System Test Plan

for

Pittsburgh Train Automation System

Prepared by Alexander Mcmoil, Cameron Kirsch, Kyle Kessler, Justin Pacella, Yuheng Lin, and Yun Dong

Aurora, Inc.

12/14/2023

1 Introduction

1.1 Purpose

The purpose of this document is to specify how we tested our software to ensure it meets all the requirements outlined in our team's SRS. It encompasses comprehensive information on our testing plan and the outcomes derived from each conducted test.

1.2 References

- Software Requirements Specification: OneDrive

1.3 Overview

The introduction section explains the purpose of the document and the purpose of each section. It also includes all references for the document. The overview section covers all general testing information including our objectives, approach, and test coverage. The individual tests section is for individual test plans and results. The tests are roughly sorted by function via the subsection headers.

2 Overview

2.1 Objectives

In summary, the objectives of our tests are to prove that the system functions as intended, highlight elements that either don't work or aren't implemented, and demonstrate our system's adherence to the requirements and constraints outlined in our SRS. Any failed tests shown in Section 3 indicate that a feature is either nonfunctional or not implemented. Each failed test includes details of what specific features failed.

2.2 General Approach

Our team's approach to planning our tests begins with listing all the features of our system as seen in Section 2.3. We used this feature list to determine whether we plan to test each feature. Using this list, we wrote our tests in terms coverage. We started by writing broad tests that test many features at once. If one of the broad tests fails for one or more features, we added subtests to test each of the failed features.

Each test includes information regarding the test's plan and outcome. All our tests are performed on a system-wide basis, implying that we may be unable to run certain tests due to failures in dependent tests. Such situations are explained in the outcome section of each test.

2.3 Feature Coverage

Feature	Description	Tests
System Architecture	Train system adheres to architecture defined in SRS	Not Tested
Automatic Dispatching	CTC Office sends dispatch signals based on schedule file	Tested
Manual Dispatching	CTC Office sends dispatch signals based on user input	Tested
Departure Timing	CTC Office uses arrival, dwell, and departure times to move trains at specific times	Tested
Authority	CTC Office and Wayside Controller determine authority and Train Controller receives authority	Tested

Suggested Speed	CTC Office determines suggested speed and train controller receives it	Tested
Maintenance	CTC Office requests blocks to be placed in maintenance and Wayside Controller receives request. Trains do not enter blocks under maintenance	Tested
Schedule Uploading	Dispatcher uses CTC Office UI to upload a train schedule	Tested
Ticket Sales	Track Model generates ticket sales and sends them to CTC Office	Tested
Track Faults	CTC Office detects track faults and notifies dispatcher	Tested
Automatic Track Logic	Wayside Controller PLC executes a program based on occupancies to perform track logic including switches, signals, and crossings	Tested
Manual Track Logic	Wayside Controller performs track logic based on user inputs	Tested
PLC Uploading	Wayside Programmer may user Wayside Controller UI to upload a PLC program	Tested
Occupancies	Track Model detects occupancies and sends them to Wayside Controller and CTC Office	Tested
Wayside Vitality	Wayside Controller is implemented with safety critical architecture	Not tested
Train Construction	Track Model constructs a train in the yard block when the dispatch signal is received	Tested
Train Destruction	Track Model destructs trains when they return to the yard	Tested
Train Positioning	Track Model tracks train positions to be used internally to determine occupancies	Tested
Track Simulation	Track Model simulates switches, signals, and crossings	Tested
Track Heaters	Track Model simulates track temperature and controls track heaters	Todo
Fault Simulation	Track Model user may trigger track faults	Tested
Beacon Transmission	Track Model stores static beacon packages and sends them to the train	Not Tested
Engine Simulation	Train Model uses a power command to simulate speed	Tested
Train Mass	Train Model uses boarding count to determine train mass	Tested
Door & Light Simulation	Train Model simulates lights and doors	Tested
Cabin Temperature	Train Model simulates cabin temperature and receives heater/air conditioning commands	Tested
Brake Simulation	Train Model simulates both service and emergency brakes	Tested
Train Fault Simulation	Train Model user may trigger train-related failures	Tested
Announcements	Train Controller determines station announcements and sends them to Train Model	Tested
PID Controller	Train Controller uses a PI controller to compute power from velocity error	Tested

