Software Requirements Specification

for the

SURTS

System for Uniform Route-Based

Transportation Simulation

Version 2.0 approved

Prepared by Group A41

CSCI 5801

11.09.2020

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Austin Wessel	10.07.2020	Original Req Specification	1.0
Wessel, et al	10.22.2020	Update and Aggregate Requirements	1.1
Wessel, et al	11.09.2020	Update Reqs, and Models, add Specs	2.0

1. Introduction

1.1 Purpose

Develop a system that will simulate a uniform route-based transportation system. Because transportation systems handle many diverse populations in motion, developing transit routes that meet the needs of the users while keeping costs low is a challenge. This simulation will be used as a testing environment to determine optimal routes, fleet usage, and profitability.

1.2 Document Conventions

Pay close attention to items that are **bolded** as they are significant. It is also to be noted that priorities for higher-level requirements are assumed to be inherited by detailed requirements.

1.3 Intended Audience and Reading Suggestions

This document is intended for developers, project managers, marketing staff, users, testers, documentation writers, and all other stakeholders in the development of this system. The rest of this document contains all the information required to build the system to the standard agreed upon by all parties. It is best to read through each section sequentially. If there is a specific section that needs to be reviewed, please find it in the table of contents.

1.4 Product Scope

The software specified in this document is to be used by transportation system staff, urban planners, statisticians, economists, and system administrators to develop a more optimal transportation system. By controlling different attributes of the simulation like traffic conditions and bus capacity, the users can test different scenarios and make decisions about the actual transit system based on that information. This system hopes to serve the transportation system in providing better insights to their staff who will then make changes to better serve their customers.

1.5 References

IEEE Recommended Practice for Software Requirements Specifications," in *IEEE Std 830-1998* , vol., no., pp.1-40, 20 Oct. 1998, doi: 10.1109/IEEESTD.1998.88286. (1)
Parking and Transportation Services and Office of Measurement Services at University of Minnesota. (2015). Campus Traveler Mode Share and Origin Destination Study 2014 Report. Retrieved from the University of Minnesota Digital Conservancy, http://hdl.handle.net/11299/171518. (2)
Elicitation Session with Joseph Dahep.

2. Overall Description

2.1 Product Perspective

The product defined in this SRS document is the first system in a family of systems to be developed for transportation companies. This will be a self-contained product, but future iterations will use the successes and failures of this specific system in the retrospect.

2.2 Product Functions

- Simulate buses transporting passengers from an origin to a destination
- Generate buses at a user-defined interval
- Generate passengers needing transportation in a configurable matter
- Configure routes for the buses under simulation
- Produce logs detailing a variable amount of information about the simulation run
- Allow users to inspect logs
- Interface with external systems by providing an externally accessible data service
- Retain logging information for a set period of time
- Provide an interface for the modification of system attributes
- Produce reports summarizing simulation results

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

2.4 Operating Environment

TBD

2.5 Design and Implementation Constraints

TBD

2.6 User Documentation

TBD

2.7 Assumptions and Dependencies

A key assumption in defining the scope of this system is that only a single simulation is run at a time.

3. External Interface Requirements

3.1 User Interfaces

TBD

3.2 Hardware Interfaces

TBD

3.3 Software Interfaces

TBD

3.4 Communications Interfaces

TBD

4. Use Cases

4.1 Simulate Buses and Routes

Iteration: 1.0

Summary: The user configures a route and simulates buses transporting passengers from an origin to a destination.

Basic Course of Events:

1. User selects a pre-configured route.
2. User starts the simulation.
3. Simulation generates buses and passengers in a default matter

Alternative Paths: In step 1, the user may choose to configure a custom route. The system will still generate buses and passengers based on the default setting.

Exception Paths: None

Extension Points: None

Trigger: User wants to begin the simulation.

Assumptions: User is familiar with how to operate the system and wants to simulate buses.

Precondition: None.

Postcondition: The simulation is running, and the user can now continue to observe, configure, and generate reports.

Author: Austin Wessel

Date: 10.22.2020

4.2 Produce Logs and Generate Reports

Iteration: TBD

Summary: A log is created upon termination of each simulation that a user can either inspect or generate a report from.

Basic Course of Events:

1. Upon termination of a simulation, the system produces a log.
2. Users can choose to view the entire log or specific attributes of the log.
3. Log is stored in a memory source TBD.
4. Users can choose to generate a report from a log and obtain the information in a more portable form.

Alternative Paths: None

Exception Paths: None

Extension Points: None

Trigger: User runs a simulation for the purpose of collecting data on that configuration.

Assumptions: User is familiar with how to operate the system. System has enough available storage to store the created log.

Precondition: None.

Postcondition: The log has been produced and can be referenced in the future for additional inspection or reporting.

Author: Austin Wessel

Date: 10.22.2020

4.3 Configure Simulation Parameters

Iteration: 1.0

Summary: The user will configure the attributes of the buses, bus routes, bus types, and passengers.

Basic Course of Events: The user will interact with the interface to modify the bus routes data, bus stops, number of buses, amounts of passengers, bus capacity, number of buses over-capacity, which buses are overcapacity, and where they are overcapacity.

Alternative Paths: None

Exception Paths: Show an error message if one of the inputs is outside of the constraints for the input

Extension Points: None

Trigger: User will use UI to trigger configuration

Assumptions: User is familiar with how to operate the system.

Precondition: None.

Postcondition: The system parameters have been configured and are ready to use.

Author: Connor Boehm

Date: 10.22.2020

4.4 Run/End Simulation

Iteration: 1.0

Summary: Running and ending the simulation successfully: the user will run the simulation and eventually be able to stop it.

Basic Course of Events: The user will start the simulation. They then let it run for an amount of time, then end the simulation if they want.

Alternative Paths: None

Exception Paths: The Simulation fails

Extension Points: None

Trigger: User hits start
Assumptions: User is familiar with the system and if the system cannot be run if it is already running.

Precondition: The system is configured correctly.

Postcondition: The log and report are generated, the simulation is no longer running, and the UI is reset in a way that the user is able to restart the simulation when ready.

Author: Connor Boehm

Date: 10.22.2020

5. User Requirements

5.1 Simulate Buses and Passengers

5.1.1 Simulate Buses

REQ: 1 **Requirement Title:** Simulate Buses **Use Case:** 1

Date: 10.22.2020

Introduction: Generate buses at a user defined interval and of the type specified by the user.

Rationale: Users need to add buses to a simulation to transport passengers.

