**SOFTWARE REQUIREMENT SPECIFICATIONS**

**FOR**

**SUBWAY SYSTEM SIMULATOR**

**Client: Dr. Michael Oudshoorn**

**Team Name: Elite**

**Prepared By: Revathi Chittepureddy**

**Mohammad Ghori**

**Mahesh Babu Kadiyala**

**Kranthi kumar Mukka**

**Grevil Gonsalo Lopes**

**Balaji Vinay Banala**

**Pruthvi Krishna Tarugu Subbaiah**

**Overview:**

This Software requirement specifications document provides the information about all the functionalities of the **Subway System Simulator** which includes train system, track system, dispatcher system and fault layer system.

The initial part (Section 1) of the document is the Introduction, which describes about the Scope, Purpose and definitions of the terms that are used in the document.

The next chapter (Section 2) describes about the General Description, this part of the document briefly describes about System Perspective, User Characteristics, Constraints and Assumptions of the project.

The functional requirements of each individual system (Train system, Track system, Dispatcher system and Fault layer system) in Section 3.

Contents

[1. Introduction 4](#_Toc468842163)

[1.1 Purpose 4](#_Toc468842164)

[1.2 Project Scope 4](#_Toc468842165)

[1.3 Definitions 4](#_Toc468842166)

[2. General Description 5](#_Toc468842167)

[2.1 System Perspective 5](#_Toc468842168)

[2.2 User Characteristics: 5](#_Toc468842169)

[2.3 Constraints: 5](#_Toc468842170)

[2.4 Assumptions: 5](#_Toc468842171)

[3. Functional Requirements 6](#_Toc468842172)

[3.1 Train System: 6](#_Toc468842173)

[3.1.1 Passenger Train details 6](#_Toc468842174)

[3.1.2 Freight Train details 8](#_Toc468842175)

[3.1.3 Live Status of Passenger train 8](#_Toc468842176)

[3.2 Track System: 9](#_Toc468842177)

[3.2.1 Section details 11](#_Toc468842178)

[3.2.2 Train position 12](#_Toc468842179)

[3.2.3 Switch direction 12](#_Toc468842180)

[3.2.4 Get Routes and Block Route 13](#_Toc468842181)

[3.3 Dispatcher System: 14](#_Toc468842182)

[3.3.1 Live Map 14](#_Toc468842183)

[3.3.2 Integration 14](#_Toc468842184)

[3.4 Fault Layer System: 15](#_Toc468842185)

[3.4.1 Corrupted Message 15](#_Toc468842186)

[3.4.2 Fail to pass some messages 16](#_Toc468842187)

[4. Non Functional Requirements 16](#_Toc468842188)

[4.1 Performance 16](#_Toc468842189)

[4.2 Reliability 16](#_Toc468842190)

[4.3 Availability 16](#_Toc468842191)

[4.4 Security 16](#_Toc468842192)

[4.5 Maintainability 16](#_Toc468842193)

[4.6 Portability 16](#_Toc468842194)

# Introduction

## Purpose

This requirements documentation is to provide detailed description of the functionalities of the Subway System Simulator Project (In general and specific to our model). The purpose of the document is to collect and analyze all ideas that have come up to define the system, its requirements. This document provides detailed overview of our subway system simulator.

## Project Scope

This project is to develop a simulator for an automated train system. In our subway system simulator, trains will operated automatically without need of driver. The subway system simulator should be operated in safe conditions giving high priority to passengers life add without any property damage.

## Definitions

|  |  |
| --- | --- |
| Term | Definition |
| System | A system is a set of interactive and interdependent component parts forming a complex whole. |
| Software Requirement Specifications | This refers to a document which contains all functions of the simulator system, assumptions and constraints |
| User | Who operates subway system simulator |
| Database | Collection of all the information monitored by this system |
| Mockup | Prototype of the user interface |
| Stakeholder | Any person who involves in the project not a developer |
| Author | Person submitting an article to be reviewed. |
| Dispatcher | It is to monitor location of the trains. Except start, stop and emergency stop it does not do anything. |
| Emergency Stop | Stops all trains |

# General Description

## System Perspective

1. This is an automated system without need of driver
2. This system provides train’s information dynamically on the user interface while it is moving.
3. The database maintains all the data about the train like location of the train, section maximum speed, etc.