Brake Control	Train Controller uses authority and current speed to control the service brake	Tested
Beacon Decoding	Train Controller decodes beacon data from Track Model	Not Tested
Door & Light Control	Train Controller uses track data and beacon package to control doors and lights	Tested
Manual Train Control	Train Controller user may control the speed/power, brakes, doors, etc.	Tested
Train Controller Vitality	Train Controller is implemented with safety critical architecture	Not Tested

3 Individual Tests

3.0 Template

Description	
Feature(s) Tested	
Inputs	
Expected Outputs	
Results	
Tester	
Date Performed	
Outcome	✅ or ❌

3.1 Dispatch Tests

Train Initialization Test	
Description	For this test, we will dispatch one train on the green line to verify that the train is constructed and receives the correct values for authority and suggested speed
Feature(s) Tested	1. Train construction 2. Full system communication 3. Authority of one block
Inputs	1. Use the CTC Office UI window to manually dispatch a train to Dormont station
Expected Outputs	1. Block 0 (the yard) shows that it is occupied on the CTC Office, Wayside Controller, and Track Model UIs 2. The authority of block 0 is shown as 4265ft (1300m) 3. The suggested speed of the train is less than the speed limit of block 0
Results	
Tester	Yun Dong
Date Performed	12/14/2023
Outcome	✅ The train has reached to Dormont and stopped with suggested speed to zero

Basic Dispatch Test	
Description	For this test, we will dispatch one train on the green line to verify that the train travels to the correct station
Feature(s) Tested	1. Authority 2. Occupancy detection & propagation 3. Automatic braking
Inputs	1. Use the CTC Office UI window to manually dispatch a train to Dormont station
Expected Outputs	1. Block 0 (the yard) shows that it is occupied on the CTC Office, Wayside Controller, and Track Model UIs 2. The authority of block 0 is shown as 4265ft (1300m) 3. The suggested speed of the train is less than the speed limit of block 0
Results	
Tester	Yun Dong
Date Performed	12/14/2023
Outcome	✅ The train has successfully go to Dormont

Two Station Dispatch Test	
Description	For this, we will dispatch a train on the green line to Dormont station, then to Mount Lebanon, with a one-minute dwell time.
Feature(s) Tested	1. Dwell time/train departure 2. Authority for multiple stops
Inputs	1. Use the CTC Office UI in manual dispatch mode to add stops for Dormont and Mount Lebanon and dispatch the train
Expected Outputs	1. In the Track Model UI, the train should stop at Dormont station, wait one minute, then travel to Mount Lebanon station
Results	
Tester	Justin Pacella
Date Performed	12/11/2023
Outcome	❌ Train stopped on block before Dormont station and remained there indefinitely

Two Train Dispatch Test	
Description	For this test, we will dispatch two trains to two different stations on the green line.
Feature(s) Tested	1. Authority for multiple trains
Inputs	1. Use the CTC Office UI to manually dispatch a train to Mount Lebanon 2. After 30 seconds, manually dispatch a second train to Dormont
Expected Outputs	1. On the Track Model UI, the first train stops at Mount Lebanon 2. The second train stops at Dormont
Results	
Tester	Justin

Date Performed	12/11/2023
Outcome	❌ Train 2 becomes stuck in block 64 despite having a speed of 15mph. The Train Controller UI for train 2 also displays a decreasing negative authority value.

Timed Dispatch Test	
Description	For this, we will manually dispatch a train to Dormont station with a set arrival time.
Feature(s) Tested	1. Arrival/departure time
Inputs	1. Use the CTC Office UI to manually dispatch a train to Dormont station with an arrival time one hour past the current simulation time
Expected Outputs	1. The train should not spawn until roughly a few minutes before the arrival time 2. The train arrives at Dormont station within a few minutes of the set arrival time
Results	
Tester	Justin Pacella
Date Performed	12/11/2023
Outcome	❌ When the dispatch button was clicked, the program threw an overflow error from CTC Office