Inputs: Type of bus, time interval, route, bus generation point(default to bus depot)

Requirement Description: Select a bus from those previously defined and add it to the selected route in the current simulation. Additionally, may set an interval to automatically add buses.

Outputs: The selected bus is added to the simulation at the generation point and travels the selected route.

Persistent Changes: None

Related Requirements: 2,3

Conflicts: None

Support Material: None

Test Cases: 5.2.1, 5.2.2

i. 5.1.2 Simulate Passengers

REQ: 2 **Requirement Title:** Simulate Passengers **Use Case:** 1

Date: 10.22.2020

Introduction: Simulate passengers at a user defined level with attributes specified by the user.

Rationale: Passengers are required for a transportation simulation to have meaning. These passengers must be able to be moved in the system in accordance with buses.

Inputs: Origin, Destination, Time of arrival.

Requirement Description: The simulated Buses must be able to pick up generated passengers from their bus stop origin, and transport them to their destination.

Outputs: Passenger information in the form of a log.

Persistent Changes: System Passengers

Related Requirements: 1, 7

Conflicts:

Support Material: None

Test Cases: 5.3.1

5.2 Generate Buses

5.2.1 Create and Configure Buses

REQ: 3 **Requirement Title:** Create Buses **Use Case:** 1

Date: 10.22.2020

Introduction: Allow the user to create different types of buses. These buses can then be saved to use in future simulation runs.

Rationale: This feature will allow users to add buses matching the specifications of those currently in their fleet. It will also allow them to experiment with different model buses they could purchase in the future.

Inputs: Maximum capacity, Length, Fuel consumption
Requirement Description: Create buses with user defined attributes and save them.

Outputs: The newly created bus added to the simulation's bus repository.

Persistent Changes: System buses

Related Requirements: 1, 4

Conflicts: Needs to be validated via error checking (Validate Bus Configuration).

Support Material: None

Test Cases: 5.2.4, 5.2.3

ii. 5.2.2 Delete Buses

REQ: 4 **Requirement Title:** Delete Buses **Use Case:** 1

Date: 10.22.2020

Introduction: Allow the user to delete buses from the simulation's bus repository.

Rationale: This feature will allow users to remove buses from the simulation's scope. Deleted buses will not be available for use in future simulations.

Inputs: Indication to delete bus through user Input

Requirement Description: Delete buses from the simulation's bus repository.

Outputs: Success or Failure.

Persistent Changes: System buses

Related Requirements: 3

Conflicts: None

Support Material: None

Test Cases: 5.2.5, 5.2.6

b. 5.3 Generate Passengers

i. 5.3.1 Create Passengers

REQ: 5 **Requirement Title:** Generate Passengers **Use Case:** 1

Date: 10.22.2020

Introduction: For this system to simulate passengers, said passengers must be generated. This is done by the system.

Rationale: Generating passengers is the first step in simulating passengers, which is a prime functionality of this system.

Inputs: Passenger origin bus stop, passenger destination bus stop, time of arrival at initial origin.

Requirement Description: The system creates a passenger based on given inputs and adds it to the system's passenger repository.

Outputs: The created Passenger.

Persistent Changes: Simulation passengers

Related Requirements: 1

Conflicts:

Support Material: None

Test Cases: 5.3.1

ii. 5.3.2 Passenger Historical Data

REQ: 6 **Requirement Title:** Upload Historical Passenger Data **Use Case:** 1

Date: 10.22.2020

Introduction: Allow the user to enter data collected by the transportation system employees in the past. This includes information regarding how many passengers are at each stop, the final destination of most passengers, which hours have the most passengers, etc.

Rationale: The transportation system employees are already knowledgeable in this area. This would allow them to create a basis of a passenger pattern for the simulation to build upon.

Inputs: Passenger count per hour for each day of the week, correlation between temperature and passenger counts, start point popularity, destination popularity

Requirement Description: Utilize data collected by transportation system employees prior to simulation implementation. This data will be used as a basis to build the simulation on.

Outputs: None

Persistent Changes: System passenger data

Related Requirements: 3

Conflicts: None

Support Material: None

Test Cases: 5.3.2

iii. 5.3.3 Create Passenger Pattern

REQ: 7 **Requirement Title:** Create and Save Passenger Pattern **Use Case:** 1

Date: 10.22.2020

Introduction: To have an accurate representation of the frequency of passenger arrival at bus stops, a system to define this frequency must be in place.

Rationale: Passenger arrival frequency will be different at each stop based on outstanding circumstances. For the simulation to represent these differing frequencies, the pattern will be in place.

Inputs: Passenger arrival frequencies per bus stop based on time of day.

Requirement Description: Passenger generation in the system will be tied to a passenger pattern. This pattern defines the rate of passenger generation at each bus stop for any time of day during operating hours. The system must be capable of creating these passenger patterns based on the above input and saving these patterns for later simulation.

Outputs: The created passenger pattern

Persistent Changes: System passenger patterns

Related Requirements: 3

Conflicts: None

Support Material: None

Test Cases: 5.3.3

iv. 5.3.4 Delete Passenger Pattern

REQ: 10 **Requirement Title:** Delete Passenger Pattern **Use Case:** 1

Date: 10.22.2020

Introduction: Passenger patterns define the frequency of passenger generation. The system must be capable of removing these patterns from the system.

Rationale: Depending on the longevity and implementation of SURTS, the saved passenger patterns may either become absolute or not warrant the space they are occupying. The system then needs to free memory or organization space by deleting these patterns.

Inputs: Passenger pattern to be deleted

Requirement Description: Any given passenger pattern must be capable of being removed from the system.

Outputs: Success or Failure

Persistent Changes: System Passenger patterns

Related Requirements: 3

Conflicts: None

Support Material: None

Test Cases: 5.3.4

c. 5.3 Generate Routes

i. 5.3.1 Create and Configure Routes

REQ: 11 **Requirement Title:** Create/Configure Route **Use Case:** 1

Date: 10.22.2020

Introduction: The user creates or configures a route and to allow the system to simulate buses transporting passengers from an origin to a destination.

Rationale: For the system to accurately simulate a bus traversing the UMN campus, it must have a defined route to take. The system must be able to save these user created/configured bus routes.

Inputs: Bus stops in order of traversal.

Requirement Description: The system must be able to create/configure a bus route based on user input. This route is composed of bus stops that exist within the system. This route is then saved within the system.

Outputs: None

Persistent Changes: Routes in system

Related Requirements: 3

Conflicts: None

Support Material: None

Test Cases: 5.3.5

ii. 5.3.2 Delete Routes

REQ: 12 **Requirement Title:** Delete Route **Use Case:** 1

Date: 10.22.2020

Introduction: Routes are used by buses to accurately simulate a traversal throughout the UMN campus bus stops. These routes must have a method of deletion from the system.

