
Software Requirements Specification

for

System for Uniform Route-based Transportation Simulation (SURTS)

**Prepared by: Nathan Bourgeois, Jonathan
Larsen, David Ma, Thomas Pfeifer**

University of Minnesota

Date: 11/05/2020

Table of Contents

Table of Contents.....	
Revision History.....	
1. Introduction.....	
1.1 Purpose.....	
1.2 Document Conventions.....	
1.3 Intended Audience and Reading Suggestions.....	
1.4 Product Scope.....	
2. Overall Description.....	
2.1 Product Perspective.....	
2.2 Product Functions.....	
2.3 User Classes and Characteristics.....	
2.4 Operating Environment.....	
2.5 Design and Implementation Constraints.....	
2.6 User Documentation.....	
2.7 Assumptions and Dependencies.....	
3. External Interface Requirements.....	
3.1 User Interfaces.....	
3.2 Hardware Interfaces.....	
4. System Features.....	
4.1 System Feature 1.....	
4.2 System Feature 2	
5. Other Nonfunctional Requirements.....	
5.1 Performance Requirements.....	
5.2 Security and Safety Requirements.....	
5.3 Software Quality Attributes.....	
5.4 Business Rules.....	
6. Test Cases.....	
6.1 Objective.....	
6.2 Definitions and Acronyms.....	
6.3 Traceability Matrix.....	
6.4 Test Cases.....	

Revision History

Name	Date	Reason For Changes	Version
Nathan B Jonathan L David M Thomas P	10/22/2020	Combine all four SURTS requirement documents	1.1
Nathan B Jonathan L David M Thomas P	11/05/2020	Refining requirements and adding test cases	1.2

1. Introduction

1.1 Purpose

The purpose of this document is to describe the requirements of the **SURTS** system completely, accurately and unambiguously without the use of overly technical language. All attempts have been made to use everyday language and commonly understood technical terminology.

1.2 Document Conventions

The **System for Uniform Route-based Transportation Simulation** will hereafter be referred to as “**the system**” or “**SURTS**”.

This document is written in user language and is not meant to describe any of the details of implementation on a programming level.

1.3 Intended Audience and Reading Suggestions

This document is intended for all users who are to provide insight into the creation of SURTS. This includes: Transportation system staff, urban planners, statisticians, economists, system administrators, project managers and software developers. It is written in user language and aims to capture all major system functionalities.

1.4 Product Scope

The **SURTS** systems main objective is to be able to simulate buses, bus routes, and passengers along with producing understandable results from these simulations. The goal of using **the system** is to interpret the simulation data to develop transit routes which meet the needs of the

users as well as minimizing expenses. The scope of this document includes the initial user requirements for **the system**.

2. Overall Description

2.1 Product Perspective

The **SURTS** project has been greenlit by the University of Minnesota transportation role holders to better understand how resources can be more appropriately utilized. This project is currently planned to be a self-contained product.

2.2 Product Functions

The main functions of the system shall be to:

- Simulate buses transporting passengers
- Add or remove buses at specified times
- Add or remove passengers needing to use the bus
- Add or remove routes for the buses
- Produce a log of various information about the simulation run
- Store log information
- Allow users to inspect the logs
- Store logging information for a user defined period of time up to a month
- Allow users to modify the simulation attributes
- Produce reports that summarize the simulation results, including passenger wait times, bus travel times, and total simulation throughput. Specific report contents can be found in 4.7.

2.3 User Classes and Characteristics

The various classes of users who will use the system are as follows:

- System administrators, Transportation system staff, Urban planners, Statisticians, and Economists.
 - View Log
 - View/Generate Report
 - View/Modify System Configuration
 - Run Simulation

2.4 Operating Environment

The system shall be able to operate correctly on a device capable of running an operating system, and have an amount of available storage for storing logs.