Automatic Dispatching Test	
Description	For this, we will upload a schedule using the CTC Office UI and see if trains are dispatch properly
Feature(s) Tested	1. Automatic Dispatching 2. Schedule Uploading 3. Timed Dispatching
Inputs	1. Use the CTC Office UI to load Schedule.xlsx
Expected Outputs	1. Train should be dispatched at the time specified by the schedule 2. Train should travel to each station in the schedule
Results	
Tester	Justin Pacella
Date Performed	12/11/2023
Outcome	❌ Unable to complete test. Schedule file is not loaded

3.2 PLC-Related Tests

Basic Track Logic Test	
Description	For this test, we will dispatch one train on the green line to verify the functionality of switches, signals, and crossings.
Feature(s) Tested	1. PLC functionality 2. Wayside Controller/Track Model communications
Inputs	1. Use the CTC Office UI window to manually dispatch a train to Pioneer station

Expected Outputs	1. When the train passes through block 65, the signal on block 65 will be red while the train is in that block. 2. When the train enters block 74, the switch on block 77 will connect to block 76. When the train reaches block 77, the switch will connect back to block 101. 3. When the train enters block 21, the crossing in block 19 will close. It will remain closed until the train exits block 17.
Results	
Tester	Justin Pacella
Date Performed	12/10/2023
Outcome	✅

Switch, Signal & Crossing Logic Test (Two trains)	
Description	For this test, we will dispatch two trains on the green line to verify the functionality of switches, signals, and crossings.
Feature(s) Tested	1. Multi-train PLC functionality
Inputs	1. Use the CTC Office UI window to manually dispatch a train to Pioneer station 2. Wait 30 seconds, then perform step one a second time.
Expected Outputs	1. Train 2 will wait on block 76 until all blocks between 77 and 85 are clear.
Results	
Tester	Justin Pacella
Date Performed	12/11/2023
Outcome	❌ Test couldn't be completed. Train 2 becomes stuck on block 64

Alternative PLC program Test	
Description	For this test, we will dispatch one train on the green line. We will also load an alternative PLC file for Wayside Controller 3 that inverts the states of the crossing to demonstrate that the PLC is not hard coded into the software.
Feature(s) Tested	1. PLC loads an alternative program 2. PLC functions with an alternative program
Inputs	1. Use the CTC Office UI window to manually dispatch a train to Pioneer station 2. Use the Wayside Controller UI window to load node_3_alt.plc for Wayside Controller 3
Expected Outputs	1. The crossing in block 19 will be open when the train is between blocks 17 and 21. It will remain closed otherwise.
Results	
Tester	Justin Pacella
Date Performed	12/11/2023
Outcome	✅

3.3 Train Simulation Tests

PI Controller Test	
Description	For this test, we dispatch a train and use the Train Controller UI to verify the functionality of the PI controller and speed physics
Feature(s) Tested	1. PI Controller 2. Manual Train Control 3. Engine Simulation
Inputs	1. Use the CTC Office UI to dispatch a train to Pioneer station 2. Use the Train Controller UI to manually set the commanded speed 3. Manually change the gain values for K_p and K_i
Expected Outputs	1. Upon dispatching, the train should accelerate to the automatically set commanded speed 2. When the commanded speed is changed, the train should accelerate or decelerate to the new command 3. When gain values are increased or decreased, the train's acceleration should increase or decrease
Results	
Tester	Yun Dong
Date Performed	12/10/2023
Outcome	✅

Manual Door & Light Control Test	
Description	For this test, we will dispatch a train on the green line, stop it, and manually control lights and doors
Feature(s) Tested	1. Door simulation 2. Lights simulation
Inputs	1. Manually dispatch a train to Poplar station 2. Engage the emergency brake 3. Use the Train Controller UI to open/close the left and right doors 4. Use the Train Controller UI to toggle interior and exterior lights
Expected Outputs	1. Train Model UI shows that train model responds to manual commands
Results	
Tester	Justin Pacella
Date Performed	12/12/2023
Outcome	✅

Cabin Temperature Test	
Description	This test is for the train model temperature simulation and setpoint control
Feature(s) Tested	1. Cabin temperature simulation 2. Cabin temperature control
Inputs	1. Manually dispatch a train

	2. Use the Train Controller UI to manually change the temperature setpoint
Expected Outputs	1. The Train Controller UI temperature indicator should change according to the setpoint
Results	
Tester	Justin Pacella
Date Performed	12/12/2023
Outcome	✅