Rationale: In the case where a system implementation has a long lifespan, routes may become obsolete and serve no purpose. To eliminate wasted memory and increase organization, these retired routes must be capable of deletion.

Inputs: Route to be deleted.

Requirement Description: Any route in the system must be capable of deletion. This deletion takes an input from the user defining said route to be deleted. The system then removes this route from the system.

Outputs: Success or Failure

Persistent Changes: Routes in system

Related Requirements: 3

Conflicts: None

Support Material: None

Test Cases: E5.3.6

d. 5.4 Generate Logs

i. 5.4.1 Create Log

REQ: 13 **Requirement Title:** Create Log **Use Case:** 4.1, 4.4

Date: 10.22.2020

Introduction: The system should create a file and store data collected during runtime.

Rationale: The log is to be created in order for it to be evaluated later.

Inputs: Bus routes data, bus stops, number of buses, location of buses, locations of passengers, amounts of passengers, bus capacity, number of buses over-capacity, which buses are overcapacity, and where they are overcapacity.

Requirement Description: Create the log using the data collected during the simulation and place it in a file

Outputs: Generated Log

Persistent Changes: Logs

Related Requirements: 14, 15, 16

Conflicts: None

Support Material: None

Test Cases: 5.4.1, 5.4.2

ii. 5.4.2 Store Log

REQ: 14 **Requirement Title:** Store Log **Use Case:** 4.1

Date: 10.22.2020

Introduction: Create a way for the system to save a log produced by the simulation.

Rationale: The user may want to reference the log after they exit the program, and this would provide them with the means to do that.

Inputs: Previous log, memory storage

Requirement Description: Save a log to a memory location

Outputs: Stored log in a memory location

Persistent Changes: Logs in memory

Related Requirements: 13

Conflicts: None

Support Material: None

Test Cases: 5.4.1, 5.4.3

iii. 5.4.3 Inspect Log

REQ: 15 **Requirement Title:** Inspect Log **Use Case:** 4.1

Date: 10.22.2020

Introduction: The program should provide a means to view the log.

Rationale: It will allow the user to be able to look at the log while still in the program, so allows for ease of access.

Inputs: Previously produced log

Requirement Description: The program will allow the user to view the logs

Outputs: Viewable log

Persistent Changes: Logs in system

Related Requirements: 13

Conflicts: None

Support Material: None

Test Cases: 5.4.4, 5.4.5

iv. 5.4.4 Delete Log

REQ: 16 **Requirement Title:** Delete Log **Use Case:** 4.1

Date: 10.22.2020

Introduction: The system should allow for a way to delete a previously saved log.

Rationale: The user should be able to delete logs they no longer want to keep freeing up memory space.

Inputs: Previously saved log

Requirement Description: Allow for a means to delete the log from the memory device.

Outputs: Freed memory space from removing the log

Persistent Changes: None

Related Requirements: 13, 14

Conflicts: None

Support Material: None

Test Cases: 5.4.6, 5.4.7

e. 5.5 Generate Reports

i. 5.5.1 Generate Report from Simulation

REQ: 17 **Requirement Title:** Generate Report from Simulation **Use Case:** 1

Date: 10.22.2020

Introduction: After the simulation has ended, generate a summary report.

Rationale: A summary report of the simulation will allow the user to look over the results of the simulation, and other prior simulations that were run on the system.

Inputs: None.

Requirement Description: Upon completion of the simulation, the system will generate a report that summarizes what the bus utilization was and what were wait times for passengers during the simulation.

Outputs: A report will be stored within the system.

Persistent Changes: The report will be stored and persistent within the system.

Related Requirements: 1

Dependency: A simulation must have run and completed on the system.

Conflicts: None

Support Material: None

Test Cases: E.5.5.1

ii. 5.5.2 Store Report

REQ: 18 **Requirement Title:** Save Report **Use Case:** 1

Date: 10.22.2020

Introduction: Allow a user to save a report outside of the system.

Rationale: Users need to be able to store reports for later reference, or to be able to share reports outside of the system.

Inputs: Selected report, save destination

Requirement Description: Save reports for future inspection.

Outputs: A report of a user-configurable file type stored at a user specified location.

Persistent Changes: file stored outside of the system.

Related Requirements: 1

Conflicts: None

Support Material: None

Test Cases: E.5.5.3

iii. 5.5.3 Delete Report

REQ: 19 **Requirement Title:** Delete Report **Use Case:** 1

Date: 10.22.2020

Introduction: Allow users to delete reports.

Rationale: Users may choose to delete reports after a period of time.

Inputs: Selected report

Requirement Description: User will be able to delete a previously saved report from memory.

Outputs: Memory recovered from deleted report.

Persistent Changes: System reports

Related Requirements: 1

Conflicts: None

Support Material: None

Test Cases: E.5.5.4

6. Specifications

SPEC: 1 **Title:** Simulate Buses **Use Case:** 1

Date: 11.9.2020

Introduction: During the course of the simulation, the system must record data into logs, place them into a log.

Rationale: Users need to add buses to a simulation to transport passengers.

Specification Description:
Select a bus from those previously defined and add it to the selected route in the current simulation. Additionally, may set an interval to automatically add buses.

1. Prompt the user to select a bus of a defined bus type.
2. User selects the route from the set of available bus routes.
3. Allow option for the user to select an interval for the program to automatically add buses.
4. The system will then add the bus to the simulation with the specs that have been chosen.

Preconditions: The simulation is running

Postconditions: The buses are added and simulated

Related Requirements: 1,2,3

Assumptions: Create Bus, Create Passenger, Bus Route is functioning correctly

Author: Connor Boehm

SPEC: 2 **Title:** Create Bus types **Use Case:** 1

Date: 11.9.2020

Introduction: Allow the user to create different types of buses. These buses can then be saved to use in future simulation runs. During the course of the simulation, the system must record data into logs, place them into a log
.

Rationale: This feature will allow users to add buses matching the specifications of those currently in their fleet. It will also allow them to experiment with different model buses they could purchase in the future.

Specification Description:
The user will have to input the following to add a bus, within constraints (input: [min, max]:

1. Maximum capacity -- inputs [0, 10000], integer
2. Length -- input [0, 1000]
3. Fuel consumption -- input [0, 1000]
4. Bus Type Name -- input: String length < 1000 characters

If any one of the inputs is beyond the constraints, raise an error and display some sort of error message to prompt the user to fix the issue.

