Table of Content

1 Overview

This document has described the architectural design of the electrification of the STM Route 211 simulation system. The high-level components such as control panel, display panel and their interactions, suitable architectural patterns, the physical arrangement of components and design decisions applied to the whole system.

In addition, it is on the component and detailed design. Includes design pattern, sequence diagrams, class diagrams, database design in detail and user interface design with screenshots of the interface.

1.1 Goals and Objectives

The document will provide developers an insight in meeting client's needs efficiently and effectively. In addition, by providing several views of the system design, this the document facilitates communication and understanding of the system

This document is a customized design document which will be mainly used by developer or coder as an aid to develop a software.

1.2 Statement of Scope

The Software design document demonstrates how the design will accomplish the defined in software Requirement specification (SRS) document such as the functional and non-functional requirements. The document will provide a framework to the programmers by describing the high-level components and architecture, subsystems, interfaces, database design. This is achieved using architectural patterns, design patterns, sequence diagram, class diagram, relational models and user interfaces.

1.3 Terminology & Definitions

1.3.1 Abbreviations

Architectural Design	Establishing the overall structure of software system
Algorithm Design	Specific method to create a mathematical process in solving problem
Class Diagram	A Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects.
Control Panel	To enter the values and according to the inputs the suggested pre- configured plan will be displayed.
Display Panel	This panel will show the charging schedule of the bus on the route 211 which includes information on charging time and usage of the battery during the trip between two terminus.
STM	Société de transport de Montréal. The city transport of Montreal which consists of metros and buses.
Route 211	The buses run between Terminus Lionel-Groulx station and Terminus MacDonald is on Route 211
GUI	Graphical User interface (GUI) is an interface through which the user interacts visually
Sequence Diagram	An interaction Diagram that shows how process interact with one another and in what order
SDS	System Design Specification
SQA	Software Quality Assurance

2 Functional Descriptions

This system provides the information on buses, location of chargers at each terminus, optimal battery size according to charger specification, and prices through selecting the optimal charging models from the manufacturing company. In addition, this system also shows the charging schedule of the buses and can be seen on the webpage.

The functional requirements are as follows:

FD 1.1 Control Panel

- FD 1.1. Selection of manufacturers.
- FD 1.2. Selection of charging models according to manufacturers.
- FD 1.3. Cost input by the user for charger models.
- FD 1.4. Selection of battery for the bus.
- FD 1.5. User input the unit cost of the bus with selected battery option.
- FD 1.6. View / update the number of chargers required on each terminus.
- FD 1.7. View / update the number of buses to be purchased.
- FD 1.8. View / update overall expenditure.

FD 2. Display Module

- FD 3.1. View the charging schedule of the buses.

3 Non-Functional Descriptions

Other than functional descriptions in accordance with the software quality attributes should be taken into consideration like solution quality, flexibility, real time response and reliability. This category describes the usability and other features of the system which can be fulfilled to satisfy the customer.

3.1 Solution quality

The solution quality refers to the after – sales service that customer needs. With the user manual the customer can contact the technical department about the software related issues like errors and defects. If, the service provider can upgrade the versions timely and do the maintenance monthly then customer faces less problem. The software service should be time – oriented and the provider should understand the problem to give an optimal solution. The end – user should never be disappointed by the support system given by the developer after the final product is released.

3.2 Flexibility

This project provides multidisciplinary flexibility concept that is studied in different domains. There are different definitions of flexibility depending on the context and domain application in our project. Across the different definitions of flexibility, it can be summarized as the ability of an entity to cope with changes with:

(1) Effectiveness, timeliness, and satisfaction with a change.

(2) Balance of the amount of change and stability.

(3) Minimal difficulty, cost, time, effort, and risk of the change.

(4) The universality of the entity expected to cope with variations of input.

