URS Document

Group B

Member: Ishant Upadhyay

Raditya Surya Pratama

Franky Ngabo

Ji Nan

Hamid Reza Heydari

Hasan Lapa

Table of Contents

[1. Introduction 5](#_Toc413111904)

[2. Functional Requirements 5](#_Toc413111905)

[Use-Cases: 5](#_Toc413111906)

[1. Open and close the program. 5](#_Toc413111907)

[2.Save File 6](#_Toc413111908)

[3.Load File 7](#_Toc413111909)

[4.Create a new traffic simulation file. 7](#_Toc413111910)

[5.Adding crossing map 8](#_Toc413111911)

[6.Removing crossing map 8](#_Toc413111912)

[7.Increase or decrease Number of cars or pedestrians 9](#_Toc413111913)

[8. Increase or decrease Number of cars 9](#_Toc413111914)

[9. Adjust Traffic light duration 10](#_Toc413111915)

[10.Start/Pause Simulation 10](#_Toc413111916)

[11.Stop Simulation 11](#_Toc413111917)

[System Components 11](#_Toc413111918)

[Flow Diagram for Use Cases: 12](#_Toc413111919)

[Open application 12](#_Toc413111920)

[Close applicaton 12](#_Toc413111921)

[Save File 13](#_Toc413111922)

[Load File 13](#_Toc413111923)

[Create New 14](#_Toc413111924)

[Add map to grid 14](#_Toc413111925)

[Remove map 15](#_Toc413111926)

[Car Setting 15](#_Toc413111927)

[PedesteriaSettings 17](#_Toc413111928)

[Start/Pause the application 19](#_Toc413111929)

[Stop Application 20](#_Toc413111930)

[Light Time Duration 21](#_Toc413111931)

[3.User Interface 22](#_Toc413111932)

[4.Non-Functional Requirements 22](#_Toc413111933)

[Usability 22](#_Toc413111934)

[Testability 22](#_Toc413111935)

[Performance (Response Time) 22](#_Toc413111936)

[Robustness 22](#_Toc413111937)

[System Requirement 23](#_Toc413111938)

[5.MoSCoW 23](#_Toc413111939)

[6.Limitations: 23](#_Toc413111940)

[Transportation stream 23](#_Toc413111941)

[Pedestrian stream 24](#_Toc413111942)

[Speed 24](#_Toc413111943)

[Crossing 24](#_Toc413111944)

[Operation 24](#_Toc413111945)

[7.Traffic Light 24](#_Toc413111946)

# Introduction

The goal of the current document is to find out the needs and requirements of traffic light project.

In the following parts, we will be discussing the necessary requirements of the project. It is very important to initialize the user’s requirements in advance so we can avoid any misunderstanding and help improve the quality of the project.

We can categorize the requirements in two sections:

1. Functional Requirement

2. Non-functional Requirement

# Functional Requirements

Some features of the product which represent the functionality of the software. The following is the functional requirement:

1) Open and close the program.

2) Save the file.

3) Load the file

4) Create new file.

5) Adding crossing map

6) Removing Crossing map

7) Increase or decrease Number of cars

8) Increase or decrease Pedestrians

9) Adjust Traffic light duration

10) Pause –Start Simulation

11) Stop Simulation

## **Use-Cases**

1. Open and close the program

#### Open Program

**Goal:** Open the application.

**Actor**: User.

**Pre-Condition:** User has installed the application.

**MSS:**

1. User double-click on the icon of the application.
2. Program starts and main window appears.

#### Close Program

**Goal:** Close Application.

**Actor:** User.

**Pre-Condition:** The application is running.

**MSS:**

1. User clicks the X sign on the right-top button.
2. System close the application.

**Exception:**

1a- System asks if you want to save go to use case Save File.

1. Save File

**Goal:** To Save the File

**Actor:** User

**Pre-Condition:** The application window is open.

**MSS:**

1. User clicks save simulation button.
2. System displays a confirmation window.
3. User clicks save and choose the location of the file.
4. System saves the file.

**Exception:**

3a. User clicks no.

3b. System don’t save the file.

1. Load File

**Goal:** To load saved File

**Actor:** User.

**Pre-Condition:**  The main form is open.

**MSS:**

1. User choose the load option
2. System displays a browsing window.
3. User chooses the desired file.
4. System loads the file.

