



Concordia Institute for Information Systems Engineering (**CIISE**)

**Team Noble**

# *Transit Electrification Simulation Software*

## **Software Requirements Specification Document**

Date 28/April/2020

Version 2: INSE 6260-SRS-Template

**Version History**

VERSION	AUTHOR(S)	CHANGE DESCRIPTION	DATE	NOTES
1	Daniel	Create	22/Jan/2020	
2	Daniel	Input demo. Output result.	3/Feb/2020	
3	pDaniel	User Interface specification	28/Apr/2020	

## Table of Content

1	Introduction	4
1.1	System Overview	4
1.2	Operating Environment	4
1.3	Conceptual Model	4
2	Functional Requirement	5
2.1	Product Services	5
2.2	User Interface Specifications	6
2.3	External Interfaces and Database Requirements	10
2.4	Error Handling	10
2.5	Foreseeable Functional Changes and Enhancements	10
3	Non-Functional Requirements	10
3.1	Performance Requirements	11
3.2	User Documentation and Other User Aids	11
3.3	Development Requirements	11
3.4	Foreseeable Non-Functional Changes	11
4	Remarks and Guidelines for Later Lifecycle Phases	11
5	Glossary and Index	11

# 1 Introduction

## 1.1 System Overview

We will be developing the software that will help the STM Team of Montreal to plan the electrification of bus route 211 in their system i.e. from Lionel-Groulx to Sainte-Anne/Terminus McDonals. The software will provide the optimal number of buses and chargers at each terminus to be obtained in order to electrify the bus route 211.

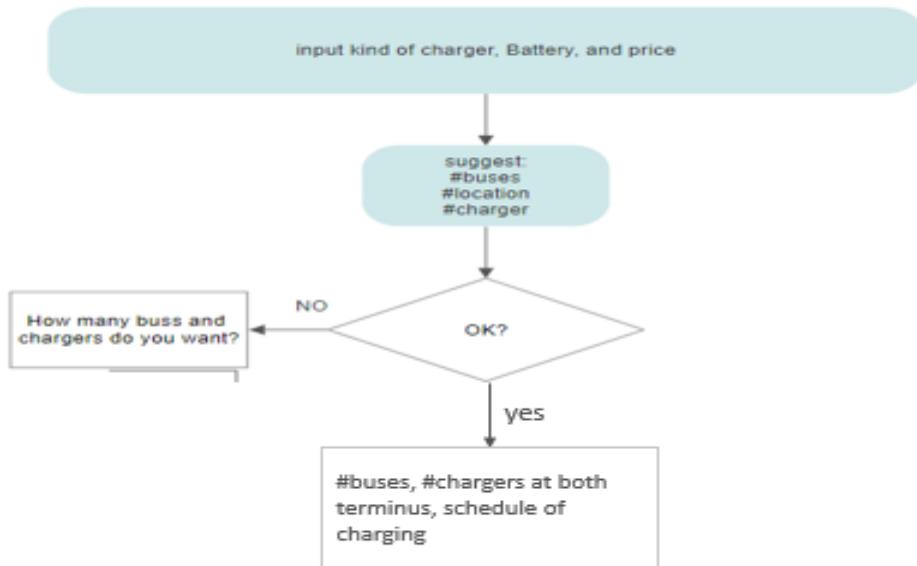
## 1.2 Operating Environment

We will be developing a console-based java software and in order to use it one need to have access to computer/laptop with java installed on it.