The bus type will then be stored in memory where it can be used later.

Preconditions: The simulation is running

Postconditions: The buses are added and simulated

Related Requirements: 3

Assumptions:

Author: Connor Boehm

SPEC: 3 **Title:** Create Bus types **Use Case:** 1

Date: 11.9.2020

Introduction: Allow the user to create different types of buses. These buses can then be saved to use in future simulation runs. During the course of the simulation, the system must record data into logs, place them into a log
.
Rationale: This feature will allow users to add buses matching the specifications of those currently in their fleet. It will also allow them to experiment with different model buses they could purchase in the future.

Specification Description:
The user will have to input the following to add a bus, within constraints (input: [min, max]:

1. Maximum capacity -- inputs [0, 10000], integer
2. Length -- input [0, 1000]
3. Fuel consumption -- input [0, 1000]
4. Bus Type Name -- input: String length < 1000 characters

If any one of the inputs is beyond the constraints, raise an error and display some sort of error message to prompt the user to fix the issue.

The bus type will then be stored in memory where it can be used later.

Preconditions: The simulation is running

Postconditions: The buses are added and simulated

Related Requirements: 3

Assumptions:

Author: Connor Boehm

SPEC: 4 **Title:** Simulate Passengers **Use Case:** 1

Date: 11.9.2020

Introduction: Simulate passengers at a user defined level with attributes specified by the user.
.
Rationale: Passengers are required for a transportation simulation to have meaning. These passengers must be able to be moved in the system in accordance with buses.

Specification Description:
The user will have to input the following, within constraints (input: [min, max]:

1. Origin -- input: valid bus stop
2. Destination -- input: valid bus stop
3. Time of Arrival --input: valid time input (up to implementation)

If any one of the inputs is beyond the constraints, raise an error and display some sort of error message to prompt the user to fix the issue.

Preconditions: Bus routes are already configured

Postconditions: The passengers will be added/simulated in the system

Related Requirements: 3

Assumptions: Creating the passengers is functional, and that they can get to their location using a bus

Author: Connor Boehm

SPEC: 5 **Title:** Create Passengers **Use Case:** 1

Date: 11.9.2020

Specification Description:
- The user specifies the number of new passengers they want to generate.
- The user is then prompted to set a routine for these passengers.
    - A passenger routine is defined as a set consisting of the bus stop of origin, destination stop, and generation time.
- After generation, the user should be able to see the passenger at the stop of origin at the generation time.
- The passenger will board the next bus that comes to their stop that has their destination on its route given that the bus is not already at capacity.

Preconditions: Validate that all passenger parameters are within allowable range.

Postconditions: Passenger is created.

Related Requirements: 5

Assumptions: N/A

Author: Austin Wessel

SPEC: 6 **Title:** Historical Passenger Data **Use Case:** 1

Date: 11.9.2020

Specification Description:

- Allow the user to select a file and upload historical passenger data.
    - Historical passenger data includes data like passenger counts during particular times on different days of the week, stop popularity, and effects of temperature and passenger count.
- Process this data and create passenger patterns from it.
    - An example of a passenger pattern would be as follows: Thursday 9/15, Sunny, 74 degrees F, 3:00PM and all of the passenger counts historically recorded in those conditions.
- When passenger data becomes outdated, users will be able to delete it or replace it with more relevant data.

Preconditions: N/A

Postconditions: Passenger patterns are created from uploaded historical data.

Related Requirements: 6

Assumptions: File type is interpretable by our system

Author: Austin Wessel

SPEC: 7 **Title:** Create Passenger Pattern **Use Case:** 1

Date: 11.9.2020

Specification Description:

- Allows the user to create a passenger pattern from either historical data or from a new custom pattern.
- The passenger pattern was defined in Specification 5.3.2.2.1, but in this case also includes the input of passenger frequencies.
- Passenger frequencies will generate passengers with the same origin and destination at a time interval the user specifies.
- Upon creation the user will then see the new created pattern added to the system and can begin running a simulation with it.

Preconditions: All parameters required for a passenger pattern are met.

Postconditions: New passenger pattern created..

Related Requirements: 7

Assumptions: N/A

Author: Austin Wessel

SPEC: 8 **Title:** Delete Passenger Pattern **Use Case:** 1

Date: 11.9.2020

Specification Description:

- Allow users to choose a previously created passenger pattern for deletion.
- Verify that the user wants to delete the pattern.
    - Once deleted, patterns cannot be recovered.
- Remove the selected passenger pattern from the system and return its previously allocated memory back to the system.

Preconditions: Passenger pattern is still active in the system and can be deleted.

Postconditions: Passenger pattern is deleted and corresponding memory is returned to the system.

Related Requirements: 10

Assumptions: Passenger pattern is not currently being used in a running simulation.

Author: Austin Wessel

SPEC: 9 **Title:** Create/Configure Route **Use Case:** 1

Date: 11.9.2020

Specification Description:

- Allow users to create a new route by selecting a series of stops and roads connecting them.
- Allow users to modify an existing route by selecting stops to add to the selected route.

Preconditions: Route is complete and does not include impassable terrain.

Postconditions: Route is created and can now be selected for simulations.

Related Requirements: 11

Assumptions: N/A

Author: Austin Wessel

SPEC: 10 **Title:** Delete Route **Use Case:** 1

Date: 11.9.2020

Specification Description:

- Allow users to choose a previously created route for deletion.
- Verify that the user wants to delete the route.
    - Once deleted, routes cannot be recovered.
- Remove the selected route from the system and return its previously allocated memory back to the system.

Preconditions: Route is still active in the system and can be deleted.

Postconditions: Route is deleted and corresponding memory is returned to the system.

Related Requirements: 12

Assumptions: Route is not currently being used in a running simulation.

Author: Austin Wessel

SPEC: 11 **Title:** Log Contents

Date: 11.9.2020

Introduction: The contents of a log must be in a defined format for clarity. This requirement details the types of formats a log may have.

Rationale: Having several types of log contents allows for clarity in each log entry. If all details of the system were confined to one log format, seeing changes in the system by viewing the logs would prove quite challenging.

Precondition: None

Specification Description: There are several types of log entries. These log entries are those for:

1. ERROR
2. WARN
3. INFO
4. DEBUG
5. TRACE

These entries entail the following:

1. ERROR = Some aspect of the simulation cannot continue due to some cause. Describe the error
2. WARN = Some aspect of the system isn't performing ideally, describe the error.
3. INFO = Some aspect of the system is occurring that requires a status update
4. DEBUG = Information that may pertain to debugging in the future
5. TRACE = The path of calls taken to reach a certain point of the simulation

Post Condition: None

Persistent Changes: Logs

Related Requirements: 13

Support Material: None

Author: Jackson McEligot

iv.