**Exception:**

4a. System displays a message that the type of file is not a valid type and asks user to select valid one.

1. Create a new traffic simulation file

**Goal:** Creation of a new simulation project.

**Actor:** User.

**Pre-Condition: \***The application is running.

\*There is no other existing

**MSS:**

1. User choose Create New option.
2. System creates a new traffic simulation.
3. Adding crossing map

**Goal:** Add Map to application

**Pre-Condition:** User is on the main form.

**Actor:** User

**MSS:**

1. User selects the map and chooses a place inside the grid to put that map.
2. The system show whether the selected grid is empty or not.
3. The system add new map on the selected grid.

**Extension:**

2a. there is already existing map on that location.

* System asks actor if it should override existing map.
* User selects Yes and the system overrides map.

2b. User selects No and return to the main screen.

1. Removing crossing map

**Goal:** Removing Map from the application

**Pre-Condition:** User is on the main form.

**Actor:** User

**MSS:**

1. User select the map which he/she want to remove.
2. User right click inside that selected map and choose delete.
3. The system show pop-up window whether he/she want to delete it or not.
4. User select yes.
5. The system delete the map in that selected grid.

**Extension:**

4a. User select No.

* System will not delete the map.

1. Increase or decrease Number of cars or pedestrians

**Goal:** To increase Number of Pedestrians

**Actor:** User

**Pre-:**  The simulation is running,

**MSS:**

1. Actor double clicks the desire area on the map.
2. System pause the simulation
3. System zoom the desired area.
4. Actor increase or decrease the number of Pedestrians.
5. Actor choose Confirm option.
6. System save the changes made.

**EXT:**

3)1. Actor chooses to close the zoomed area. Use case ends.

5)1.Actor chooses the cancel option. User goes back to case 4.

1. Increase or decrease Number of cars

**Goal:** To increase the number of cars.

**Actor**: User

**Pre-:** The simulation is running.

**MSS:**

1. Actor double clicks the desire area on the map.
2. System pause the simulation.
3. System zoom the selected area
4. Actor chose the incoming or outgoing area
5. Actor increase or decrease the number of cars.
6. Actor choose the confirm option.
7. System save the changes made.

**EXT:**

3)1. Actor chooses to close the zoomed area. Use case ends.

5)1.Actor chooses the cancel option. User goes back to case 4.

1. Adjust Traffic light duration

**Goal:** Light time setting.

**Pre-Condition:** User is in the main form.

**Actor:** User

**MSS:**

1. User selects the map and double click it.
2. The system shows the current map in enlarge form.
3. User chooses the time duration from options and select apply.
4. The system adjusts to the new setting.
5. The system shows the map with new setting.

**Extension:**

3a. User did not select apply.

* The time duration will be the set to the default one.

1. Start/Pause Simulation

**Goal:** Start and Pause application

**Pre-Condition:** User is in the main form.

**Actor:** User

**MSS:**

1. User click Start/Pause button.
2. The system check the status.
3. The system show the current status in the main form.

**Extension:**

2a. if the simulation system is running.

* The system will paused the application.

2b. if the simulation system is paused or not started.

* The system will start the system again.

1. Stop Simulation

**Goal:** Stop Simulation

**Pre-Condition:** User is in the main form.

**Actor:** User

**MSS:**

1. User click Stop button.
2. The system check the status.
3. The system show the current status on the main form.

**Extension:**

2a. if the simulation system is not running.