2.5 Design and Implementation Constraints

The system shall have some level of security such that only authorized users will be able to access it. There are no constraints on the specific frameworks and technologies that need to be implemented. The system will be maintained by the developers, since it would not differ too much between each user.

2.6 User Documentation

Documentation components including a user manual, and a tutorial are to be delivered with the system.

2.7 Assumptions and Dependencies

Assumptions:

- A bus route must have at least two stops.
- A bus must have a minimum capacity of one passenger.
- A bus must have no passengers after the last stop on the route.
- A passenger must have a destination.
- A passenger will wait at the stop until the bus arrives.
- A route must have at least two stops.
- A report is generated the simulation has completed.

3. External Interface Requirements

3.1 User Interfaces

The system will have a user interface. This interface will allow the user to modify simulation parameters, run the simulation, view simulation logs, and generate reports.

3.2 Hardware Interfaces

The system will be able to run on a device running some kind of operating system, where the device has sufficient computational power and speed to run the system. The system will not need to support mobile devices.

4. System Features

4.1 System Configuration Requirement

4.1.1 Description and Priority

This is a high priority requirement, as **SURTS** would be incomplete without the ability to customize the various simulations. It is this feature that makes **SURTS** into a complete

simulation that can provide useful information to its users. Users will need to be able to customize several attributes, including:

- Default number of passengers
- Default number of buses
- Default bus capacity
- Default bus ID
- Default number of routes
- The routes the buses travel
- The number of buses and their capacity
- The number of passengers

4.1.2 Stimulus/Response Sequences

The user will configure the bus, passenger, and route system variables to start the simulation. To generate a report the user will need to configure the report requirements.

REQ-1: Error Handling: If the user uses an improper specification, the system will use predefined default specifications for the appropriate field. If no default has been entered, the system will prompt the user to define one.

REQ-2: Interface: The system will have an interface which allows the user to modify system parameters. The interface also allows the user to run the simulation (4.5), view logs (4.6), and generate reports (4.7).

REQ-3: Configure Buses to generate. Generate a bus with a type, route, and passenger count starting at 0.

REQ-4: Configure Passengers to generate. Generate a passenger with a start location and end location.

REQ-5: Configure a route with a list of bus stops. This configuration will be passed to the bus when it is generated.

4.2 Bus representation

4.2.1 Description and Priority

This is a high-priority requirement as the functionality of buses is important to accurately represent in order to accomplish the goals of the **SURTS**.

4.2.2 Stimulus/Response Sequences

Buses will have routes upon generation. Routes may be changed. The bus will run its route and at each stop will load and unload passengers. When the bus is removed from the simulation, it will complete the rest of its route before being removed.

REQ-1: All buses shall be able to traverse a user defined route.

REQ-2: Each bus shall be able to on-load passengers.

REQ-3: Each bus shall be able to off load passengers.

REQ-4: Each bus shall have a minimum capacity of one passenger.

REQ-5: Each bus shall have a user defined maximum capacity.

REQ-6: Each bus shall have a unique ID.

REQ-7: Each bus shall have a number of passengers on board.

REQ-8: Each bus shall have a number of passengers boarded and debarked.

REQ-9: Each bus shall have a start time and location.

4.3 Passenger representation

4.3.1 Description and Priority

This is a high-priority requirement as the functionality of passengers is important to accurately represent in order to accomplish the goals of **SURTS**.

4.3.2 Stimulus/Response Sequences

Passengers will be generated using user-defined configuration. A passenger will have a boarding stop and debarking stop on an appropriate route.

REQ-1: The passenger shall be able to board a bus.

REQ-2: The passenger shall be able to deboard a bus.

REQ-3: The passenger shall have an onboard stop.

REQ-4: The passenger shall have a deboard stop on the current bus route.

REQ-5: The passenger shall have a wait time.

REQ-6: The passenger shall have a ride time.

REQ-7: The passenger shall have a bus boarded.