3.3 Real time responsiveness

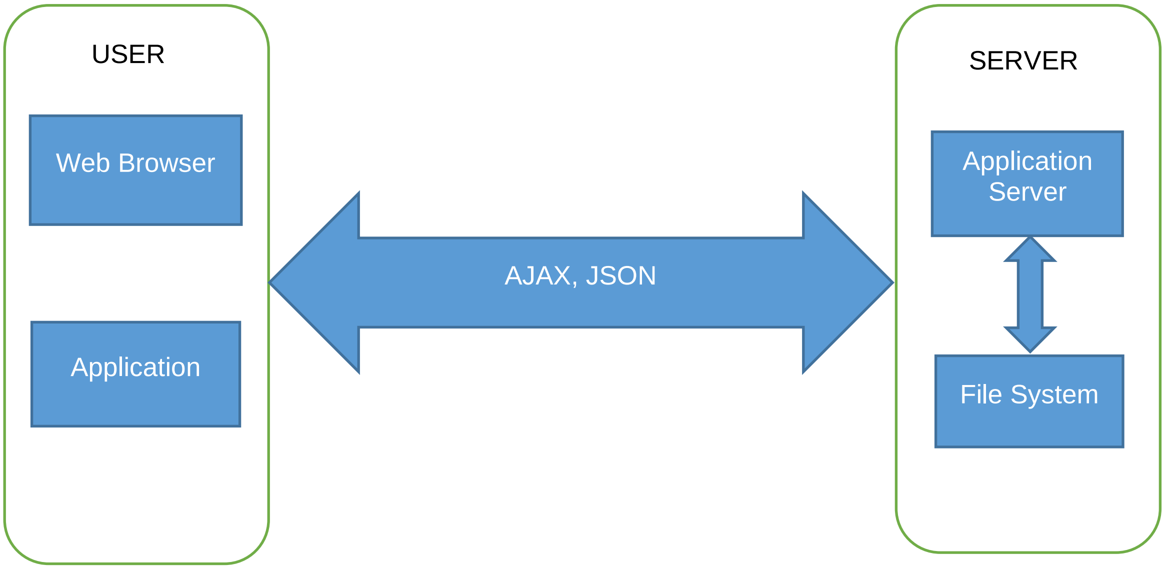
The core quality of any software is the responsiveness when user needs the results in real – time. Software should be developed with the constraint of the system response time like produce any results of given inputs in certain amount of time including calculation time. The simulation software should be lag – free and end – user cannot afford any time delay of the results. If, the simulation software is a web application then redirection between different pages should be quick (Note: Redirection and loading of the pages depends on the speed of the internet connection).

The pages of any web application transmitted via internet connection between the client and the server should be quick and the system of both server and client must upgraded to the latest versions to achieve the quality response time.