SPEC: 12 **Title:** Log Entry addition

Date: 11.9.2020

Introduction: A log entry addition is the act of adding a log entry to a log. This is used to keep logs up to date with the simulation status.

Rationale: A log stores past progressions of a simulation. To add the current instance, a log must take additions.

Precondition: A log exists.

Specification Description: For each simulation run, a log entry must be capable of being added to the current simulation log without manipulating past logs or log entries.

Postcondition: The existing log has the added log entry

Persistent Changes: A simulation run's log.

Related Requirements: 13

Support Material: None

Author: Jackson McEligot

v.

SPEC: 13 **Title:** Log Update Timing

Date: 11.9.2020

Introduction: A simulation log should update as actions happen within the system, this specification entails when those updates happen.

Rationale: For a log to be useful, it must indicate when aspects of the system don't act as intended. These scenarios must then trigger a log update.

Precondition: Simulation run is underway.

Specification Description: A log entry should be written to a simulation log each time The system detects that something does not go as intended. This includes errors, warnings, general important information, information relevant to a debug, or a trace of the system's methodology to reach a certain point.

This new log entry should entail what did not go as intended.

Postcondition: Log entry addition is triggered.

Persistent Changes: None.

Related Requirements: 13

Support Material: None

Author: Jackson McEligot

vi.

SPEC: 14 **Title:** Log memory

Date: 11.9.2020

Introduction: Covers how a log is stored during and after simulation run.

Rationale: To manipulate a log, it must be known to the system and thus stored in memory. When a simulation run has ended, the log will no longer be manipulated, and thus should be stored in a more permanent location.

Precondition: None

Specification Description: Log details must be capable of being stored in system memory for updates, and written to a more permanent memory location for storage

Postcondition: None

Persistent Changes: None

Related Requirements: 14

Support Material: None

Author: Jackson McEligot

vii.

SPEC: 15 **Title:** Log retrieval

Date: 11.9.2020

Introduction: This requirement covers how the system must allow for log retrieval.

Rationale: For writing, reading, and deleting a log, the system must have a method of determining where in memory a log is stored based off the provided input log id.

Precondition: Log exists, log retrieval based on log id requested

Specification Description: To retrieve a log, the system is provided a log id. The system must be able to determine the memory location of the related log. The system must also be able to interpret previous log information based on this saved log's memory location.

Postcondition: Log memory location determined.

Persistent Changes: None

Related Requirements: 15, 16

Support Material: None

Author: Jackson McEligot

viii.

SPEC: 16 **Title:** Log size retrieval

Date: 11.9.2020

Introduction: This provides the system with a size of memory being freed for deletion

Rationale: This requirement is used for the appropriate deletion of a log, to know how much memory should be freed from the starting address given to the system in a deletion scenario.

Precondition: Log exists, Memory location of log known

Specification Description: The system must have a method of determining the size of a log based off its memory location.

Postcondition: Size of log known to system

Persistent Changes: None

Related Requirements: 16

Support Material: None

Author: Jackson McEligot

ix.

SPEC: 17 **Title:** Log memory freeing

Date: 11.9.2020

Introduction: Entails how the system handles log deletion.

Rationale: Used to free up memory for other uses upon deletion, mainly in the case of logs.

Precondition: A memory address and an amount of memory freed are given to the system

Specification Description: The system must have a way to remove a set of memory from the system scope, freeing it for other uses.

Postcondition: Previously occupied memory is freed.

Persistent Changes: Occupied memory

Related Requirements: 16

Support Material: None

Author: Jackson McEligot

SPEC: 18 **Title:** Information in Report **Use Case:** 1

Date: 11.9.2020

Introduction: After a single simulation has finished running, a report generates information based on all of the statistics generated during the simulation.

Rationale: Being able to report the average capacity of each bus allows the user to observe which buses spend too much time over capacity, and make decisions about which buses to use on a route to meet rider demand.

Specification Description:
The report contains the following information:

-Total time elapsed in the simulation
-Type and total capacity of each bus
-Route specified for each bus
-Average capacity of each bus, as a percentage
-Duration of time in the simulation that each bus was over capacity
-Average and longest length of time that passengers spent waiting at each stop that were not picked up by a bus on time.

Preconditions: A simulation must have run to generate the report.

Postconditions: A report is created with the data given above.

Related Requirements:

Assumptions: Simulations are run independently, one a time.

Author: David Ibarra

SPEC: 19 **Title:** Report Stored on System **Use Case:** 1

Date: 11.9.2020

Introduction: After individual simulations are run by the user, a report for each simulation is stored on the system.

Rationale: Reports may need to be looked at later by the user who has access to the system and kept for later without creating an unnecessary number of files outside of the system.

Specification Description: The total amount of storage for the reports stored on the system is a user-configurable amount that is specified when starting the system for the first time. The report is generated after each simulation automatically, and is stored within the SURTS system.

Preconditions: A simulation has been run by the user.

Outputs: A report is generated on the system, using an amount of the preconfigured memory.

Related Requirements:

Assumptions: The simulations are run one at a time by the user, and there is not a significantly large number of simulations created or reports generated at any time.

Author: David Ibarra

7. Other Nonfunctional Requirements

7.1 Performance Requirements

The system must be capable of of handling a simulation with thousands of passengers and hundreds of buses without maxing out CPU usage or completely filling memory/storage space.

7.2 Security Requirements

There may be security concerns with the acquisition and storage of data provided by individual transportation service organizations.

7.3 Software Quality Attributes

The transit organization specifies that the product should be realistic and make the simulation as close as possible to the management of transit on campus, when it comes to simulating variables and creating a model of student demand.
However, more precise information has not been asked of the user yet, so more information will need to be collected.