4.4 Route representation

4.4.1 Description and Priority

This is a high-priority requirement as being able to represent a bus route that has a number of stops where the buses can board and deboard passengers is important to accurately represent in order to accomplish the goals of **SURTS**.

4.4.2 Stimulus/Response Sequences

The routes shall be generated using user-defined configuration which includes information such as the path to be traveled and the number of stops.

REQ-1: The route shall allow buses to follow its path.

REQ-2: The route shall have at least two stops where passengers are able to board and deboard.

REQ-3: Stops on the route shall have a passenger on/off boarding frequency.

REQ-4: The route shall have a unique route ID.

4.5 Simulate Buses Transporting Passengers

4.5.1 Description and Priority

This is a high-priority requirement, as it serves as the primary basis for **SURTS'** functionality. At its core, **the system** needs to simulate a network of buses transporting passengers from stop to stop. It needs to do this by tracking all relevant information - passengers, buses, and the various routes. It will also track information about the simulation throughout its runtime, so that it can be used in logs and reports. The exact information to be tracked will differ by log and report, and more information can be found in 4.6 and 4.7, respectively.

4.5.2 Stimulus/Response Sequences

The number of buses, along with their routes, and passenger information shall be generated based on user configurations mentioned above in sections 4.1-4.4. The simulation shall allow for the addition and subtraction of both buses and passengers both before the beginning as well as while the simulation is in progress and will run until completion or stopped.

REQ-1: The results of a simulation will be logged.

REQ-2: General Error Handling: If for some reason the simulation crashes in the middle of runtime, it will exit gracefully and notify the user of the error, as well as any information related to the error.

REQ-3: SURTS will track the various actors and objects within the simulation.

REQ-4: The simulation shall have a list of active buses.

REQ-5: The simulation shall have a list of active routes.

REQ-6: The simulation shall have a list of passengers.

REQ-7: The simulation shall have a start and end time.

REQ-8: The simulation shall generate a report upon completion.

4.6 Record Logs of the Simulation

4.6.1 Description and Priority

This is a high-priority requirement as allowing users to glean information about the simulation is required for appropriate analysis. Logs should detail a variable amount of information about each simulation run, since different users may be interested in different information. Logs will be stored for a user defined period of time up to one month so that users can go back and inspect previous logs. Users should be able to easily access the logs from the **SURTS** interface.

4.6.2 Stimulus/Response Sequences

Data in the form of logs shall be captured and saved to a database. This data shall have an interface to be accessed by users or other systems. Data shall be sorted by time and simulation run.

REQ-1: Logs shall be recorded along with the following attributes:

1. Time and Date of log.
2. Unique simulation run ID.

REQ-2: Logs shall be accessible by a user interface.

REQ-3: The system shall maintain the logged information for a user defined time period up to a month.

REQ-4: There shall be different log levels to track such as: ERROR, WARN, INFO, DEBUG, TRACE.

4.7 Generate reports

4.7.1 Description and Priority

This is a high-priority requirement as having **the system** generate reports that summarize simulation results in an understandable manner is key to being able to gain useful insight into the modeled transportation system.

4.7.2 Stimulus/Response Sequences

When the system finishes running a simulation, it will generate a report based on the most recently run simulation. Alternatively, the user may generate a report by selecting a log in which they wish to generate a report for.

Req-1: The system shall generate a report which includes the following information:

- The simulation parameters
- Number of passengers serviced
- Maximum, minimum, and average passenger wait time
- Maximum, minimum, and average passenger ride time
- Maximum, minimum, and average bus capacity utilized
- Number of passengers not serviced
- Number of busses in service
- Routes in service
- Number of busses per route in service
- Duration of simulation (start and end simulated times)

5. Other Non-functional Requirements

5.1 Performance Requirements

The longest amount of time that the simulation will need to run for is a single day, since we can treat each work-day as independent of the others. The system can be run in real time or accelerated up to as quickly as the hardware running the system can handle. That is to say, a user can run the entire simulation instantly just to inspect results, but they can also run the simulation in real time to inspect various aspects of the system. The system ought to be able to run smoothly on most average computers. The maximum limits on the various attributes of the system - the routes, the passengers, and the buses - will be somewhat larger than the current maximums of real life transportation systems. This makes it so that ambitious transportation systems can be simulated, but that there does exist a cap on its size to maintain performance.

