Trafic Control System
User Requirements Document

Agnes Wadee
Coen Stange
Wen Li
Yongshi Liang

Background and Context

This user requirements document specifies the software requirements for the "Traffic Control System". This application allows traffic to be simulated with the purpose of noticing traffic jams related to traffic lights.

Definitions and abreviations

User	The person who is controlling this application.
System	The implementation of this application.
Grid	A place on the screen where a component can be added for the traffic situation.
Component	A visible representation of an object on the screen of the user.
Crossing	A component that can be used in the traffic simulation which has traffic lights.
Traffic light	A component of the crossing which controlls the traffic by displaying colors red, yellow green. For which green the traffic is allowed to go.
Pedestrian	A simulation of a pedestrian crossing a road from the traffic light.
Lane	A component that represent a piece of road.
Cars	A component that represent a car on the road.

Contents

1 Requirements

The following chapter describes the functionality that the application has to implement.

1.1 General requirements

Code	Requirement
GEN-010	The program is compatible with Windows 7.
GEN-020	The system allows to design a traffic situation.
GEN-020A	The traffic situation can be designed with the following components • Crossroad without pedestrian lane • Crossroad with pedestrian lane • Straight road • Curved road
GEN-020B	All crossroads have sensors for cars and pedestrians, traffic lights don't go green for no cars or pedestrians.
GEN-020C	Components can be rotated.
GEN-025	Components can be dragged onto the grid.
GEN-025A	The grid by default is 4x3 but it is possible to change the size of the grid before adding components.
GEN-030	From the traffic lights of the crossroads it is possible to change the amount of time that traffic light is green.
GEN-040	The system allows simulate traffic in realtime, and allow to change the simulation speed.
GEN-040A	The system can return a report in which it highlight where many traffic jams are in a graphical representation of the traffic situation. Which can be saved as an image file.
GEN-050	The system allows to open and save the traffic situation to a file.
GEN-060	The system allows to specify for each open incoming lane to set the amount of traffic coming.
GEN-200	The sytem will be delivered as a standalone application (.exe file).
GEN-200A	The system design and implementation will also be delivered to the client and has all the rights to it.

1.2 Non functional requirements

Code	Requirement
NFR-010	The system can run on a regular computer/notebook build less than 4 years ago running Windows.
NFR-020	The user interface of the system is straightforward and easy to learn.
NFR-030	The application can be used offline.
NFR-040	The system is stable, and is not allowed to give unclear error messages.
NFR-050	The system is designed to be used with mouse and keyboard.

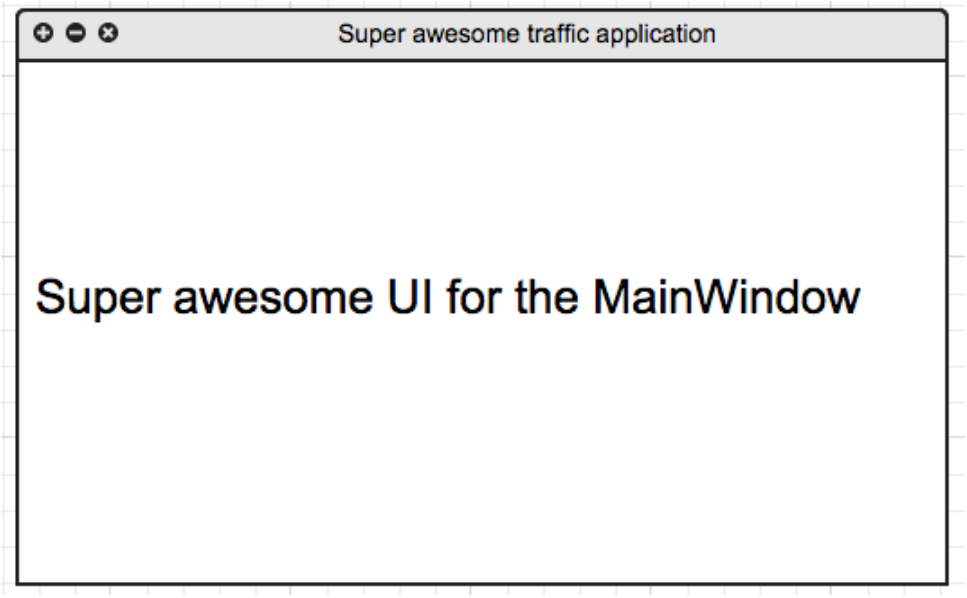
2 Specification

The following chapter describes the implementation of the application.

