User Requirement Specification

2015

Mikaeil Shaghelani Lor - Mervin Vrolijk - Bolarinwa Iruemiobe - Nibras Shawy - Tarwiya Al Ismaili | ei7S

Group 3

Date: 16/Sep/2015

version: 1.0.1

Contents

[Introduction 2](#_Toc430251028)

[Functional Requirements 2](#_Toc430251029)

[Place crossing 2](#_Toc430251030)

[Remove crossing 3](#_Toc430251031)

[Enter specifications 3](#_Toc430251032)

[Run simulation 4](#_Toc430251033)

[Post: Pop-up results visible on screen 4](#_Toc430251034)

[Save as 4](#_Toc430251035)

[Save 5](#_Toc430251036)

[Save simulation results 5](#_Toc430251037)

[Open 5](#_Toc430251038)

[Reset 6](#_Toc430251039)

[Stop simulation 6](#_Toc430251040)

[Exit Application 6](#_Toc430251041)

[About 7](#_Toc430251042)

[User Manual 7](#_Toc430251043)

[New 7](#_Toc430251044)

[User Interface 8](#_Toc430251045)

[Non-functional Requirements 9](#_Toc430251046)

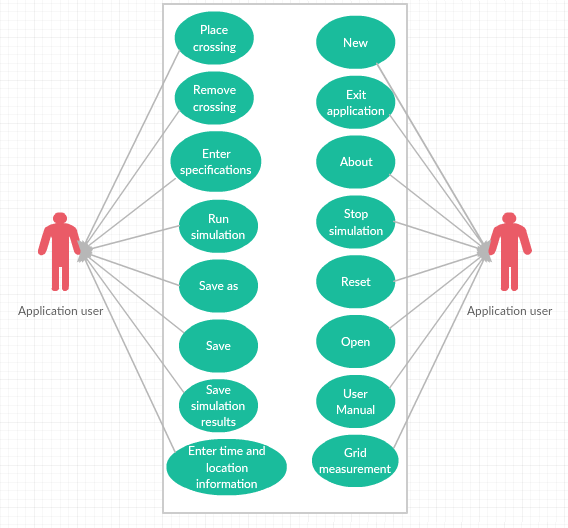
# Introduction

Assignment: Building a traffic simulation application.

In order to secure the transportation system in the city and decreasing the number of accident, our group of developer assigned to this project. Our main goal is to create a simulation system for the user to analysis the traffic flow and statistics. In this document you will find the use cases, design, non-functional requirements.

# Functional Requirements

**-Use Case diagram**



**-UseCases**

## Place crossing

Actor: The person who is using the software

Pre: The software is running

MSS:

1. Actor chooses crossing type from the toolbox
2. Actor clicks crossing and drags it to the desired place on the grid
3. System shows the item on the grid-cell where the user places it.

Ex:

1. System checks if actor overlaps crossing

Then system places it in the nearest empty grid-cell

Post: Selected crossing is visible on grid

## Remove crossing

Actor: The person who is using the software

Pre: The is at least one crossing on grid

MSS:

1. Actor clicks delete button from the toolbox
2. Actor clicks on the crossing
3. System remove crossing

Ex:

1. Actor doesn’t click on crossing

System informs actor about the error in status section.

Post: Clicked crossing is not visible anymore on grid

## Enter specifications

Actor: The person who is using the software

Pre: There is at least one crossing on grid

MSS:

1. Actor chooses crossing layer from Crossing Panel
2. Actor specifies required crossing information for every available feeder (see figure 2.b)
3. Actor clicks on apply button
4. System saves values in the model
5. System shows success message

Ex:

* 1. Actor didn’t specify all required information
  2. System informs actor of the error in status section

Post: Success message is visible in status section

## Run simulation

Actor: The person who is using the software

Pre: The software is running and there is an open model

MSS:

1. Actor clicks on Start from the simulation box
2. System checks if all input is complete
3. System starts the simulation
4. System shows results at the end in a pop up form

Ex:

* 1. Actor didn’t input all required information

2.2 System informs actor of error in the status section and system goes stops the use case

## Post: Pop-up results visible on screen

## Save as

Actor: Person who is using the software

Pre: System is running

MSS:

1. Actor clicks on save as
2. System shows save as dialog
3. Actor choose the directory and inputs the file name
4. Actor clicks on save
5. System saves the project into a file.