7.4 Business Rules

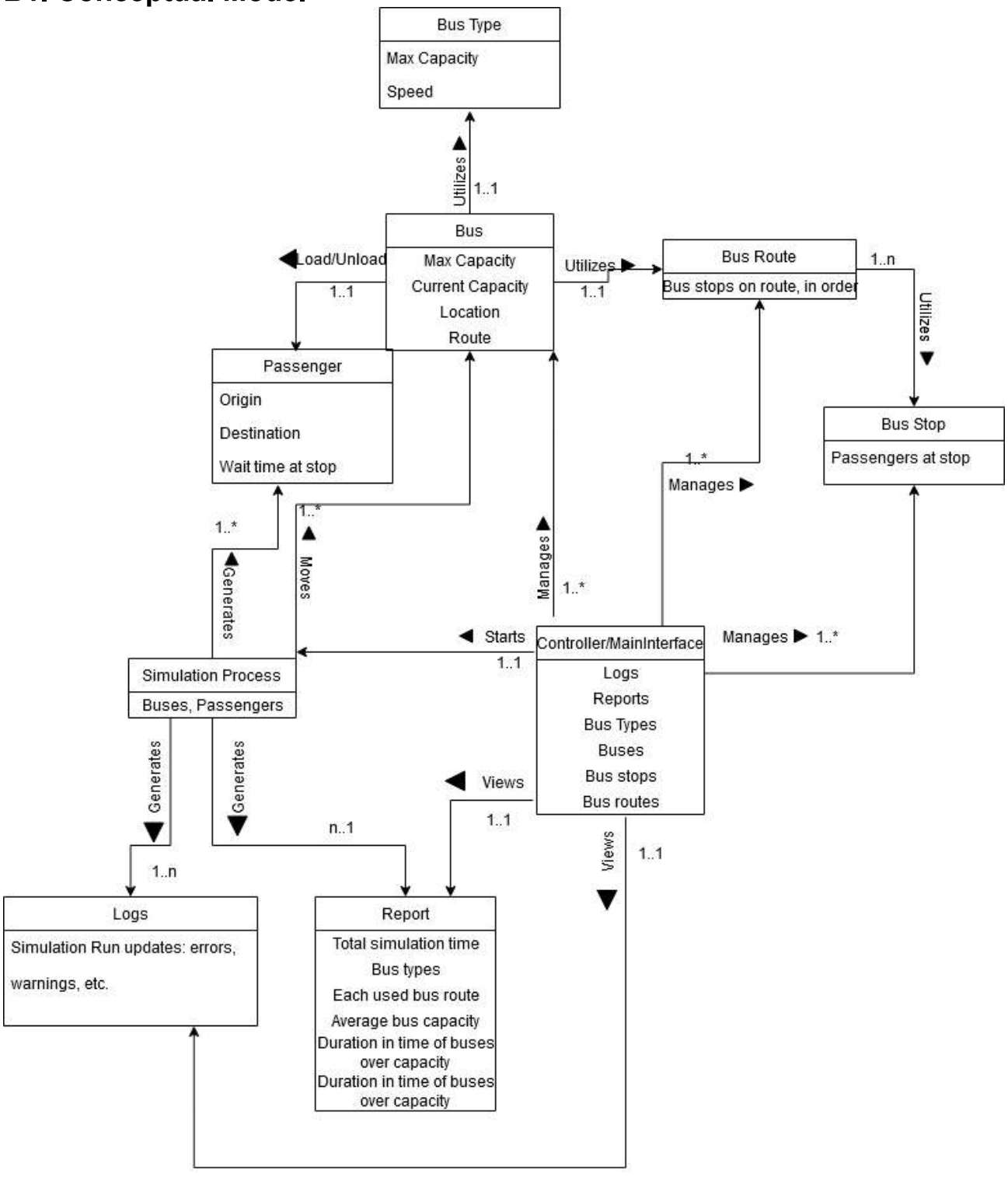
Only designers and producers of the system or staff within the transit organization will be allowed access to the data for previous Parking and Transportation Services' ridership. No user external to the transit organization should have the ability to control, see, collect, delete, or otherwise change this data during normal use of the system.

Appendix A: Glossary

SURTS . . . . . .System for Uniform Route-Based Transportation Simulation
IEEE . . . . . Institute of Electrical and Electronics Engineers
P&TS . . . . . .Parking and Transportation Services
UMN . . . . . University of Minnesota