2.1 Main window

The main window is divided into two parts, see figure 1. On the very top of the application is a menubar where the user can do actions like saving their work. Then the window is split up in two parts. On the left side is the menubar and on the right side the grid. The grid is the representation of the traffic situation. The components (see section 2.2) can be dragged from the sidebar to the grid. To remove a component right-click on it and press on Delete in the context-menu. All open incoming lanes have a text-box which allows the user to specify the amount of traffic coming. The simulation can simply be started by the play button and the simulation speed can simply be changed with a slider. With the button "Show Report" the user can get a report of that moment of the simulation. The report will contain a still image of the current situation. It highlights the traffic jams and the image can be saved.

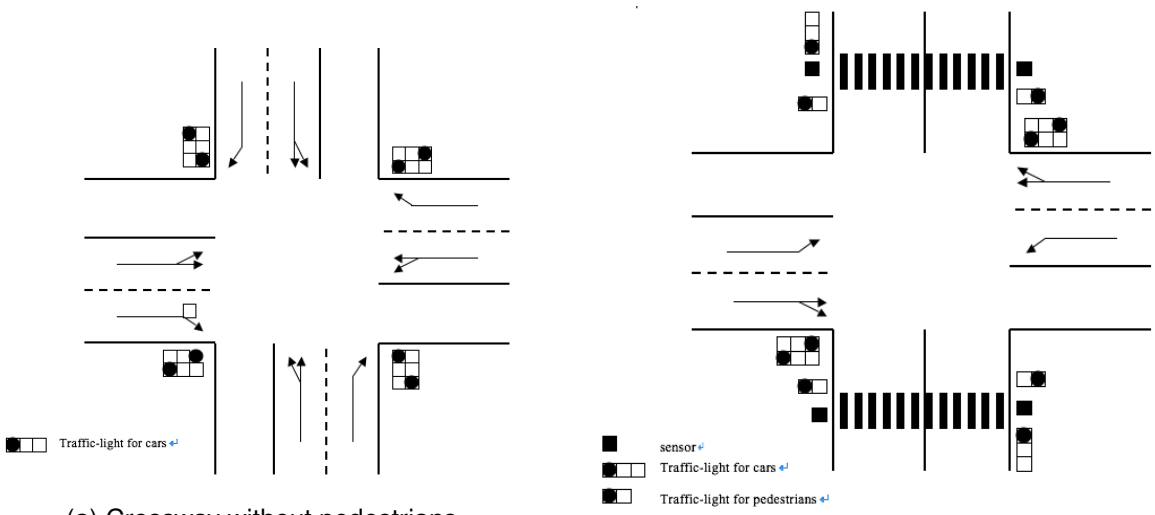
Figure 1: Mockup of the Main window



Code	Specification
MWS-010	When the application just started it will create a new grid 4x3
MWS-020	The main window has a menubar.
MWS-020A	The menu bar has the following structure: • File – New – Open – Save – Save As • Help
MWS-020B	A new simulation can be started by pressing on new, a window will prompt for the width and height for the size of the grid.
MWS-020C	The manual can be opened by pressing on Help.
MWS-030	The window has a sidebar on the left.
MWS-032	The sidebar contains all the components described in section 2.2.
MWS-032A	The component can be added to the grid by dragging it to the desired location.
MWS-034	The sidebar contains a button which allows the simulation to start/stop.
MWS-035	The sidebar contains a button which allows the simulation to pause.
MWS-036	The simulation-speed can be changed by adjusting the slider.
MWS-038	In simulation the button "Show report" will generate a report.
MWS-038A	The report is shown in a new window and contains the current traffic situation including cars and pedestrians.
MWS-038B	In the report the traffic jams are highlighted.
MWS-038C	The report can be saved as an image file.

To change the amount of time each traffic light is green press right-click on the crossway and click in the context-menu on "Traffic-light configuration". A new window will pop up which allows to set the time for each group of lanes.

Code	Specification
MWS-100	All open incoming lanes have a textbox to specify the amount of traffic coming in.
MWS-110	When pressing right-click on any component placed on the grid a context-menu appears which allows to rotate or delete the component.
MWS-120	When pressing right-click on a crossway it gives an option "Traffic-light configuration"
MWS-120A	A new window will pop-up with a list of all the lane groups.
MWS-120B	The user can select a lane group and change the amount of time the traffic-light is green.



(a) Crossway without pedestrians.

(b) Crossway with pedestrians.