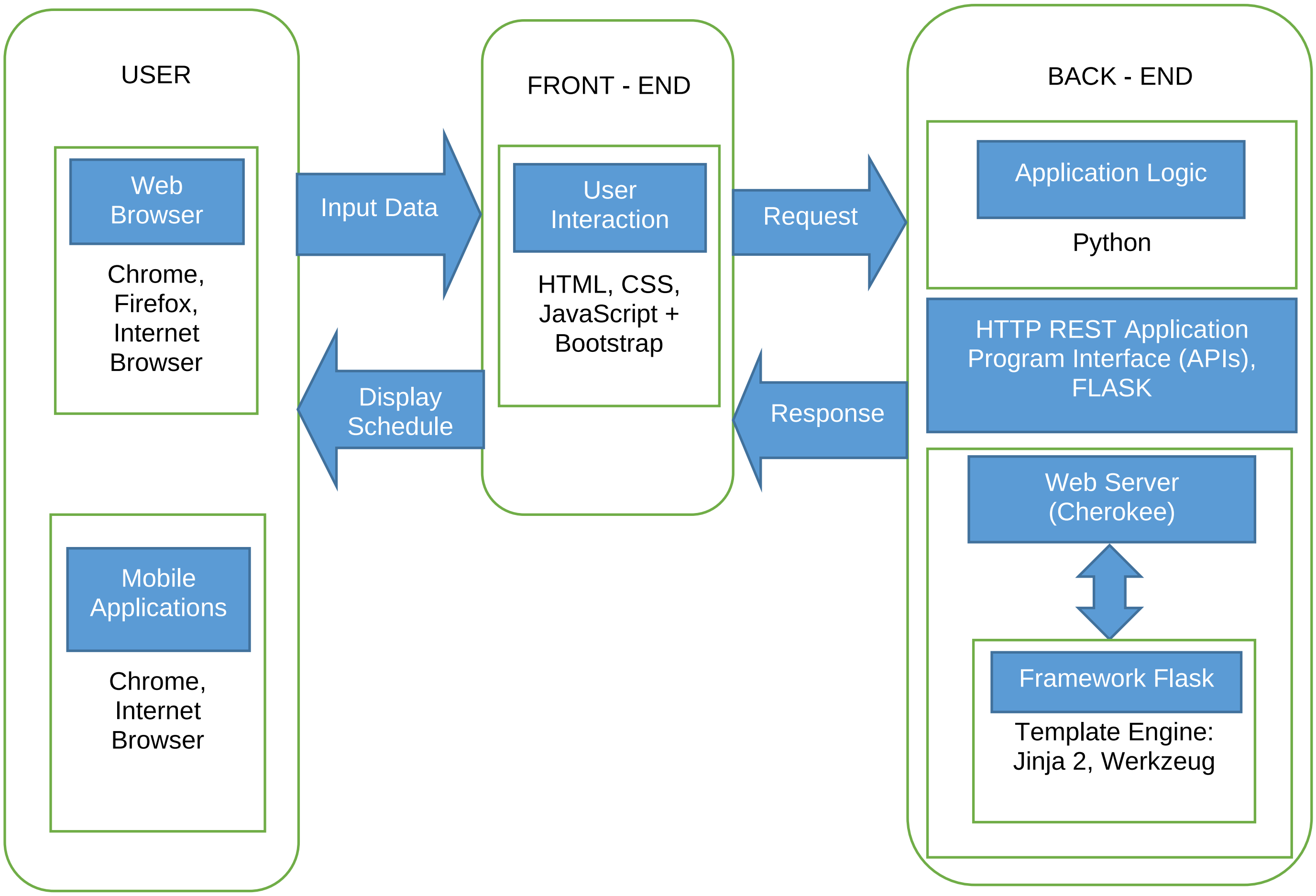
3.4 Reliability

The system should be reliable and produce the correct results according to the given inputs. Reliability requirements are typically part of a technical specifications document. They can be requirements that we set for our simulator or what it reports as its reliability to its customers. The end – user can rely on the data which is produced in the output through given inputs.

4 System Architecture



5 Detailed Architecture



6 Detailed System Design

The detailed design of the system that is developed by the designer and helps the developer/coder of the simulation software. This section depicts the GUI (Graphical User Interface) of the whole system which will be implemented through programming language with essential functions stated in the SRS (Software Requirement Specification) document. The basic layout of the system and the raw functionalities of the software can be seen in the virtual manner.