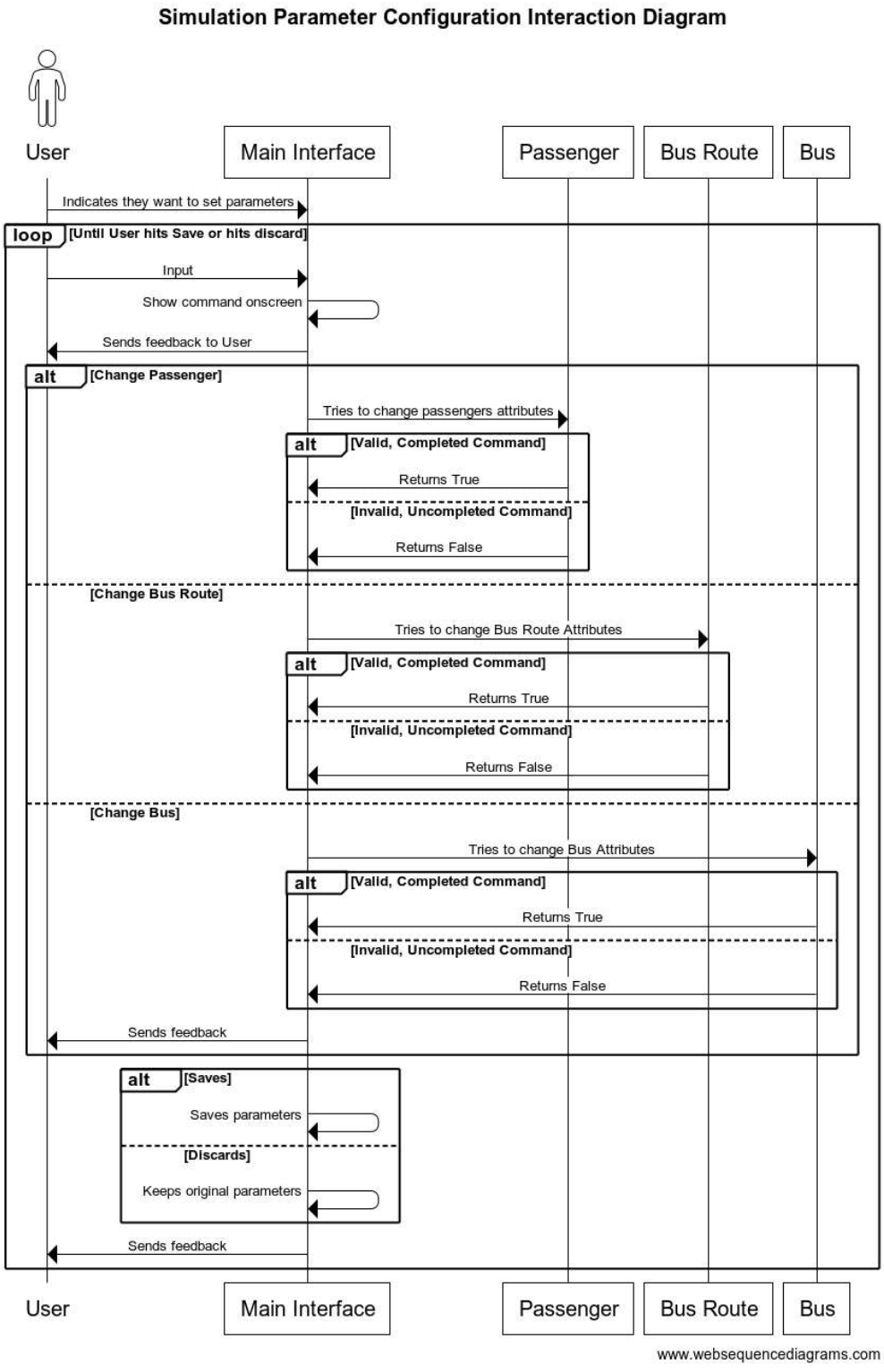
Appendix B: Analysis Models

B1. Conceptual Model

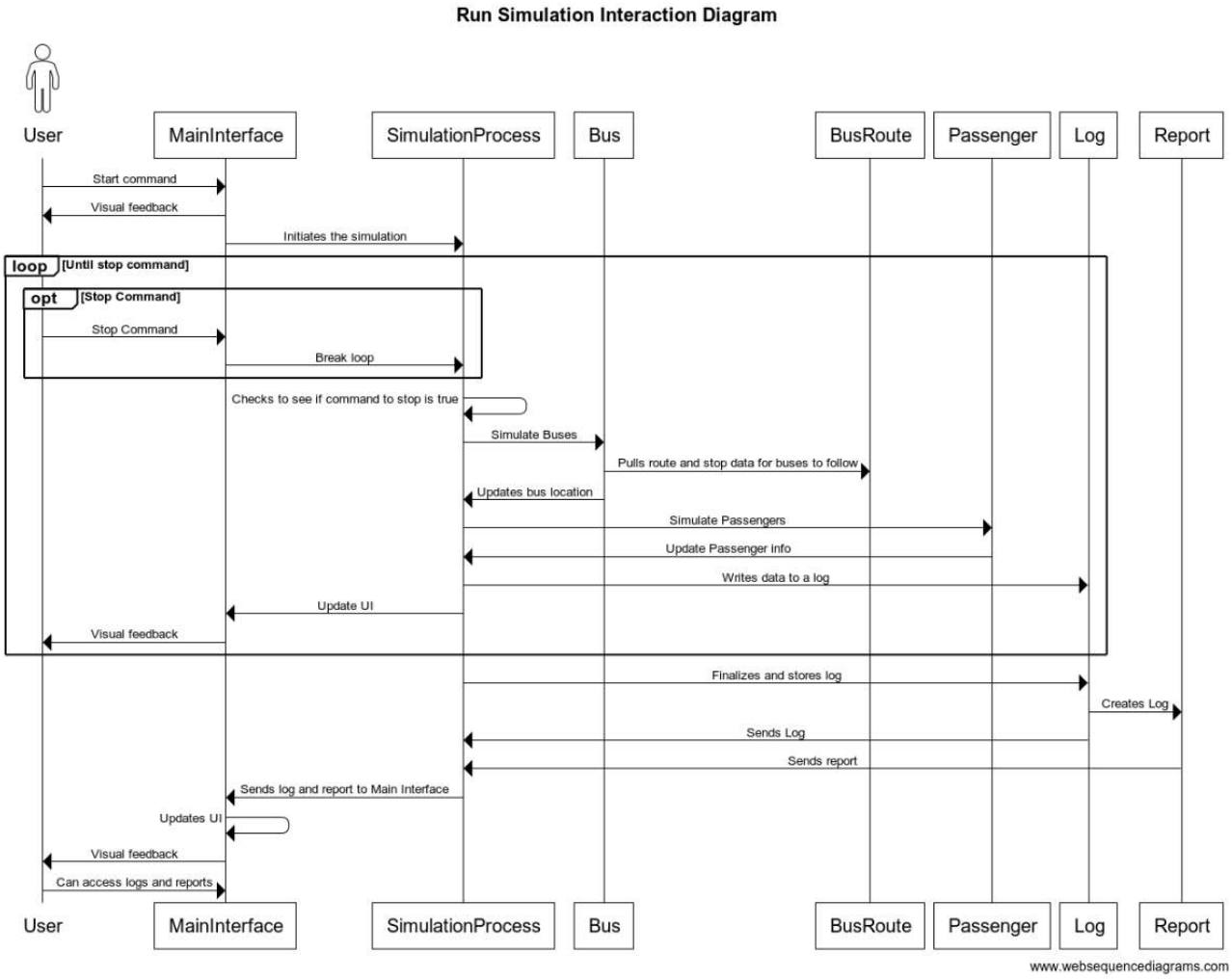


B2. Interaction Models

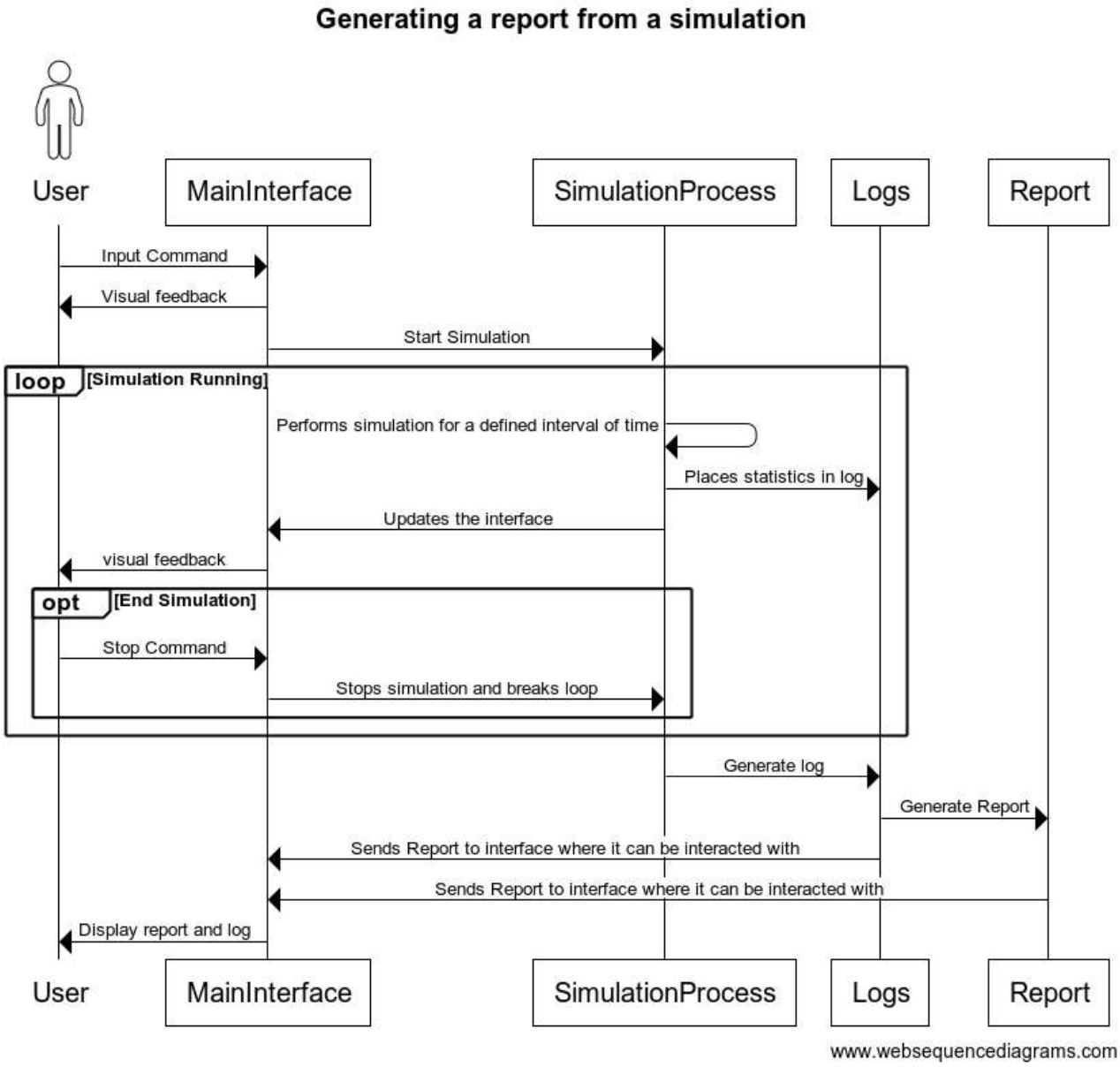
B2.1 Simulation Parameter Configuration