## User Characteristics:

1. There are two types of users to interact with this system. One is admin/developer and second is user. Admin can update the maximum speed of the sections.
2. User can interact with the system and can view where the train is located, he can have access to stop the train in case of emergency(emergency situation means If heartbeats are not received more than thrice in a row by the dispatcher from train or if user observes something went wrong on the dispatcher UI).

## Constraints:

1. This is a standalone application i.e., The simulator should operate only on desktop (Windows/Linux/Mac) it will not be operated on mobile operating systems like android.
2. If the user wants to stop any train then he need to stop whole train system is a general constraint in this system.
3. User can restart the system but trains should start from starting point rather than where they were stopped

## Assumptions:

1. We are assuming limited stops for express trains and regular trains will stop at all stations.
2. Dispatcher is not allowed to modify the speed limits of the sections.
3. There is tolerance rate of 8% is used for the corrupted messages, 2% of missing messages and rest of the messages are considered to be delivered accurately.
4. A log file is maintained to store the message transfers from one system to the other system which gets cleared at the end of the day.

# Functional Requirements

1. Subway System’s features are the functional requirements of the system. Each functionality is described in detail inside the core features. Core features are the essential functionalities of the system. These core features are subdivided based on the sub system identified. These sub-systems include Train system, Track system, Dispatcher system and Fault layer system.

## Train System:

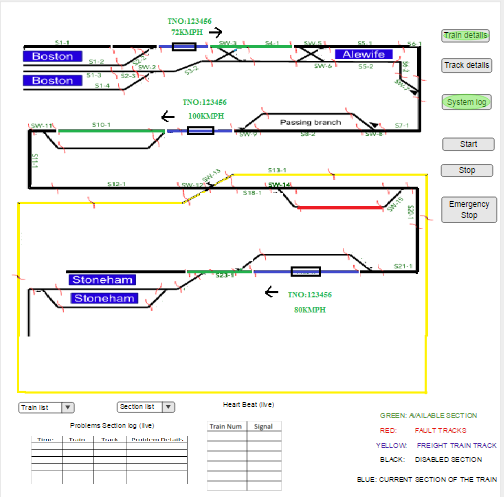
1. Functional requirements about train system includes each and every detail necessary for the system to ensure that trains reaches the correct destination taking the right route and proper functioning of the entire system. This train system includes four passenger trains and one freight train. This freight train have a separate track which has an intersection with the passenger train track. This freight train stops within its own track. Passenger train is not allowed to enter the freight track and freight train is not allowed to enter the passenger track. Among both trains i.e. freight train and passenger train, freight train is given high priority. Trains acceleration or deceleration depends on the gradient of each individual section. These gradient details are stored in the track database as well.

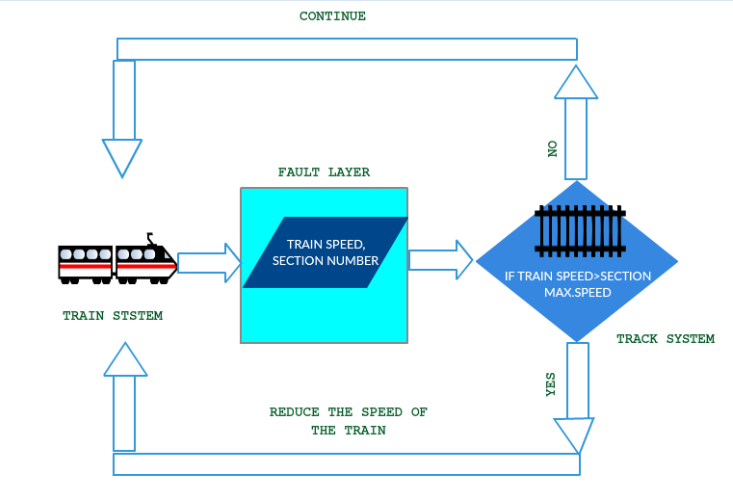
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Train ID | 2663 | 0860 | 6070 | 1995 | 3214 |
| Train type | Passenger | Passenger | Passenger | Passenger | Freight |
| Length (m) | 120 | 120 | 100 | 140 | 350 |
| Unloaded\_weight (kg) | 367,335 | 284,798 | 251,239 | 400,894 | 750,984 |
| Max\_passengers | 740 | 544 | 512 | 480 | 0 |
| Max\_speed (km/hr) | 145 | 135 | 137 | 110 | 100 |
| Minimum time to reach 100 km/hr (seconds) | 112 | 138 | 156 | 124 | 480 |
| Minimum time to stop from 100 km/hr (seconds) | 26 | 38 | 43 | 31 | 69 |