Train Mass Test	
Description	This test is for the train model mass simulation based on number of passengers and crew
Feature(s) Tested	1. Train mass
Inputs	1. Manually dispatch a train to a station 2. Use Track Model UI to observe the increase in ticket sales 3. Use Train Model UI to observe increase of passengers when doors open at a station 4. Use Train Model UI to observe the increase in train mass
Expected Outputs	1. The Train Model UI train mass indicator and passenger count should change according to the ticket sale
Results	
Tester	Yuheng Lin
Date Performed	12/14/2023
Outcome	❌ Test couldn’t be completed. Ticket sales is not being sent by Track Model

Train Announcements Test	
Description	This test is to test Train Controller sending correct announcement to Train Model and Train Model displays it correctly
Feature(s) Tested	1. Train announcements
Inputs	1. Manually dispatch a train to a station 2. Use Train Controller UI to observe the correct next station and current station 3. Use Train Model UI to observe the announcement being sent to Train Model and the correct announcement
Expected Outputs	2. The Train Model UI displays the correct announcements
Results	
Tester	Yuheng Lin
Date Performed	12/14/2023
Outcome	✅

3.4 Track Simulation Tests

Switch Functionality Test	

Description	For this test, we will dispatch a train on the green line and manually change switch 85 to see if the train follows the switch
Feature(s) Tested	1. Track simulation
Inputs	1. Manually dispatch a train to Overbrook station 2. Use the Wayside Controller UI to manually change switch 85 to connect blocks 85 and 100
Expected Outputs	3. On the Track Model UI, when the train reaches block 85, it continues to block 100 (wrong direction)
Results	
Tester	Justin Pacella
Date Performed	12/11/2023
Outcome	✅

3.4.1 Light Test

Light Functionality Test	
Description	For this test, manually change the signal states from the Wayside Controller
Feature(s) Tested	1. Track Simulation
Inputs	1. Open Wayside Controller and Track Model. 2. Select the node for the light that will be tested 3. Select Signal Control 4. Set Manual Mode on 5. Select Signal to toggle 6. Click Toggle light 7. Verify in Track Model Signal switched visually 8. Select the crossing block in Track Model and verify block information changed as well
Expected Outputs	1. Expecting Light on Track Model to change colors and match the color of the Wayside Controller
Results	
Tester	Kyle Kessler
Date Performed	12/14/2023
Outcome	✅

3.4.1 Crossing Test

Crossing Functionality Test	
Description	For this test, manually change the crossing states from the Wayside Controller
Feature(s) Tested	2. Track Simulation
Inputs	1. Open Wayside Controller and Track Model.

	2. Select the node for the Crossing that will be tested 3. Select Crossing Control 4. Set Manual Mode on 5. Select Crossing to toggle 6. Click Toggle Crossing 7. Verify in Track Model crossing switched visually 8. Select the crossing block in Track Model and verify block information changed as well
Expected Outputs	1. Expecting Crossing on Track Model to change colors and match the status of the Wayside Controller
Results	
Tester	Kyle Kessler
Date Performed	12/14/2023
Outcome	✅

3.4.2 Occupancy Test

Occupancy Functionality Test	
Description	For this test, send a train through the system and verify occupancies in all nodes show occupied
Feature(s) Tested	1. Track Simulation
Inputs	1. Open Wayside Controller, CTC, and Track Model. 2. Schedule a train in CTC (do not dispatch) 3. NOTE: As the train runs move to node that receives blocks information. 4. As the train crosses each track block on the track model Verify that the block is also “occupied” on the Wayside Controller.
Expected Outputs	1. As every block is crossed by a train, it shows the block status is occupied in the Track Model, and Wayside Controller.
Results	
Tester	Kyle Kessler
Date Performed	12/14/2023
Outcome	✅

3.4.3 Temperature Test

Switch Functionality Test	
Description	For this test, we adjust the temperature of the rail line to verify heaters work.
Feature(s) Tested	1. Track simulation
Inputs	1. Open Track Model, testbench 2. Select a track block on the line you wish to test 3. Type a value of “35” or less into the text box right of “Ambient Temperature(F) 4. Click button “Temperature”