Ex:

1. File name already exists

System informs actor

* 1. If coming from use case open system goes to use case open step 2
  2. If coming from use case exit application system goes to use case exit application step 2
  3. If coming from use case new system goes to use case new step 2

Post: Current project is visible on screen

## Save

Actor: The person who is using the program

Pre: Software is running

MSS:

1. The actor clicks on save from the file tab
2. System checks if project has been saved before.
3. System saves the project

Ex:

2.1 If the project hasn’t been saved before

2.2 System goes to use case save as step 2

3.1 If coming from use case open system goes to use case open step 2

3.2 If coming from use case exit application system goes to use case exit application step 2

3.3 If coming from use case new system goes to use case new step 2

Post: Current project is visible on screen

## Save simulation results

Actor: The person who is using the program

Pre: Simulation results pop-up is visible on screen

MSS:

1. The actor clicks on save.

2. System shows save as dialog

3. Actor choose the directory and inputs the file name

4. Actor clicks on save

5. System saves the results in a file

Post: Current project is visible on screen

## Open

Actor: The person who is using the software

Pre: The software is running

MSS:

1. The actor clicks on open from the file tab
2. System closes previous project
3. System opens file/project

Ex:

* 1. If a file is already open, system goes to save use case step 2

Post: Opened project is visible on screen.

## Reset

Actor: The person who is using the software

Pre: The simulation is running

MSS:

1. The actor clicks on reset from edit tab
2. System reset the simulation project

Post: New empty project is visible on screen

## Stop simulation

Actor: The person who is using the software

Pre: The simulation is running

MSS:

1. Actor clicks on Stop from the simulation control section
2. System Stop the simulation
3. System shows a message that simulation stopped in status section.

Post: Pop-up message is visible on screen

## Exit Application

Actor: The person who is using the software

Pre: The Software should be running

MSS:

1. Actor clicks on Exit from File tab
2. System closes the application.

Ex:

* 1. If project is modified system goes to save use case step 2

Post: Application is not visible on screen anymore

## About

Actor: The person who is using the software

Pre: The Software should be running

MSS:

1. Actor clicks on About in Help tab
2. System opens a pop-up page.

Post: Pop-up page is visible on screen

## User Manual

Actor: The person who is using the software

Pre: The Software should be running

MSS:

1. Actor clicks on the User’s Manual in Help tab
2. System opens the Digital User’s Manual.

Post: User’s manual is visible on screen

## New

Actor: Person who is using the software

Pre: The Software should be running

MSS:

1. Actor clicks on New from File tab
2. System closes previous file
3. System Opens a new project

Ex:

* 1. If a project is open or modified, system goes to save use case step 2

Post: New empty project is visible on screen

## Draw grid

Actor: Person who is using the software

Pre: The Software should be running

MSS:

1. Actor specifies measurements (see figure 2.c)
2. Actor clicks create grid
3. System checks if measurement is correct
4. System draws the grid

Ex:

* 1. If measurement is incorrect system informs actor about the error in status section.

Post: Grid is drawn on screen.

## Enter time and location information

Actor: Person who is using the software

Pre: The Software should be running

MSS:

1. Actor enters time and location information (see figure 2.a)
2. Actor clicks submit button
3. System checks if time is valid
4. System shows simulation form

Ex:

* 1. If time is invalid system shows error in message box

Post: Simulation form is shown on screen.

# User Interface

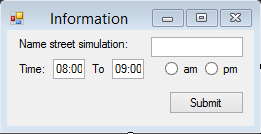


Figure 2.a

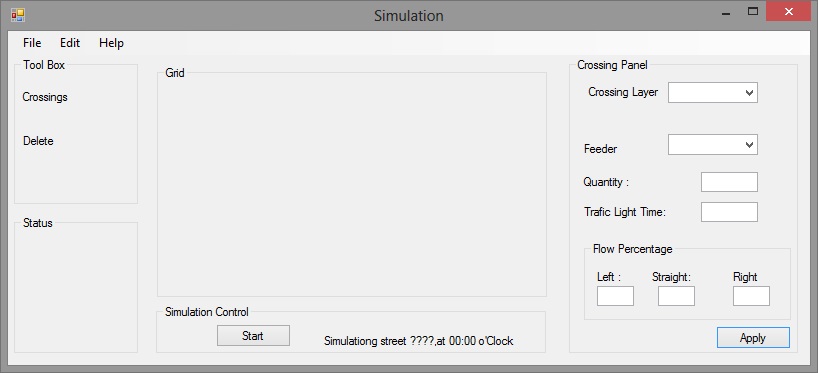
****

Figure 2.b

# Non-functional Requirements

(rules, restrictions?)

* This application will run on windows machines.
* Crossing’s positions: users can create crossing by dragging a crossing on the grid work space inside a grid cell.
* Completed Crossing simulation can be saved.
* Project can be saved.
* The user manual use case is a could requirement. All other use cases are must requirements.
* Every grid cell must have a crossing before running simulation.
* Traffic light time input must be in seconds.
* Car flow percentage must be whole numbers.
* A new crossing cannot be placed on a full grid.
* A maximum of 12 crossings.