### Passenger Train details

1. If the dispatcher needs any other information about a particular train that is not available in the map he can view about that particular train in more detail by pressing the train details button available on the dispatcher system. Passenger train is considered to be stopped at a single section when it comes to halt at a station with the help of sensors available at the front and rear ends of the sections.  This halted train stays at the station for 2 minutes and resumes its travel to the next destination. One of the passenger train is an express train which travels between Boston and Stoneham which stops at Boston, Melrose and Stoneham. The other trains stop at each and every station. The details include

* Current Location of the train.
* Speed of the train.
* Sections of the track that train occupies.
* Sections of the track that train requires to reach the destination.
* Information about the switch which is ahead of the train that may change the direction of the train.





#### User Requirements

When the dispatcher wants to know more information about a particular train in detail, he can click on the train details. This displays a page describing about all the four trains and also freight train as well.

#### System Requirements

Train system and Track system should communicate with each continuously to know the position of train at a particular point of time. As the train keeps moving the details keep updating and the same should be reflected to the dispatcher while viewing.

### Freight Train details

1. Freight train will be stopped by the user only during the emergency situations. Freight train is considered to be in motion continuously at constant speed in clockwise direction. Dispatcher can just view the train, to make sure that the passenger train and freight train cannot come in contact with each other.

#### User Requirements

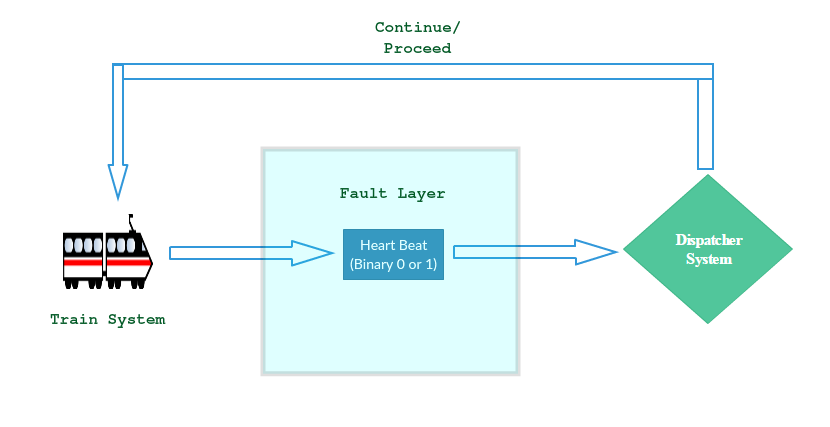
To know the position which refers to the current section(s) of track on which the freight train is on, dispatcher can click on the train details.

#### System Requirements

As the train keeps moving the details will get updated and the same should be reflected to the dispatcher while viewing.

### Live Status of Passenger train

1. This is one of the main functionality of the system. This is the feature that intimates the dispatcher if that particular train in working continuously. To intimate this, the train sends the signals referred as heartbeat to the dispatcher system to intimate that it is working fine for every 3 seconds. If the dispatcher system fails to receive the signal from a particular train for 9 seconds, then the emergency stop functionality is invoked itself halting all the trains for safety purposes. After the dispatcher identifies the issue and solves it, trains can be resumed to move one by one.



#### User Requirements

User just sees the status of each train.

#### System Requirements

Each train should be able to transmit signal to the dispatcher system every 3 seconds, if it fails to send heartbeat within 9 seconds the system triggers the function of the emergency stop module.

## Track System:

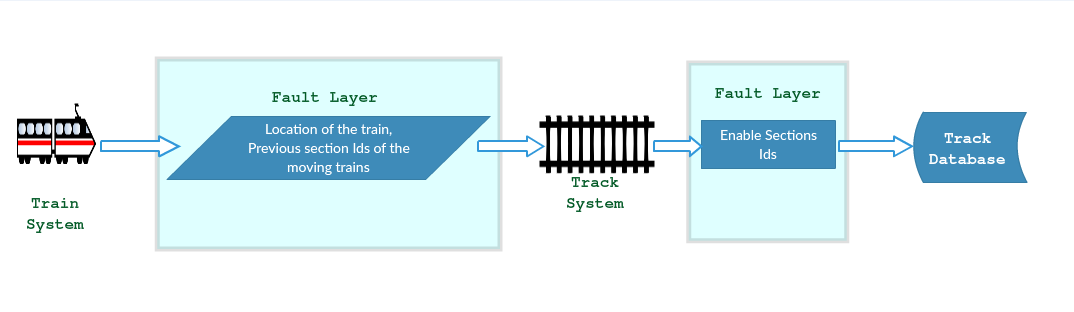
1. The Entire track of the system is divided into multiple track sections and each section is assigned with its own id. This track information regarding the section ids, speed limits is with the track database. Track is electrified. Trains acceleration and deceleration also depends on the natural factors like gravity, weather conditions. The end stations of the track have a buffer region for the train so that when the train reaches the end station the train can use that space to be at rest when the train is not in use. In emergency situations when the dispatcher hits the emergency button, the electricity to the track is considered to be shut off which makes the trains gradually slow down and comes to rest to the immediate next station.