	5. Verify Heaters turned ON under “block information”, and that the Temperature also changed. 6. Change the value to “36” or higher 7. Verify Heaters turned OFF and Temperature changed
Expected Outputs	1. Expecting temperature to change and heaters to turn on
Results	
Tester	Kyle Kessler
Date Performed	12/14/2023
Outcome	✅

3.4.4 Block Information Test

Block Information Test	
Description	For this test, we select blocks with switches, stations, and crossings and verify that the block information outputs for them on the right.
Feature(s) Tested	1. Track simulation
Inputs	1. Open Track Model 2. Select Green and Red Lines 3. Select various blocks with stations, rail blocks, crossings, and switches 4. Verify that the information matches what it should, and everything is in Imperial on the UI 5. Repeat steps 3 and 4 as needed
Expected Outputs	1. All block, station, crossing, switch, information displays properly
Results	
Tester	Kyle Kessler
Date Performed	12/14/2023
Outcome	✅

3.4.5 Track Module Dwelling/Departing Test

Train Dwelling/Departing Test	
Description	For this test, when a train is “dwelling” verify that CTC is calling the functions to set the train to dwell in a station and departing a station. Once this occurs Disembarking/Boarding information in stations should change, and Ticket Sales decreases by the amount of people boarding the Train. Also, Throughput should increase, and Total Passengers (total number of passengers in the system) should increase, and decrease based on boarding/disembarking.
Feature(s) Tested	1. Track simulation 2. CTC
Inputs	1. Open Track model, and CTC 2. Schedule a train to a station several stations in CTC 3. In Track Model, select the station the train is to stop at first. a. Verify once the train reaches station, the color changes of the train. b. Verify “Disembarking”, “Boarding” also displays values 4. Once train leaves station

	a. Verify “Throughput” increases, and the “Total Passengers” increase matches the number of people that are in the system.
Expected Outputs	2. All block, station, crossing, switch, information displays properly
Results	
Tester	Kyle Kessler
Date Performed	12/14/2023
Outcome	❌CTC does not call the functions, Station does not know if a train is dwelling or not. Therefore, Train color does not change, no boarding/disembarking information is displayed, nor is throughput or total passenger's change.

Commented [KK1]: If CTC calls function, perform this test

3.5 System Safety Tests

Collision Prevention Test	
Description	For this test, we will dispatch two trains on the green line with directly conflicting routes
Feature(s) Tested	1. Vital authority 2. Wayside authority selection 3. Automatic braking
Inputs	1. Manually dispatch a train to Dormont station 2. After 30 seconds manually dispatch a train to Mount Lebanon
Expected Outputs	1. Train 1 stops at Dormont station 2. Train 2 stops on block before Dormont station
Results	
Tester	Justin Pacella
Date Performed	12/11/2023
Outcome	❌ Unable to perform test. Train 2 becomes stuck on block 64

Emergency Brake Test	
Description	For this test, we will dispatch a train on the green line. Before it reaches its destination, we will trigger the emergency brake
Feature(s) Tested	1. Emergency Brake
Inputs	1. Manually dispatch a train to Poplar station 2. Trigger the emergency brake from the Train Controller UI after 1 minute 3. Disengage the emergency brake 4. Re-perform the test using the Train Model UI instead of Train Controller
Expected Outputs	1. In the Train Controller (or Train Model) UI, the current speed drops to zero 2. In the Track Model, the train comes to a complete stop 3. When the brake is released, the train accelerates and continues to Poplar station on both UIs

Results	
Tester	Justin Pacella
Date Performed	12/11/2023
Outcome	✅

3.6 Track Failure Tests

Block Maintenance Test	
Description	For this, we place a block in maintenance and test if trains stop or not
Feature(s) Tested	1. Maintenance mode 2. Vital Authority
Inputs	1. Use the CTC Office UI to place block 69 under maintenance 2. Manually Dispatch a train to Poplar Station
Expected Outputs	1. The train stops before entering block 69
Results	
Tester	Justin Pacella
Date Performed	12/11/2023
Outcome	❌ Unable to perform test. CTC Office UI doesn’t have maintenance implemented