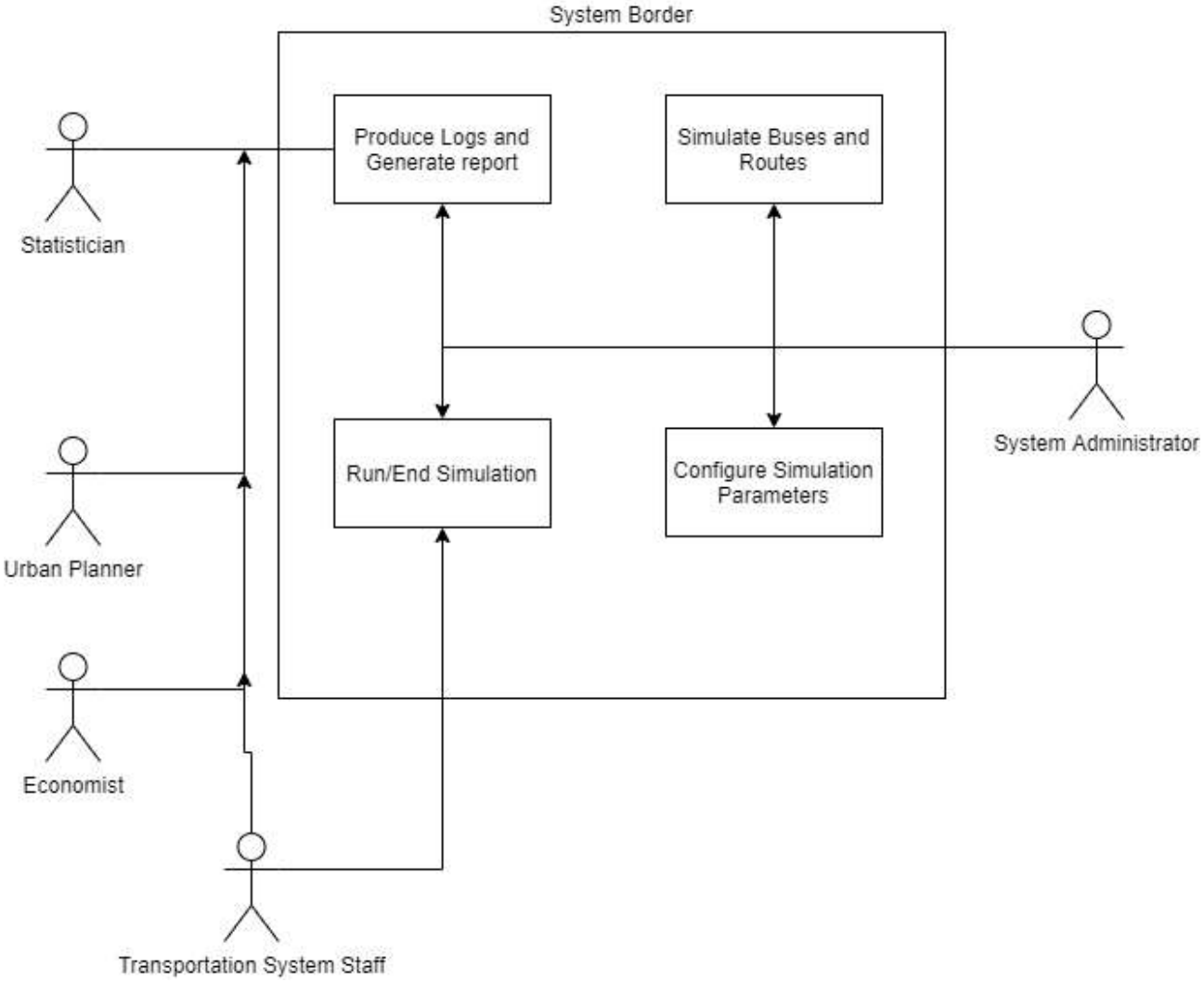
B2.2 Simulation Runtime



B2.3 Generating a Report from a Simulation



B3 Use-Case Diagram



Appendix C: To Be Determined List

All remaining TBDs do not belong to the business layer of the application.