|  |  |  |
| --- | --- | --- |
| *Section ID* | *length (m)* | *max speed (km/hr)* |
| S1-1 | 500 | 15 |
| S1-2 | 500 | 15 |
| S1-3 | 500 | 15 |
| S1-4 | 500 | 15 |
| S2-1 | 100 | 100 |
| S2-2 | 100 | 100 |
| S2-3 | 100 | 100 |
| S2-4 | 100 | 100 |
| S3-1 | 1500 | 150 |
| S3-2 | 1500 | 150 |
| S4-1 | 1800 | 150 |
| S4-2 | 1800 | 150 |
| S5-1 | 500 | 100 |
| S5-2 | 500 | 100 |
| S6-1 | 2000 | 130 |
| S6-2 | 2000 | 130 |
| S7-1 | 2500 | 80 |
| S8-1 | 1000 | 120 |
| S8-2 | 1000 | 120 |
| S9-1 | 1500 | 130 |
| S10-1 | 500 | 100 |
| S10-2 | 500 | 100 |
| S11-1 | 1600 | 130 |
| S12-1 | 1950 | 110 |
| S13-1 | 400 | 75 |
| S14-1 | 400 | 75 |
| S15-1 | 1500 | 80 |
| S16-1 | 16000 | 70 |
| S17-1 | 400 | 65 |
| S18-1 | 500 | 85 |
| S19-1 | 500 | 100 |
| S19-2 | 500 | 100 |
| S20-1 | 1250 | 85 |
| S21-1 | 2300 | 95 |
| S22-1 | 2500 | 100 |
| S22-2 | 2500 | 100 |
| S23-1 | 1700 | 85 |
| S24-1 | 50 | 85 |
| S25-1 | 500 | 15 |
| S25-2 | 500 | 100 |
| S25-3 | 500 | 100 |
| S26-1 | 2450 | 75 |
| S27-1 | 3670 | 65 |
| S28-1 | 1000 | 90 |
| S29-1 | 500 | 15 |
| S29-2 | 500 | 15 |

### Section details

1. As specified the entire track is divided into multiple sections. Each section can be of any length. There is a speed limit for each and every section. When the train is about to enter a section, it gets the information about the speed limit of that section. If the speed limit of the next section is less than the current section the train reduces its speed to the speed limit of that section prior to entering that section. If the speed limit of the next section is more than the speed limit of the current section, the train gradually reduces its speed until it reaches the maximum speed limit of the next section. After train enters the next section the speed increases gradually. If the speed limit exceeds, the user should stop the train and rectify the issue and resume. If speed limit of all trains get exceeded beyond the specified limit by 10% of the limit, then the train derails which is an emergency situation.



#### User Requirements

In the map, the maximum speed of a particular sections is specified along with the speed of the train.

#### System requirements

System should be able to display both the max speed of a section and current speed of the train in the dispatcher system.

### Train position

1. Each section is considered to have sensors. These sensors are used to indicate if a particular train started at a particular section or it crossed the current section. From this information the position of the train is identified. There is internal manual calculation identifying the position of the train with the inputs of the speed of a particular section, distance the train travelled and time taken by the train. These both positions are compared with each other to identify if the system is working fine.