Figure 2: Crossways

2.2 Components

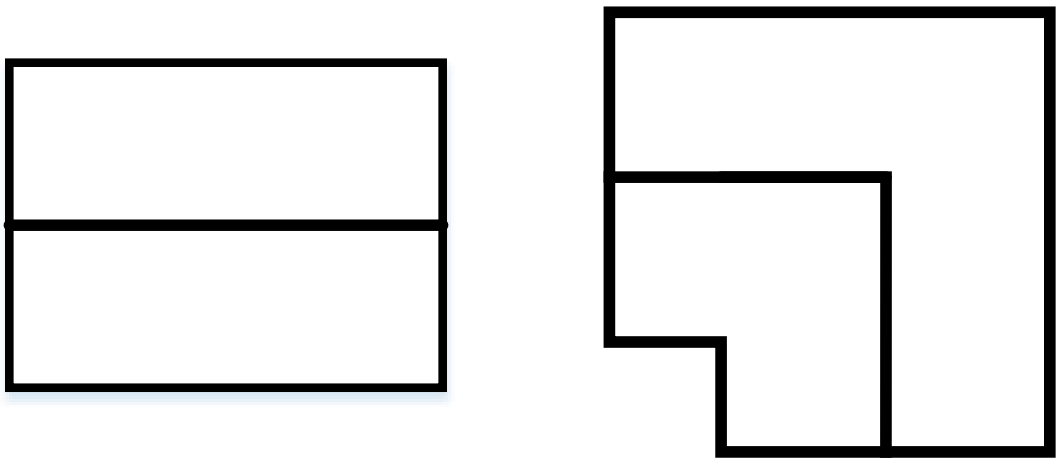
2.2.1 Crossway

Code	Specification
CWC-010	All cossways are connected to 4 roads
CWC-020	There are 2 different types of crossways see figure 3.
CWC-020	Type A crossway is without pedestrian lane.
CWC-020A	From each side the crossway A has 2 incoming lanes and 1 outgoing lane.
CWC-025	Type B crossway is with an pedestrian lane.
CWC-025A	Type B has from 2 opposite sides a crossway for pedestrians, which only has 1 incoming lane.
CWC-030	Traffic light for cars have the colors red, orange and geen.
CWC-035	Traffic light for pedestrians have the colors red and green.
CWC-040	Traffic light for cars only turn orange after green.
CWC-040A	Amount of time for the orange light is fixed, and is set to 2 seconds in normal simulation time.
CWC-045	Traffic light for green are set to default for 4 seconds.
CWC-045A	The amount of time each light group of an crossway can be changed.
CWC-050	The order of the light groups are fixed.
CWC-050A	When there are no cars or pedestrians on the sensors for the according light-group it will be skipped in the simulation.

2.2.2 Road

2.3 Traffic light groups

At the moment of writing this document it is unclear what the traffic light groups are.



(a) Crossway without pedestrians.

(b) Crossway with pedestrians.

Figure 3: Crossways

3 Use cases

Use Case 1	**Positioning a lane**
Level:	User-goal
Primary Actor:	End-User
Preconditions:	The application is open and no simulation is running.

Main Success Scenario:

1. User drags the lane from the sidebar.
2. User places the lane on the grid.
3. System sets the new placed lane and updates the grid.

Extensions:

2.a If the space where the new lane is put is already engaged, then the system will ignore the new lane and goes back to step 1.

1. System informs end-user that the lane cannot be placed, because at that space already exists a lane.
2. End of use case

2.b If the lane is placed outside the grid, then the system will ignore the new lane and goes back to step 1.

1. System informs end-user that the lane must be placed inside the grid.
2. End of use case

Post condition:	The program stays in "Positioning a road" state.

Use Case 2	**Rotating the lane**
Level:	User-goal
Primary Actor:	End-User
Preconditions:	There is at least one lane on the grid. No simulaion is running.

Main Success Scenario:

1. User right clicks on a lane.
2. System shows a menu with several options
3. User select "Rotate".
4. System turns the lane 90 degrees.
5. System updates the grid.

Extensions:

3 User does not want to execute any operation from the right click menu.

1 User clicks on any space outside the right click menu area.

2 End of use case.

Post condition:	The program stays in "Rotating the lane" state.

Use Case 3	**Positioning a crossing**
Level:	User-goal
Primary Actor:	End-User
Preconditions:	The application is open and no simulation is running.

Main Success Scenario:

1. User drags the crossing from the sidebar.
2. User places the crossing on the grid.
3. System displays a pop-up window.
4. User sets the initial settings for the selected crossing.
5. System sets the new placed lane and updates the grid.

Extensions:

2.a If the space where the new crossing is put is already engaged, then the system will ignore the new crossing and goes back to step 1.

1. System informs end-user that the crossing cannot be placed, because at that space already exists a crossing.
2. End of use case

2.b If the crossing is placed outside the grid, then the system will ignore the new crossing and goes back to step 1.

1. System informs end-user that the crossing must be placed inside the grid.
2. End of use case

4.a User does not set any of the required crossing attributes

(a) A system default value will be applied to undefined attributes.

4.b User does not want to set any settings.