6.1 Graphic User Interfaces and a Basic Demo Scenario

1. Control Panel

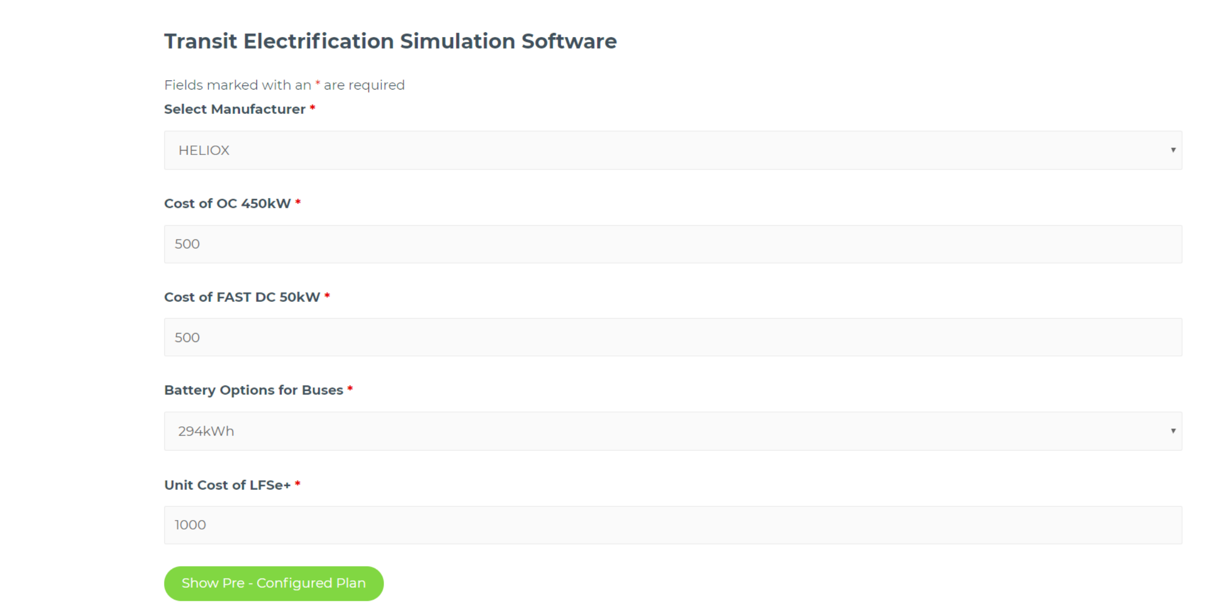


Figure 1 Control Panel

2. Pre – configured Plan

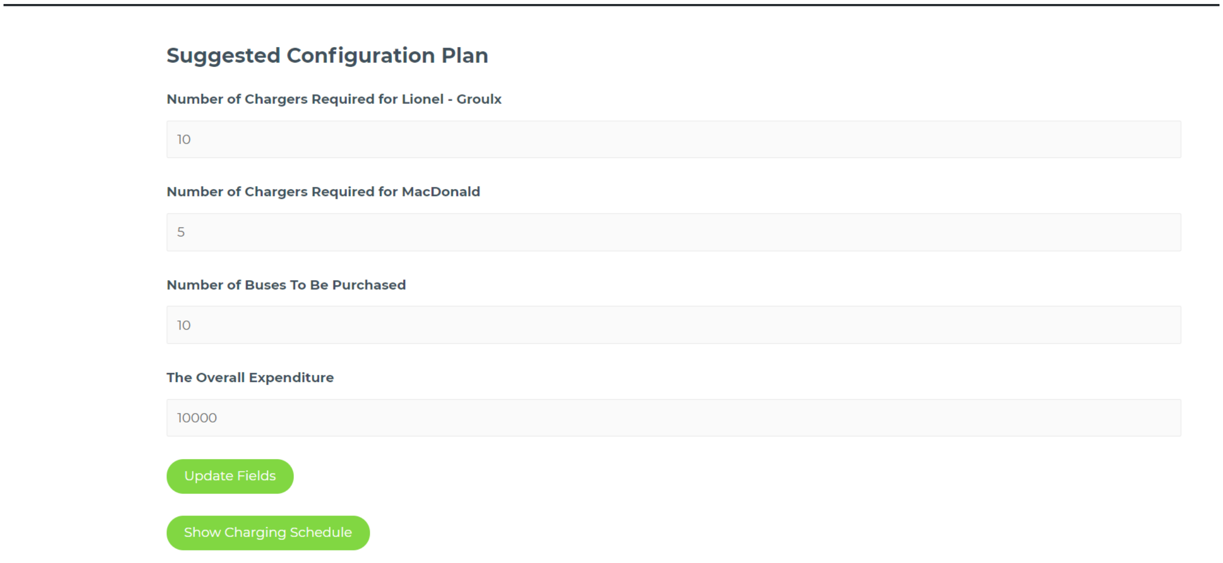


Figure 2 Pre - configured Plan

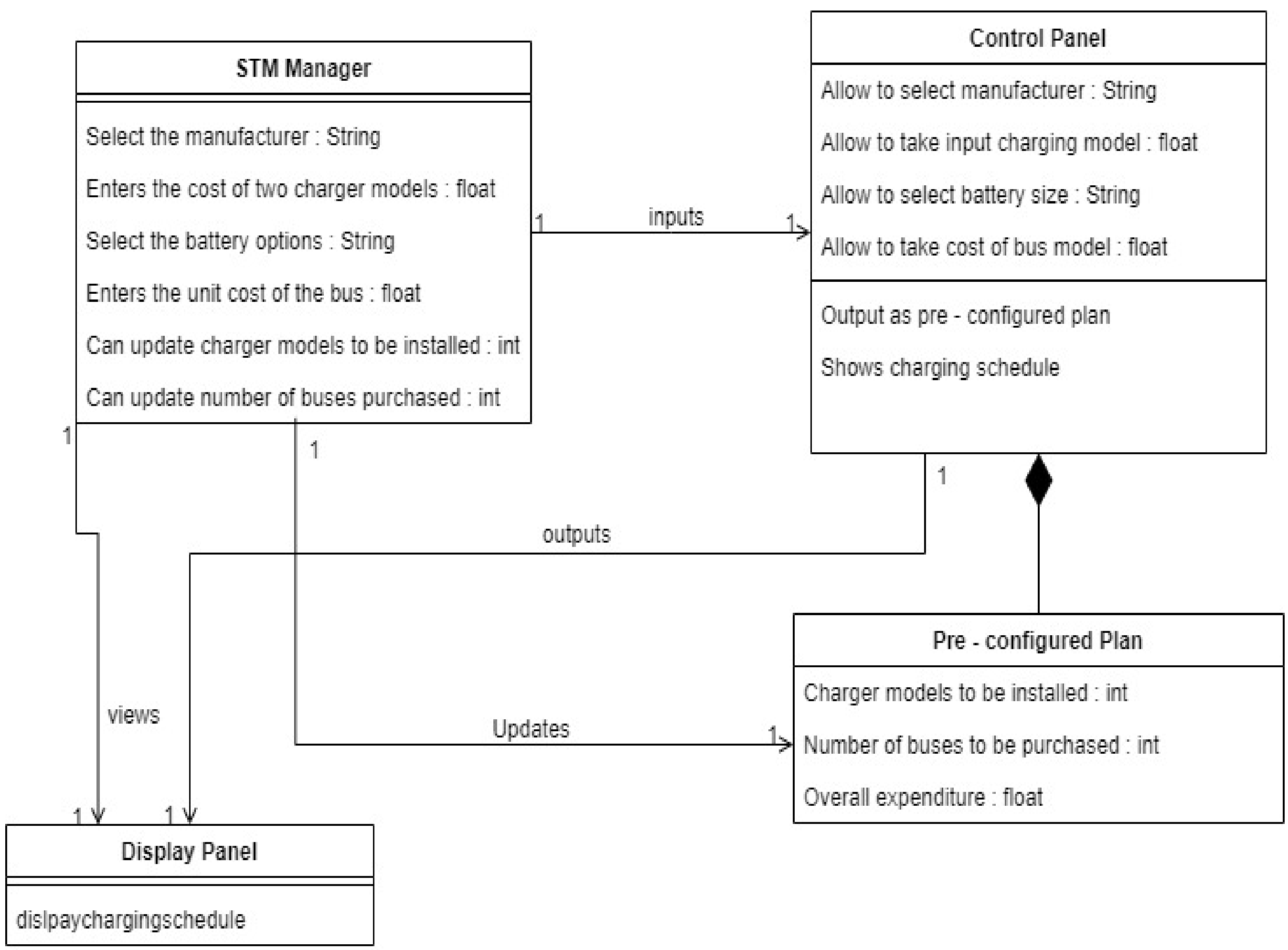
3. Display Schedule

Display Panel

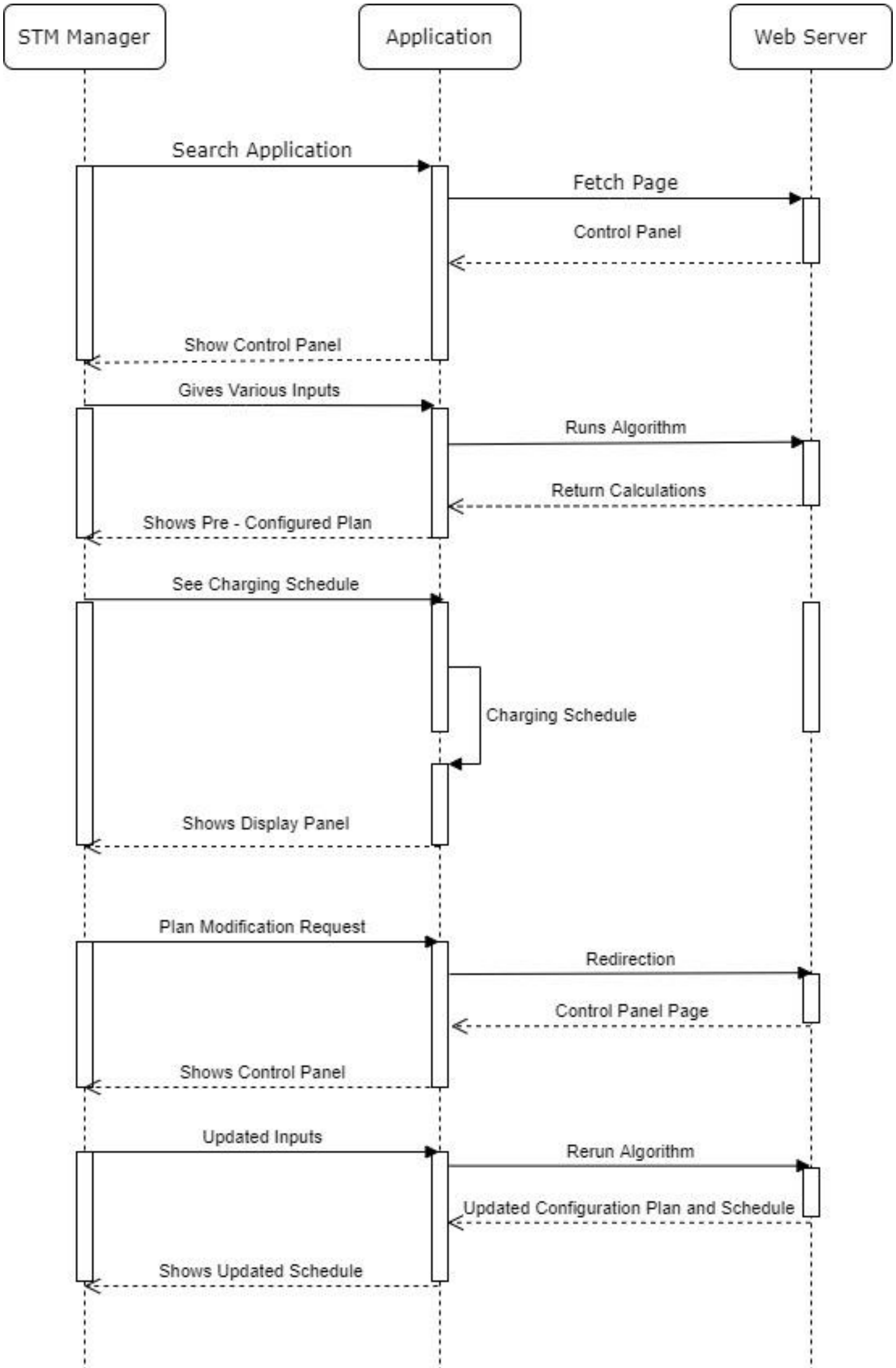
Bus		Trip Schedule		Charging Schedule		
Bus ID	Bat. Size	East Time (E)	West Time (W)	Charger ID	Start Time	End Time
STM - 1	294kWh	4:55	6:00			
		7:01	8:15			
		9:58	11:18			
		12:37	13:43	W - DA - 1	13:43	13:47
		15:09	16:35			
		17:37	18:53			
		20:11	21:16			
		22:42	23:59	E - ON - 1	0:59	2:59
STM - 2	294kWh	6:28	5:22			
		10:21	8:44			
		12:58	11:49			
		15:38	14:11	E - DA - 1	15:38	15:42

Figure 3 Display Panel

6.2 Class Diagram



6.3 Sequence Diagram



6.4 Suggestions for Web Service Integration

The simulation software can be integrated with different web services and APIs. The whole simulation software becomes dynamic through the integration with the server which runs latest API technology. However, the application calls different APIs and gives real - time results. Through the integration with different web services the system gets live schedules of different buses and charging stations. If the whole platform shifted to the cloud SaaS (Software as a Service) which provides all the functionalities of the simulation software accessible from anywhere in the world by all the consumers of the software just with their credentials including live internet connection.