#### User Requirements

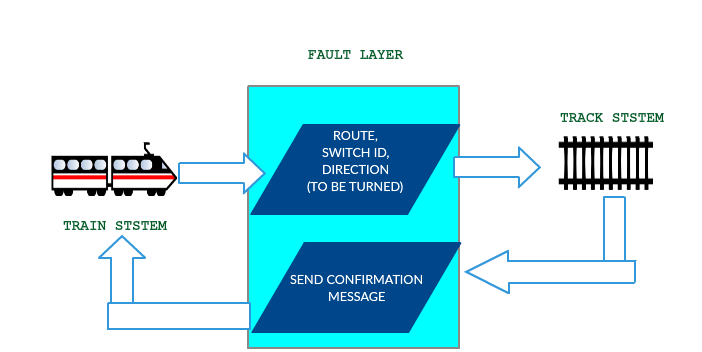
User just observes the behavior of the system on the subway map. x

#### System Requirements

Train system and track system will communicate with each other.

### Switch direction

1. Track system also consists of the switches. Switches are used to change the direction of the train. Switch takes up to 10 seconds to respond to the request change. Switch needs 1 or 2 seconds to change its direction, during this change the switch is neither in not changed or completely changed. When a train is about to reach a switch, prior to that the train should confirm the direction of the switch. This direction should be compared with that of the route in which it should actually travel as the track system already blocks the certain sections of the track for that particular train. This information about the direction of the switch should be verified thrice to ensure that direction of the switch is set in direction and also for the right path to reach the destination. Each switch also have its speed limit with in which the train should travel. If the switch doesn’t move the train should gradually reduce its speed and come to rest. When a train is over a switch, then the direction of the switch cannot be changed.



#### User Requirements

If the information obtained from the switch and the direction the train should travel didn’t match, this should be intimated to the dispatcher to stop the train.

#### System Requirements

Response from the switch should be immediate. It should again request for the same to make sure both are the same. The obtained response from the switch should be compared with the direction the train should travel and then the train should proceed.

### Get Routes and Block Route

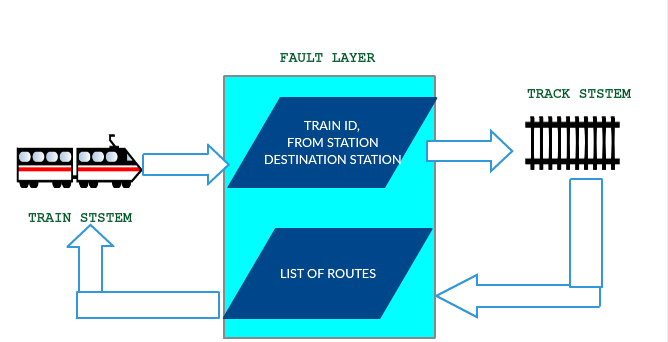
1. Trains request tracks for available routes to its destination. Track system sends back a list of available routes. Trains selects the route and sends request to track system to lock it. Track system locks the route for the train.

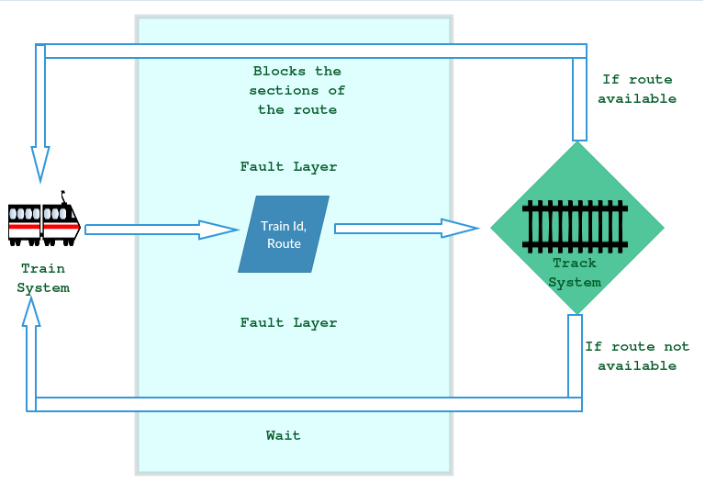
#### User Requirements

There is no involvement of the user in this scenario of blocking route to reach a particular destination.

#### System Requirements

Train system and Track system communicates with each other to transfer information between the systems.





## Dispatcher System:

1. The dispatcher system is considered to be the most important system of the entire subway system. This is the system which the dispatcher mainly watches continuously for the working of the system. This displays a map including the stations and the position of the trains. This system also includes other important information about the system. This system have the emergency stop button to stop the entire system. If the dispatcher needs any other information in more detail he can be able traverse to that particular system for the information and should get back to this system while system is working. It has one more functionality that is start button. It is used to start the train, by using this button we can start our train at any station.