5.2 Security and Safety Requirements

No sensitive data of any type will be stored on the system. Therefore, the security measures that are inherited from the implemented language will be sufficient.

5.3 Software Quality Attributes

The software shall work correctly and produce the correct output based on the applicable input. The software shall be able to be operated with less than two hours of training. The software will exit gracefully in the event of an error occurring during operation.

5.4 Business Rules

The system is simple enough that complex stratified access to its features are unnecessary, but there can still be a couple different roles to be represented. A default user has access to all of its features. A system administrator merely grants and takes away access from default users.

6. Test Cases

6.1 Objective

The purpose of the Test Requirements section is to list ALL hardware and software test requirements, whether explicitly determined from any relevant documents or implicitly determined from experience and product knowledge. For most projects, the documents referred to may be the Product Definition Document, Software/Hardware Requirements Specification and perhaps the Software/Hardware Design Specification. A Test Case Matrix is provided that simply lists all the test cases by title or description, and includes a method of tracking when the test case was run and whether it passed or not.

6.2 Definitions and Acronyms

List any technical terms or acronyms used in the document, along with their meanings.

Examples for this document:

SRS - Software Requirements Specification

TM - Traceability Matrix

SURTS - System for Uniform Route-based Transportation Simulation

6.3 Traceability Matrix

Requirements / Test Case	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14
Allow user to modify simulation attributes	X	X	X	X	X	X	X	X	X	X					X
Simulate Buses Transporting Passengers									X	X					X
Add/Remove Routes							X	X							
Add/Remove Buses					X	X									
Add/Remove Passengers			X	X											
Produce a log											X	X			
Store a log									X	X					X
Produce a report									X	X			X	X	X

6.4 Test Cases:

T0 Default System Configuration

Description: Ensures that if there are input errors, a default behavior is enacted.

Test Inputs:

User defined default number of passengers.

User defined default number of buses.

User defined default bus capacity.

User defined default route.

Expected Results: The number of passengers, buses, bus capacity, and route defaults are set based on the user's input.

Dependencies: None

Initialization: None

Test Steps:

1. Enter the default route ID.
2. Enter the default list of stops with their respective passenger frequency.
3. Enter the default number of buses.
4. Enter the default bus capacity.
5. Enter the default bus ID.
6. Enter the default number of passengers.
7. Verify that the default attributes are correctly set to the above entered values.

T0 error case

T1 Default System Configuration Check

Description: Ensures that if invalid values are entered for defaults, the user is prompted until valid values are provided.

Test Inputs:

Invalid default values of number of passengers, buses, bus capacity, and route.

Expected Results: The user is prompted to reenter valid values for the above fields.

Dependencies: None

Initialization: None

- Test Steps:**
1. Enter invalid default number of passengers.
 2. System prompts for valid default number of passengers.
 3. Enter valid default number of passengers.
 - 4-15. Repeat steps 1-3 for default number of buses, bus capacity, bus ID, and route.
 16. Verify that valid default values have been set.

T2 Generate Passengers

Description: Checks that the simulation can represent a number of passengers.

Test Inputs: Total number of passengers desired.

Passenger contains the following fields:

On board stop: Stop ID

Off board stop: Stop ID

Wait time: time

Ride time: time

Bus boarded: Bus ID

Expected Results: Passengers are generated with on and off boarding stops based on route stop frequency.

Dependencies: At least one bus route is available for service.

Initialization: At least one bus route is available for service.