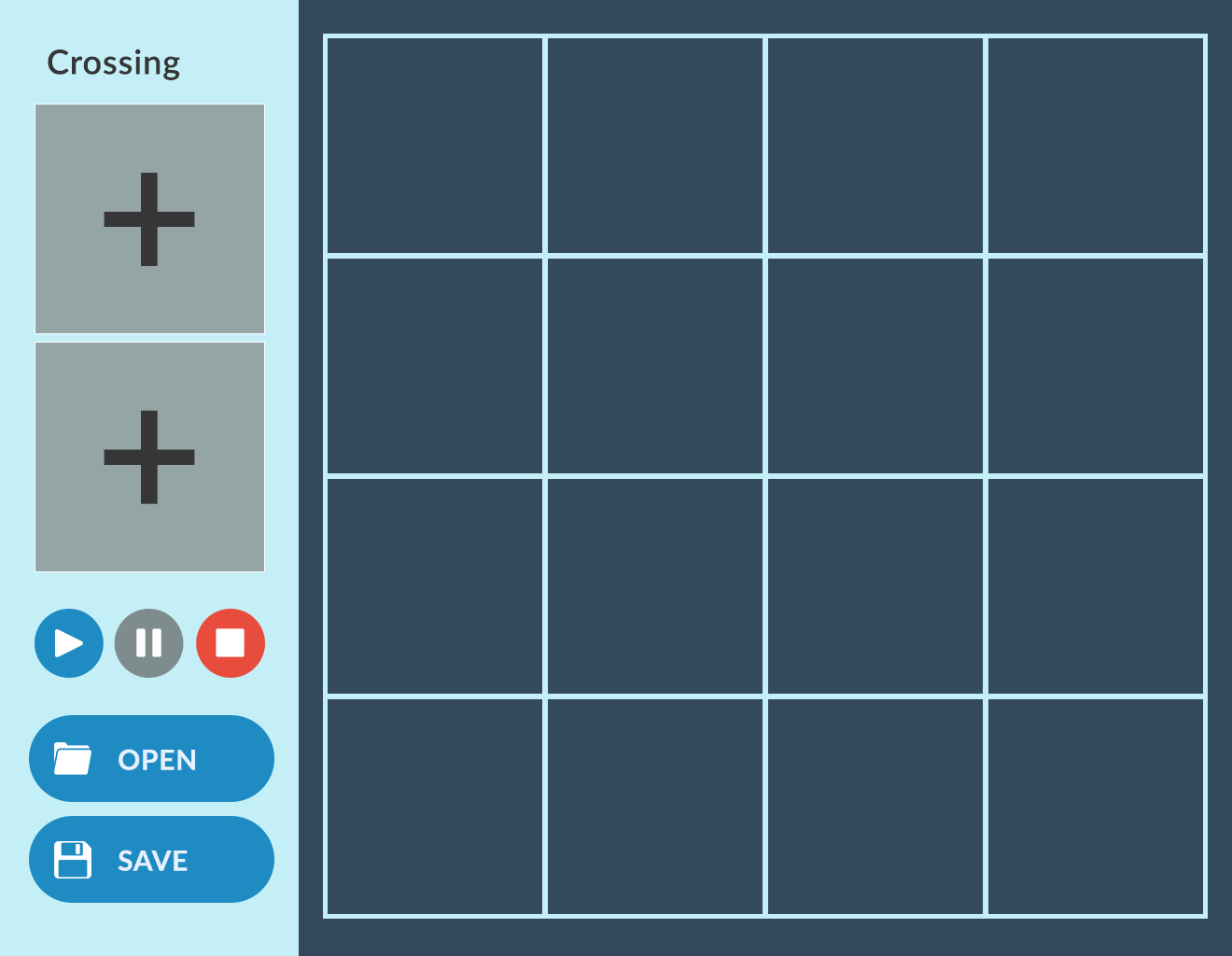
-The system will show main form.

## **System Components**

The main components of our simulation system is as follow:

* Traffic-Control-form: The project will load to the main form.
* Crossroads.
* Traffic- lights.

# 3. User Interface



4. None-functional Requirements

## Usability

The application should be easy to use and understand by the user.

## Testability

The application should be tested by some people, to ensure the application are working properly without bugs, problem and giving the specified result that user expected.

## Performance (Response Time)

The application should start quickly without taking a long period of time to load.

## Robustness

The application should be able to cope with errors during execution. It should give a meaningful error message if it’s encounter any problem and should not crash.