Broken Rail Test	
Description	For this test, we will break a rail on the red line and observe the results in Track Model, Wayside Controller, and CTC Office UIs
Feature(s) Tested	1. Track Faults
Inputs	1. Use the Track Model UI to trigger a broken rail fault on block 18 of the red line
Expected Outputs	1. The Track Model shows the failure and block 18 lights up orange 2. The Wayside Controller and CTC Office show that block 18 is occupied
Results	
Tester	Justin Pacella
Date Performed	12/11/2023
Outcome	✅

Track Circuit Failure Test	
Description	For this test, we will bridge a track circuit the red line and observe the results in Track Model, Wayside Controller, and CTC Office UIs
Feature(s) Tested	1. Track Faults
Inputs	1. Use the Track Model UI to trigger a track circuit failure on block 77 of the red line
Expected Outputs	1. The Track Model shows the failure and block 77 lights up orange 2. The switch on block 27 changes as if block 77 were occupied 3. The Wayside Controller and CTC Office show that block 77 is occupied
Results	

Tester	Justin Pacella
Date Performed	12/11/2023
Outcome	✅

Power Failure Test	
Description	For this test, we will cut the power to block 47 on the red line and observe the results in Track Model, Wayside Controller, and CTC Office UIs
Feature(s) Tested	1. Track Faults
Inputs	1. Use the Track Model UI to trigger a power failure on block 47 of the red line
Expected Outputs	1. The Track Model shows the failure on for block 47 2. The Wayside Controller and CTC Office show that block 47 is occupied
Results	
Tester	Justin Pacella
Date Performed	12/11/2023
Outcome	❌ It seems like the power failure button is not connected.

Brake Failure Test	
Description	For this test, we will trigger brake failure on the Train Model UI and observe the results in Train Controller and Train Model
Feature(s) Tested	1. Train Faults
Inputs	1. Dispatch a train manually to a station 2. Use the Train Model UI to trigger a brake failure and try to engage emergency brake 3. Open Train Controller UI to try to engage service and emergency brake
Expected Outputs	1. The Train Model shows brake failure active 2. The service brake will not engage 3. The emergency brake will not engage 4. Power is zero in Train Model
Results	
Tester	Yuheng Lin
Date Performed	12/14/2023
Outcome	✅

Power Failure Test	
Description	For this test, we will trigger power failure on the Train Model UI and observe the results in Train Controller and Train Model
Feature(s) Tested	1. Train Faults
Inputs	1. Dispatch a train manually to a station 2. Use the Train Model UI to trigger a power failure 3. Open Train Controller UI to try to increase the power by increasing commanded speed or kp/ki
Expected Outputs	1. The Train Model shows power failure active 2. The power in the Train Model is 0

Results	
Tester	Yuheng Lin
Date Performed	12/14/2023
Outcome	✅

Signal Pickup Failure Test	
Description	For this test, we will trigger signal pickup failure on the Train Model UI and observe the results in Train Controller and Train Model
Feature(s) Tested	1. Train Faults
Inputs	1. Dispatch a train manually to a station 2. Use the Train Model UI to trigger a signal pickup failure 3. Open Train Controller UI to see if authority and suggested speed is updated
Expected Outputs	1. The Train Model shows signal pickup failure active 2. The power in the Train Model is 0 3. Authority and suggested speed is no longer being updated
Results	
Tester	Yuheng Lin
Date Performed	12/14/2023
Outcome	✅

3.7 Throughput Tests

Ticket Sales Test	
Description	For this test, we will simply let the simulation run and view the ticket sales for each station via the CTC Office
Feature(s) Tested	1. Ticket sales generation
Inputs	
Expected Outputs	1. In the Track Model, ticket sales increase
Results	
Tester	Justin Pacella
Date Performed	12/11/2023
Outcome	✅ Ticket sales increase.

3.7.1 Throughput Tests

Ticket Sales Test 2	
Description	For this test, we will simply let the simulation run and view the ticket sales for each station via the CTC Office
Feature(s) Tested	2. Ticket sales generation
Inputs	
Expected Outputs	2. In the Track Model, ticket sales increase
Results	

Tester	Justin Pacella
Date Performed	12/11/2023
Outcome	❌ Ticket sales increase when application is “paused”. Does not use simulation time to increase tickets