- Test Steps:**
1. Enter the total number of desired passengers.
 2. Generates each passenger with an on and off board stop using available routes.
 3. View passenger information and verify that each passenger has one on and off board location.
 4. Verify that passengers' wait time, ride time, and bus boarded information is empty.
 5. Verify that passengers' on and off board locations are on a route in service.

T2 Error case

T3 Default Generate Passengers

Description: Sets the number of passengers to a user defined default if an invalid number of passengers is entered.

Test Inputs: An invalid number of passengers.

Expected Results: A user defined default number of passengers are generated with on and off boarding stops based on route stop frequencies.

Dependencies: The default system configuration has been set.

Initialization: None

Test Steps:

1. Enter an invalid number of desired passengers.
2. Notifies the user that default values will be used.
3. Generates a user defined default number of passengers with an on and off board stop using available routes.
4. View passenger information and verify that there are the correct number of passengers generated based on the defined default number.
5. Verify that passengers each have one on and off board location.
6. Verify that passengers' wait time, ride time, and bus boarded information is empty.
7. Verify that passengers' on and off board locations are on a route in service.

T4 Generate Buses

Description: Checks that the simulation can represent a number of buses.

Test Inputs: Number of buses desired. Bus ID. Buses capacity. Buses route.

Buses contain the following fields:

Bus ID: ID

Bus route: Route ID

Capacity: maximum

Passengers onboard: number

Passengers boarded: number

Passengers deboarded: number

Start Location: Stop

Start Time: time

Expected Results: Buses are generated with a unique ID, a maximum capacity, and route to travel.

Dependencies: The default system configuration has been set.

Initialization: None

Test Steps:

1. Enter the total number of buses desired.
2. Enter each bus ID, max capacity, and route.
3. Verify that each bus has a unique ID, max capacity and route.
4. Verify that buses' boarded, deboarded, and on board fields are empty.

T4 error case

T5 Default Generate Buses

Description: Sets the number of buses to a user defined default if an invalid number of buses is entered.

Test Inputs: Invalid number of buses, bus ID, bus capacity, and bus route.

Expected Results: The user defined default number of buses are generated with the default capacity, default unique ID, and default route.

Dependencies: The default system configuration has been set.

Initialization: None

Test Steps:

1. Enter an invalid number of buses.
2. Notifies the user that default values will be used.
3. Generates a user defined default number of buses.
4. Enter an invalid bus ID.
5. Sets the bus ID to a user defined default unique bus ID.
6. Enter an invalid bus capacity.
7. Sets the bus capacity to a user defined default bus capacity.
8. Enter an invalid route.
9. Sets the bus route to a user defined default route.
10. Verify that all buses generated have the correct default values for their ID, capacity and route.

T6 Generate Route

Description: Checks that the simulation can represent a number of bus routes.

Test Inputs: Number of routes desired. Route ID's. Route stops. Stop frequency.

Routes contain the following fields:

Route ID: ID

Stops: List of stops with their respective passenger frequency

Expected Results: Routes are generated with a unique ID, and a list of stops on the route with each stop having a frequency of passengers.

Dependencies: None

Initialization: None

Test Steps:

1. Enter the total number of routes desired.
2. Enter the route ID.
3. Enter the route stops.
4. Enter the stops passenger frequencies
5. Verify that each route has at least 2 stops.
6. Verify that each stop has a passenger frequency.
7. Verify that each route has a unique ID.

T6 Error Case

T7 Default Generate Route

Description: Sets a route to a user defined default route if an invalid route is entered.

Test Inputs: Invalid number of routes, route ID, and list of stops with their respective passenger frequencies.

Expected Results: Sets the number of routes, route ID's, and list of stops with their respective passenger frequencies to a user defined default if invalid number of routes, route ID's, or lists of stops with their respective passenger frequencies is entered.

Dependencies: The default system configuration has been set.