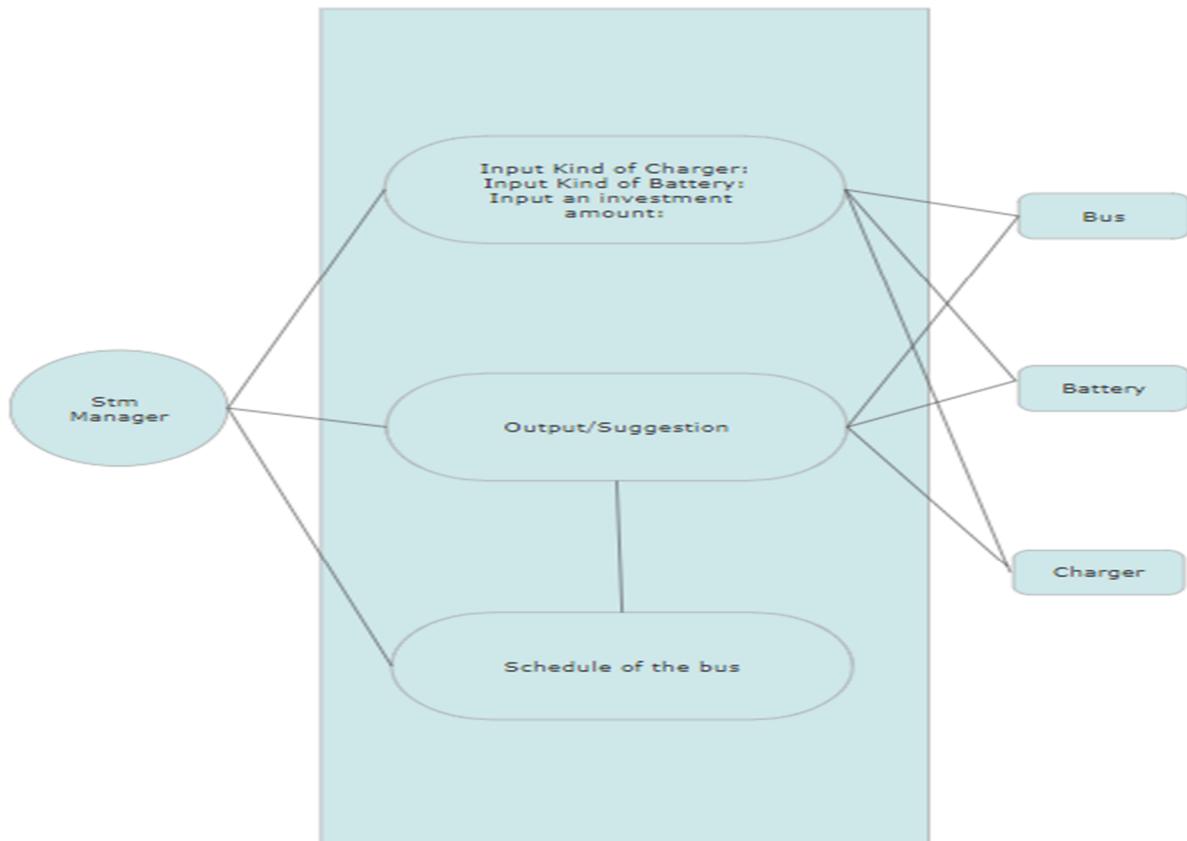
## 1.3 Conceptual Model

Present a high-level graphical view of the major functionalities and/or components of the product and their relationships with each other (preferably using use-cases).

### Activity Diagram



## Use-Case Diagram




---

## 2 Functional Requirement

Our software will be able to perform the functionality as mentioned below:

1. Insert the kind of battery and charger, and price for chargers (slow speed, high speed, bus battery). - User
2. Display the suggestion (# buses, # Chargers at both terminus) - Interface
  - 2.1. Confirmation - User
  - 2.2. Change - User
    - 2.2.1. Input (#buses) - User
3. Create three files for Charging Schedule, Bus Schedule and Chargers - Interface

---

### 2.1 Product Services

1. System/Software prompt to input what kind of battery and charger user want to use.
2. Once user will provide all the necessary input, software will display how many numbers of buses, chargers required in order to create electric transit simulation software.
3. If user is not satisfied with the suggestion of software, he/she can edit it or else, software provide an option to see the bus schedule and charging schedule.