### Live Map

1. This is the core functionality of the system as well as the entire system. There is a map in the interface. This map have the important information of trains and tracks. This maps shows the information about each train regarding its position, speed and direction.

#### User Requirements

Dispatcher watches the system continuously to ensure that the entire system is working perfectly. If there is any error the dispatcher should stop the system.

#### System Requirements

All components of the entire system will involve to update the live map. Any defect in any part of the system will be displayed here with the help of a dialog box.

### Integration

1. This system can be considered as the integration of both the train system and track system. The information from both the systems is collected and organized for the dispatcher which is more important. The other details can be viewed by the dispatcher whenever necessary.

#### User Requirements

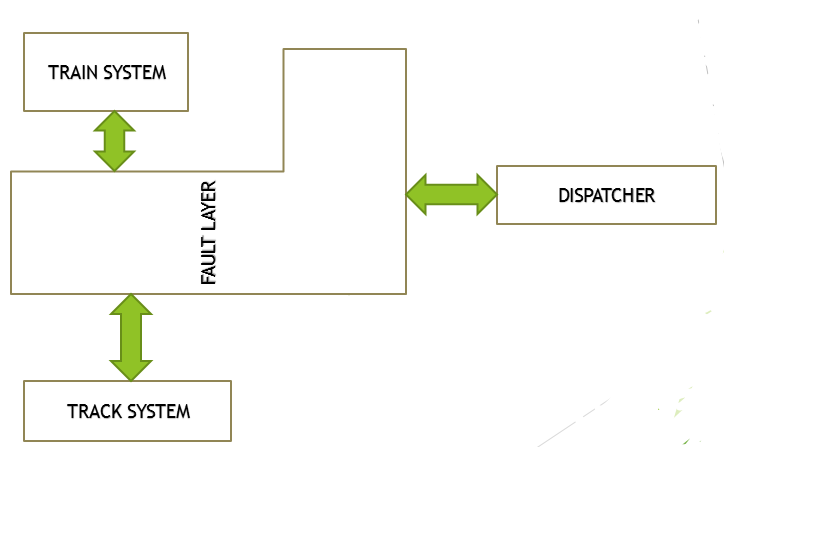
There is nothing expected from the user in this functionality.

#### System Requirements

Both the train system and track system should integrate the data and organize the data as specified in the user interface.

## Fault Layer System:

1. The main purpose of the fault layer system is to act as an intermediate layer between the other systems. Every information either can be a request or response from one system to the other system should be transferred through the fault layer system. This system can incur tolerance. This tolerance can lead to either some error in the information sent or the entire information can be wrong. All the information passed through the fault layer is maintained in a log file. So it is necessary that the system should recheck every information before proceeding.



### Corrupted Message

1. In this functionality, the message transferred from one system to another system is transferred with some errors or changed values or to be more precise wrong data is transferred. The information is passed but there might be an error in the message passed between the systems.

#### User Requirements

Dispatcher should cross check the results before checking the emergency button. There shouldn’t be delay in using emergency button as well.

#### System requirements

System is designed in such a way that only few messages will be corrupted.

### Fail to pass some messages

1. This functionality fails to send the messages. This is occurred a few number of times the messages passes through the fault layer. This failing of message can be occurred to any of the system.

#### User Requirements

Dispatcher should immediately stop the system if there is no message during the working of the system.

#### System Requirements

System is designed in such a way that only a very few messages are considered as the failure messages.

# Non Functional Requirements

## Performance

The system should be able to display all the information about the trains, tracks and the freight train which is useful in emergency situation. It will be used to monitor all the trains and their updates and the tracks so that the trains can move flawlessly. A dispatcher will be always there to monitor all the subsystems and will be able to manage the worst situations. It can stop the trains if there are any interruptions on the track or any disturbances in the trains.

## Reliability

The developed system undergoes the various testing like unit testing, Black box testing, white box testing and integration testing to make sure that system works well in all scenarios. The entire integrated system will be tested by working initially with one passenger train, later with express passenger train individually and then all the passenger trains. After all the passengers trains are working fine the freight train in included into the system followed by fault injection layer.

## Availability

This system will be available to the subway tracks for maintaining the entire subway systems.

## Security

All the user information data can able to fetch from database so there is no chance for fault misrepresentation. The system will be secured in such way that no other third party can manipulate the functionality of the subway system.

## Maintainability

The system can be flexible to add more functionality in future. System can be tested

## Portability

This app is portable among windows, mac and Linux operating systems.