Initialization: None

Test Steps:

1. Enter an invalid number of routes.
2. Notifies the user that default values will be used.
3. Generate a user defined default number of routes.
4. Enter an invalid route ID.
5. Sets route ID to a user defined default route ID.
6. Enter an invalid list of stops with their respective passenger frequencies.
7. Sets list of stops and their respective passenger frequencies to a user defined default value.
8. Verify that all routes generated have default values for their route ID, and list of stops.

T8 Run Simulation:

Description: Checks that the simulation is able to run with a number of buses.

Test Inputs: Start time: time
End time: time

Expected Results: The simulation runs to completion, displays the report, and logs the simulation results.

Dependencies: Default System Configurations have been set.

Initialization: Generate Route, Generate Buses, and Generate Passengers have been run successfully.

Test Steps:

1. Enter the simulation start and end times.
2. Begin the simulation.
3. Upon completion, view the report and verify its correctness.
4. Verify that the information from this simulation run has been logged and is correct.

T8 Error case

T9 Run Simulation Time Check:

Description: Checks that the simulation has a valid start and end time.

Test Inputs: Invalid simulation start and end times.

Expected Results: The simulation will prompt the user to enter a valid start and end time until that condition is met. Then the simulation runs to completion, displays the report, and logs the simulations results.

Dependencies: Default System Configurations has been set.

Initialization: Generate Route, Generate Buses, and Generate Passengers have been run.

Test Steps:

1. Enter an invalid simulation start time.
2. The simulation prompts the user for a valid simulation start time.
3. Enter a valid start time.
4. Enter an invalid end time.
5. The simulation prompts the user for a valid simulation end time.
6. Enter a valid end time.
7. Begin the simulation.
8. Upon completion, view the report and verify its correctness.
9. Verify that the information from this simulation run has been logged and is

correct.

T10 View Log

Description: Checks that the logs of simulations are being correctly stored.

Test Inputs: Simulation ID

Expected Results: The log of the simulation with matching ID is displayed.

Dependencies: At least one simulation has been run successfully.

Initialization: None

Test Steps:

1. Enter the simulation ID.
2. SURTS retrieves and displays the logs corresponding to the input ID.
3. Verify that all the log information is present

T10 Error Case

T11 Can't View Log

Description: Informs the user a log cannot be viewed.

Test Inputs: Invalid Simulation ID

Expected Results: The user is informed that the log cannot be viewed because it does not exist or that its simulation ID is invalid.

Dependencies: None

Initialization: None

Test Steps:

1. Enter an invalid simulation ID.
2. The system informs the user the log cannot be viewed and why it cannot be viewed.

T12 View Report

Description: Checks that the reports from a simulation can be viewed and are correctly displayed

Test Inputs: Simulation ID

Expected Results: The report of the simulation with matching ID is displayed.

Dependencies: At least one simulation has been run successfully.

Initialization: None

Test Steps:

1. Enter the simulation ID.
2. SURTS retrieves and displays the logs corresponding to the input ID.
3. Verify that all the report information is present and correct.

T12 Error Case

T13 Can't View Report

Description: Informs the user a report cannot be viewed.

Test Inputs: Invalid Simulation ID

Expected Results: The user is informed that the report cannot be viewed because its log does not exist or that its simulation ID is invalid.

Dependencies: None

Initialization: None

Test Steps:

1. Enter an invalid simulation ID.
2. The system informs the user the report cannot be viewed and why it cannot be viewed.

T14 Simulation Crash

Description: In the event that a simulation terminates prior to completion, it will exit gracefully while maintaining the user defined system configuration.

Test Inputs: None

Expected Results: The simulation terminates prior to completion, its logs are not saved and a report is not generated. The system configuration in which the system was running is maintained.

Dependencies: Run Simulation is currently running.

Initialization: Run a simulation

Test Steps:

1. While the simulation is running, terminate its execution prior to completion.
2. Verify that a log has been saved from the simulation that was running.
3. Verify that a report is not generated.
4. Verify that the system configuration is the same as when the simulation began.