## System Requirement

Operating system: Windows 7 / 8 / 8.1

CPU: Pentium 4 1.5 GHz or higher

.NET Framework: .NET 3.5 or higher

Hard Drive: 50 MB of free hard disk space

Display: 800x600 or higher resolution monitor

5. MoSCoW

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Activity (Use-case) | M | S | C | W |
| Adjust distribution of cars between lanes | X |  |  |  |
| Adjust number of pedestrians per minute | X |  |  |  |
| Adjust “Green Light” time | X |  |  |  |
| Add crossing on the grid | X |  |  |  |
| Remove croassing | X |  |  |  |
| Save crossroad map | X |  |  |  |
| Laod crossroad map | X |  |  |  |
| New crossroad map | X |  |  |  |
| Start/Pause simulation button | X |  |  |  |
| Adjust car flow into crossroad from outside | X |  |  |  |
| Stop system button | X |  |  |  |
| Change direction of cars |  |  | X |  |
| Remove a car |  |  | X |  |
| Rotate crossroad |  |  | X |  |
| Add a car in the middle of crossroad |  |  |  | X |
| Remove pedestrians |  |  |  | X |

# Limitations

In order to maintain the performance of the Traffic Control Simulation, we set a certain amount of limitations. Here is the list of different limitations.

## Transportation stream

* Type of vehicles
* Maximum number of vehicles shown at the traffic lights.
* Maximum number of vehicles coming outside the crossing.

## Pedestrian stream

* Maximum number of pedestrians shown at the traffic lights.
* Minimum and maximum number of pedestrians coming to the traffic lights.

## Speed

* Vehicle speed.
* Pedestrian speed.

## Crossing

* Minimum and maximum number of crossings.
* Adding & Removing Crossings.
* Type of crossings.

## Operation

* Operations of the timer.
* Operations of the start, pause and stop.
* Operations of the performance.

## Traffic light

* Duration of the color change.

**Transportation stream**

* Type of vehicles

As far as the simulation of traffic control system is concerned, vehicles will only be presented as cars with small squares or rectangles. Also, details about the passengers will not be specified.

* Maximum number of vehicles shown at the traffic lights

Concerning about the scales of the maps, we decide to set the maximum number of vehicles shown at the traffic lights to 20.

* Maximum number of vehicles coming outside the crossing

The number of vehicles coming outside the crossing is set to 60 maximum, which would be better benefited in observing the stream.

**Pedestrian stream**

* Maximum number of pedestrians shown at the traffic lights

Concerning about the scales of the maps, we decide to set the maximum number of pedestrians shown at the traffic lights to 10.

* Minimum and maximum number of pedestrians coming to the traffic lights

The number of pedestrians coming outside the crossing is set to 30 maximum, which would be better benefited in observing the stream.

**Speed**

* Vehicle speed

The speed of vehicles at the crossing will be two options, including 50km/h and 70km/h. Lower speed will consume time to perform the car movement as well as higher speed will affect the process of checking the stream correctly. All vehicles will move at the same speed each time.

* Pedestrian speed

The speed of vehicles at the crossing will only be in one circumstance which is 4km/h. All pedestrians will move at the same speed each time.

**Crossing**

* Minimum and maximum number of crossings

In order to work the simulation, we have to have at least 1 crossing. Suggested number is 4-5. And on the other hands, we can have maximum 10 crossings.

* Adding & Removing Crossings

The crossing will be added and removed from the simulation map (form) when the simulation is stopped or paused. It will not be possible to add a second crossing or even more if it doesn’t connect to the existing one. On the other hand, it will also not be possible to remove crossing(s) if afterwards the rest crossings are not able to connect to each other.

The simulation must restart working from the beginning when the crossing is added or removed. This is mostly because otherwise it will cause troubles due to the fact that there are cars driving on the road and pedestrians walking.

Removing crossings, which divides map in several separate maps will result blank space in the map and the communication between other crossings will be interrupted or even lost unexpectedly.

* Type of crossings

There will be two kinds of crossings, which are three-direction crossing and four-direction crossing respectively.

**Operation**

* Operations of the input

The input will be made by the user.

* Operations of the performance

The performance (changing the traffic lights, cars movement, etc.) will be done automatically by the simulation, for this purpose some of the programming language tools will be used like timer.

**Traffic Light**

To specify the time duration of traffic light, we need to consider the speed of the cars. We suppose that the cars are inside the city so they do not have speed more than 80 km/h.

Depending on the speed of car, the required duration to change from green into yellow light will be as following:

a) 50km/h: the duration of yellow light will be 3 seconds. We assume a car with speed 50km/h needs almost 2 seconds to stop.

b) 80km/h: the duration of yellow light will be 5 seconds. We assume a car with speed 80km/h needs 3 seconds to stop.