Has a weight of 150 pounds

#### The total passenger weight has an effect on the power needed to move the train

#### Can pull the emergency break

### Scheduler

#### Creates schedule for train operators

#### Accounts for breaks

#### Length of Shift

#### Trains have to return to the yard for shift changes

#### Shifts are 8.5 hours long

#### Breaks shall be 30 minutes and will be provided after 4 hours of driving

### Maintenance

#### When part of the system breaks, maintenance will be alerted to fix it

#### When the fix is accomplished, they change the status to fixed

## General Constraints

The system will run on a Windows 7 Operating System in a Java Runtime Environment (JRE). The Train Controller, MBO, and Track Controller are vital controllers and are implemented in a safety critical manner.

## Assumptions and Dependencies

We assume the computer has Java 1.7 Runtime Environment.

# Specific Requirements

## External interface requirements

### Track Controller

#### The track controller shall receive information from the CTC office that includes information about train authority and speed limits.

#### The track controller shall communicate to the CTC office the current state of the track including railway crossings, signals, and trains.

#### The track controller UI shall display the status of all of the individual PLCs. The display will include the following, PLC Identification, rail status, train presence, and intersection signals if activated.

### Train Model

#### The train model shall receive input from the MBO regarding the authority.

#### The train model shall accept input from the train controller regarding the setpoint speed command, brake command, speed limit, acceleration limit, deceleration limit, route information, temperature control, door open, door close, transponder input, track circuit input, and light controller for tunnels.

#### The train model shall accept emergency brake input from the passengers

## Functional requirements

### Track Controller

#### The track controller shall detect the presence of broken rails

#### The track controller shall control the switching of the track as well as railway crossings. For railway crossings both the lights and cross bar will be controlled. These lights shall be activated one block before the train reaches the crossing and will remain active until the train leaves the block with the intersection.

#### The track controller shall detect the presence of trains on the rails.

#### The track controller must communicate to the train when to turn the lights on.

#### The track controller shall be a programmable unit that runs a PLC program written by the user. The program shall be separate from the implementation of the other track controllers.

### Train Model

#### The train shall have an engine that provides power and acceleration

#### The train movement shall be based on Newton’s laws

#### The train shall have brakes for deceleration and emergencies

#### The train shall display information about the next stop to the passengers

#### The train shall have failure modes for Engine failure, signal pickup failure, and brake failure

## Performance requirements

### Track Controller

#### The track control shall be a vital piece of software.

#### The track controller must communicate the status of the track in time for the CTC office to take appropriate actions.

### Train Model

#### When the train receives input to change the speed, the train must accelerate or decelerate efficiently.

## Design constraints

### Track Controller

#### The track controller shall be written in a safety critical manner.

### Train Model

#### The train model shall have a maximum power output.

#### The train model shall have a maximum breaking capacity.

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

## Software system attributes

The simulation will be capable of running in real-time, 10 times normal speed, and 100 times normal speed.

## Other requirements

# System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

## System Feature 1

<Don’t really say “System Feature 1.” State the feature name in just a few words.>

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## System Feature 2 (and so on)

# Other Nonfunctional Requirements

## Performance Requirements

Inter-module communication must occur as fast as the simulation rate in order to insure that the system runs safely even at higher speeds.

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

## Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

CTC – “Central Train Control”

OCC--“Operations Control Center”

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>