**Master Test Plan (MTP) Train Model**

**Overview**

Tests for the Train Model will be organized by user type. The three user categories for tests are: Train Controller, Passenger, and Murphy. Testing will be performed periodically during development.

**Murphy Tests**

* **Engine Fail**
  + Inputs: Murphy engine fail
  + Expected Outputs: Display of failure on Train UI, power input to be zero, commanded brakes to turn on
* **Signal Fail**
  + Inputs: Murphy signal fail
  + Expected Outputs: Display of failure on Train UI, destination unknown
* **Brake Fail**
  + Inputs: Murphy brake fail
  + Expected Outputs: Display of failure on Train UI, power input to be zero

**Passenger Tests**

* **Emergency Brake**
  + Inputs: Emergency brake button is pressed
  + Expected Output: Display on UI that E-Brake has been pulled, train controller inputs emergency brake, power input is zero, train comes to a stop.

**Train Controller**

* **Set Speed**
  + Inputs: power
  + Expected Outputs: the correctly calculated commandedSpeed
* **Moving Block Authority**
  + Inputs: moving block authority
  + Expected Outputs: to reset the authority of the train as moving block authority.
* **Real Time Updates**
  + Inputs: UI
  + Expected Outputs: a continuously updating UI to display updated real time power, speed, acceleration, deceleration, doors, lights, authority, and passenger count.

**Track Circuit Data**

* **Track Circuit Data**
  + Inputs: uint64\_t
  + Expected Outputs: completion of sending Track Circuit Data received from the Track Model to the Train Controller.
* **Beacon Data**
  + Inputs: uint16\_t
  + Expected Outputs: completion of sending Beacon Data received from the Track Model to the Train Controller.

**Template for Tests**

* Templates for each test will contain the outcome of the test (Pass or Fail), the specific outcome expected versus what happened, when the test was performed and which user requested it. This data will be given as an output template as seen below formatted in lines of text after a test.

| **Test Case** | **Inputs** | **Expected Output** | **Pass/Fail** | **Failure Description** | **Tester** | **Data Tested** |
| --- | --- | --- | --- | --- | --- | --- |
| **Create Train** | | | | | | |
| Create a Train | Boolean to determine if the train is hardware or software and boolean to determine if it's on the green or red line. | A Train instance is created along with a UI for that train.  The Train is pushed into a vector for storage. | Pass | None | Justin | 12/13 |
| **Speed Calculations** | | | | | | |
| Acceleration Calculations | Input of power to calculate initial acceleration. | Acceleration is calculated and displayed. Speed is calculated through acceleration and displayed. | Pass | None | Bryan | 12/12 |
| Deceleration Calculations | When the emergency brake is activated or the service brake is activated. | Train starts decelerating before the station and stops at the station. Deceleration values are working and does not go into negative speed. | Pass | None | Bryan | 12/12 |
| **Data Transfer** | | | | | | |
| Track Circuit Data Transfer | Track Circuit Data from the Track Model is sent and I pass the data through signals and slots to the Train Controller. | Data is transferred to the Train Controller and the controller adjusts inputs to the Train Model in accor- dance with Track Circuit Data. | Pass | None | Owen | 12/13 |
| Beacon Data Transfer | Beacon Data from the Track Model is sent and I pass the encoded data to the Train Controller. | Data is transferred to the Train Controller and the controller adjusts inputs to the Train Model in accor- dance with Track Circuit Data. | Pass | None | Owen | 12/13 |
| Moving Block Authority | Receiving data from MBO from signal. | Receive an authority distance and authority speed to display on my UI and authority sent to my controller. | Pass | None | Owen | 12/13 |
| **UI Displays and Inputs** | | | | | | |
| Updating values | A system timer to update the UI. | The UI is constantly updating values. | Pass | None | Owen | 12/7 |
| Adjusting Temperature | A slider that is initiated to 60°F. | Temperature is changed inside the train and displayed on UI. | Pass | None | Owen | 12/7 |
| Engine Failure | Button activates an error message on UI and activates engine failure signal. | Error message is shown and power is cut off. | Pass | None | Owen | 12/7 |
| Signal Failure | Button activates an error message on UI and activates the signal fail signal. | Error message is shown and power is cut off. | Pass | None | Owen | 12/7 |
| Brake Failure | Button activates an error message on UI and activates brake fail signal. | Error message is shown and power is cut off. Signal for Brake failure is sent and power is cut off and slowly lets the train stop. | Pass | None | Owen | 12/7 |
| Passenger Emergency Brake | Passenger presses the button to activate an emergency brake. | Signal for emergency brakes is sent and emergency brakes are activated. | Pass | None | Owen | 12/7 |