1. User clicks on "Cancel" button.
2. System closes the setting window.

Post condition:	The system displays the updated grid.

Use Case 4	**Configurating traffic light timing**
Level:	User-goal
Primary Actor:	End-User
Preconditions:	There is at least one crossing on the grid and no simulation is running.

Main Success Scenario:

1. User right clicks on a crossing.
2. System shows a menu with several options.
3. User chooses "Traffic light configuration" from the menu.
4. System pops up a new window with configuration options.
5. User defines the amount of the time the traffic light is green.
6. User clicks on the OK button.
7. System closes the configuration window.
8. System updates the crossing.

Extensions:

3. User doesn not want to configurate anything.
    1. User clicks on "Cancel" button.
    2. System closes the configuration window.
    3. End of use case.

Post condition:	The system displays the updated grid.

Use Case 5	**Deleting an crossing**
Level:	User-goal
Primary Actor:	End-User
Preconditions:	The application is open. There is at least one crossing on the grid. No simulation is running.

Main Success Scenario:

1. User right clicks on a crossing.
2. System shows a menu of several options.
3. User clicks on "Delete".
4. System deletes the crossing and updates the grid.

Extensions:

3 User does not want to execute any operation from the right click menu.
    1 User clicks on any space outside the right click menu area.
    2 End of use case.

Post condition:	The system displays the updated grid.

Use Case 6	**Setting up a simulation**
Level:	User-goal
Primary Actor:	End-User
Preconditions:	There are components on the grid. No simulation is running.

Main Success Scenario:

1. User chooses setup to configure the simulation.
2. System displays input boxes of for each lane.
3. Users defines the amount of the cars coming through the lanes and fills in the input boxes.
4. Users clicks on "Set up".

Extensions:

3 User does not want to do any settings.

1 User clicks on "Cancel" button.

2 System hides the input boxes.

3 End of use case.

Post condition:	The system is ready for running the simulation.

Use Case 7	**Running a simulation**
Level:	User-goal
Primary Actor:	End-User
Preconditions:	The simulation is set up.

Main Success Scenario:

1. User clicks on "Play" button.
2. System runs the simulation.

Extensions:

2 System detects errors.

1. System stops running the simulation and gives an error message.

Post condition:	The simulation is running.

Use Case 8	**Stopping the simulation**
Level:	User-goal
Primary Actor:	End-User
Preconditions:	The system is running a simulation.

Main Success Scenario:

1. User clicks on "Stop" button.
2. System stops the simulation.

Post condition:	The system is not running the simulation.

Use Case 9	**Load file**
Level:	User-goal
Primary Actor:	End-User
Preconditions:	The application is open. No simulation is running. There is at least one saved file with traffic control system.

Main Success Scenario:

1. User selects file to open.
2. System closes previous project.
3. System loads the file.

Extensions:

2.a File can't be loaded.

1. System informs user that file can't be loaded and given the choice to stop or choose another file.
2. End of use case.

2.b Another project is open

1. System asks if users wants, to save project, that is already open, before closing it and opening another project.
2. End of use case.

Post condition:	The system has loaded an existing file.

Use Case 10	**Save file**
Level:	User-goal
Primary Actor:	End-User

Preconditions:	The application is open. No simulation is running.

Main Success Scenario:

1. User click the "Save" button
2. System saves the file.

Extensions:

2.a Project file has not been saved on the device yet.

1. A saving dialogue window pops up.
2. User chooses the directory where the file will be stored, and names the file.
3. User clicks " OK ".
4. System saves the file.

Post condition:	The system has saved a file.

Use Case 11	**Save file as a new file**
Level:	User-goal
Primary Actor:	End-User
Preconditions:	The application is open. Noo simulation is running.

Main Success Scenario:

1. User clicks " Save as " button.
2. A saving dialogue window pops up.
3. User chooses the directory where the file will be stored, and names the file.
4. User clicks " OK ".
5. System saves the file.

Extensions:

3.a User mis-clicks the " Save as " button and does not want to save the file as a new file.

1. User clicks " Cancel ".
2. Dialogue window closes.

Post condition:	The system has saved a file.

Use Case 12	**Resizing the grid**
Level:	User-goal
Primary Actor:	End-User
Preconditions:	The application is open. Noo simulation is running.

Main Success Scenario:

1. User clicks on Edit on the top bar.
2. System displays a menu with several options.
3. User clicks on "Document settings".
4. System shows document settings panel.
5. User defines the size of the grid by tying numbers in the width and height input boxes.
6. User clicks on OK.
7. System updates and display the new grid.

Extensions:

5 Use wants to make the grid size smaller.

(a) System discards the objects placed outside of the new grid size.

Post condition:	System displays the resized grid.