## 2.2 User Interface Specifications

---

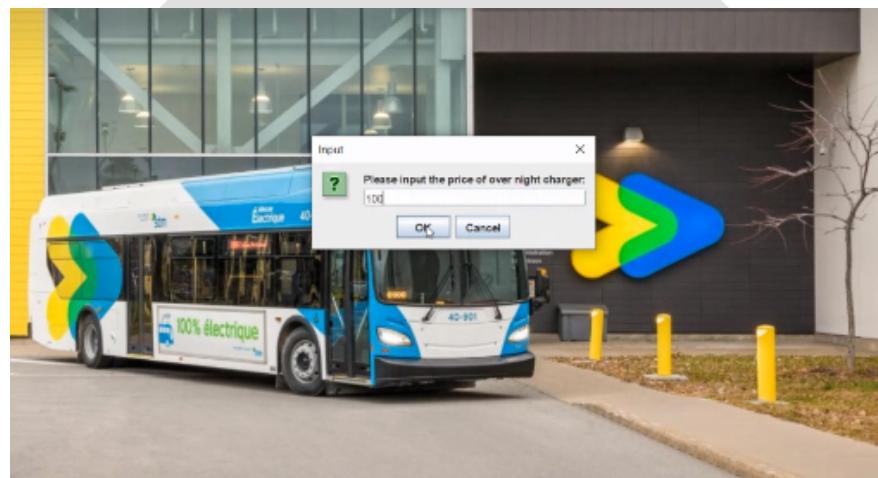
### Display Panel 1 (Input)

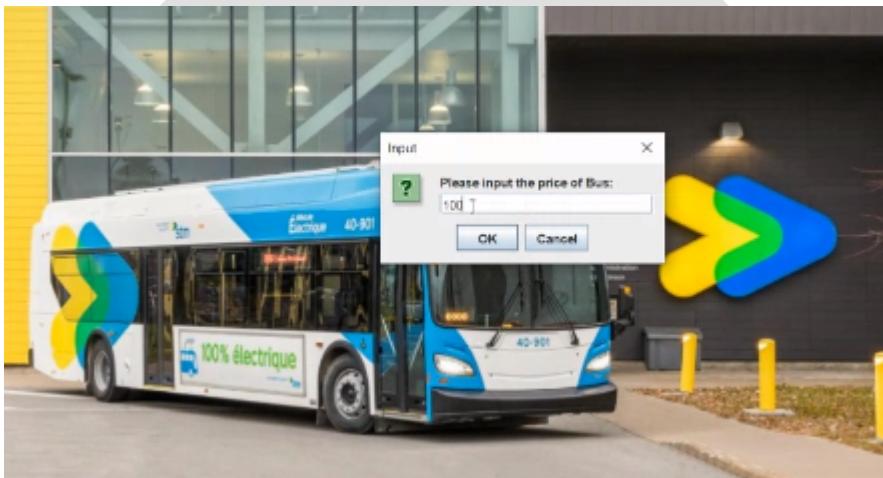
Display panel 1, ask for few inputs from users as given below:

- Please Input the manufacture of charger: ABB/HELIOX
- Please input the price of overnight charger
- Please input the price for fast charger
- Please input the kind of the battery: small(294kwh), big(394kwh)
- Please input the price of bus

The user interface for taking input from user is represented below:

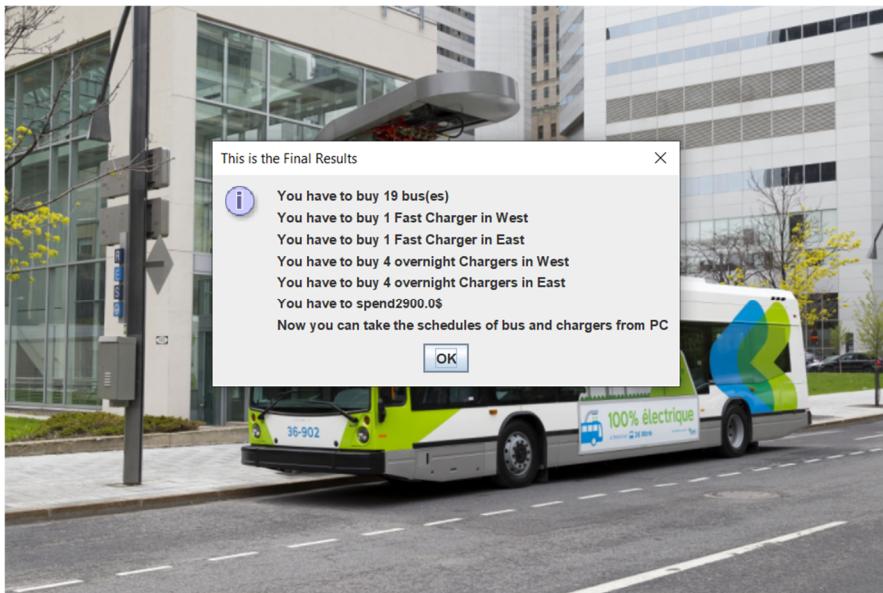






Once user provided all the necessary information, software then displays the result which consist of how many buses, chargers both high speed and slow speed at both the terminus need to buy in order to electrify the bus route along with total amount needs to be invested.

Snippet of the Display Panel 2 (Output) is given below:



Along with this output, software will generate 3 files which provide detailed information about state of charger before the trip and after the trip, schedule of buses, schedule of chargers.

Snippet of these file are given below:

Trip-Comp	Trip Completed		
AT-SOC	After trip battery state of charge		
BT-SOC	Before trip battery state of charge		

Charger ID	Charging Schedule			
	Sart Time	End Time	Bus ID	Next Trip
LG-ON-75-1	2h00	5h00	BE-01	BE01-T1-5-22
MD-ON-75-1	2h00	5h00	BE-02	BE02-T1-5-22

## 2.3 External Interfaces and Database Requirements

---

Will not be using the database.

## 2.4 Error Handling

---

1. If user enters wrong kind of input for the field, the system will display the warning message.

## 2.5 Foreseeable Functional Changes and Enhancements

---

1. Live status of the chargers to be used.
2. Animation of the bus route.
3. Create the web application for the solution.

## 3 Non-Functional Requirements

---

Non-functional requirements describe the constraints under which the deliverables must operate, the environment in which they must operate, any standards they must conform to, etc. These requirements can be categorized using the following subsections.

The requirements can be categorized using the following subsections. All requirements will be filled in the subsections below should use the following scheme:

NFR 1. Availability requirement

NFR 1.1. Users are not allowed to change the route of the bus.

NFR 1.2. Users are not allowed to change the manufacturer details.

NFR 1.3. Users are not allowed to change the charging schedule of the bus.

### **3.1 Performance Requirements**

---

**Response Time** - Processing time of the software will be ranging from 5 seconds to 10 seconds.

**Efficient** - The suggestions provided by the software will be the optimum.

### **3.2 User Documentation and Other User Aids**

---

IEEE standard document.

Wang,C. Product Request-Transit Electrification Simulator.

Wang,C. Product Request-Transit Electrification Simulator 2020-01-30

### **3.3 Development Requirements**

---

Operating system - Windows7 and above

Development Language - java

Development Environment - Eclipse/IntelliJ

### **3.4 Foreseeable Non-Functional Changes**

---

Such changes generally arise from hardware evolution, changing user needs, new systems in the operating environment, etc.

Not Applicable for our project at this moment.

## **4 Remarks and Guidelines for Later Lifecycle Phases**

---

Following activities are planned for later phases of product lifecycle :

1. Product development plan.
2. Once requirements are approved by the STM manager, development of the product will start.
3. Testing Plan

## **5 Glossary and Index**

---

AT- SOC – After trip state of charger

BT-SOC – Before trip state of charger