Use cases names –

***Add a crossing (intersection)***

***Remove a crossing***

Start the traffic

Pause the traffic

***Add an element (traffic light, sensor)***

***Remove an element***

Add pedestrian crossing

Set the flow –cars, pedestrians

***Load***

***Save***

Clear - restart

Monica, Alexandru, Rosen, Ventsislav, Dyma, Blagovest

## Deliverables

### Must

* Project plan
* User requirement specification (URS)
* Test Plan
* Class diagram design
* GUI design
* Prototype
* User’s manual
* C# Application
  + Two types of crossings possible
  + Place a minimum of 1 and a maximum of 12 crossings in a grid.
  + Adjust the car-streams coming from outside.
  + Real-time traffic movement
  + Adjust the ´green´ time of the traffic-light

* Process report

### Should

* Resizable working space
* Save and load

## Non-Deliverables

### Won’t

* Traffic lights
* Sensor
* Anything hardware relate

User wants to add a crossing.

**Actor**: User

**Precondition**: Program is running

**MSS**:

1. User selects the crossing type

2. User inputs the maximum car flow of the crossing in the input textbox

3. User presses submit button

4. User clicks on the drawing board where he wants to add the crossing (or use drag and drop?)

5. System draws the crossing

6. System updates the internal structure

**Ext:**

3a. If the input is in the wrong format (anything except positive integers), system shows a message and user has to go back to step 2

User wants to remove a crossing

**Actor**: User

**Precondition**: There is at least one crossing on the drawing board

**MSS**:

1. User selects the crossing

2. User presses remove button

3. System removes the crossing from the drawing board

4. System updates the internal structure

The user wants to add an element.

**Actor**: User

**Precondition**: There is at least one crossing on the drawing board

**MSS**:

1. User selects the crossing // Do we have to select the crossing first or?

2. User selects the element

3.The system adds the element to the crossing.

4. System updates the internal structure

The user wants to remove an element.

**Actor**: User

**Precondition**: There is at least one crossing on the drawing board and at least one element placed.

**MSS**:

1. User selects the crossing

2. User selects the element //Can you select the element? We have to decide

3. The system removes the element from the crossing

4. System updates the internal structure

**Goal:** Open an existing file

**Actors**: User of the system

**MSS:**

1. The actor presses the ‘Load file’ button.
2. The system displays a dialog box.
3. The actor presses the browse button and selects the file.
4. The actor confirms by clicking the Open button.
5. The system closes the dialog box.
6. The system loads the file.
7. The system displays all the information from the file.

**Extensions:**

**1a**.The system displays a dialog box asking the actor if he/she wishes to save the current file. If yes, go to use case save file and come back to step 2.

**3a**.The actor presses the ‘Cancel’ button and exits the use case

**5a**.The file is not in the correct form

1. The system displays a warning.
2. The actor is returned at MSS-step 3.

**Goal:** Save a file

**Actors**: User

**MSS:**

1. The actor presses the ‘Save’ button.
2. The system displays the time and date of the last save in a label informing the actor that the save is done.

**Extensions:**

1a.The file has no location or name on the disk then the system displays a dialog box asking the actor if he wants to save it.

1. b If the actor wants to save it, he is sent to ‘Save as file’ use case, step 2. If not, the use case ends.

**V.**

**Goal:** ‘Save as’ file

**Actors**: User

**MSS:**

1. The actor presses the ‘Save as’ button.
2. The system displays a dialog box.
3. The actor chooses a location.
4. The actor chooses a name for the file.
5. The actor confirms by clicking the Save button.
6. The system saves the file.
7. The system closes the dialog box.

**Extensions:**

**2a.**The actor presses the ‘Cancel’ button and the use case ends.

**4a.**There is already a file with that name

1. The system displays a warning and asking the actor if he/she wants to override the exiting file. If yes, the use case continues. If not, the actor is returned to step 4.

**Goal:** User want to start the simulation.

**Actor**: User

**Precondition**: Program is running, crossroad is added on the board

**MSS**:

1. The actor presses the ‘Start’ button.
2. The System animate a flow of cars (objects?) is shown/display on the crossroads on the board

**Ext:**

1a. If there is no crossroad on the board the system will display a warning message

1b. If the simulation was paused before this action, the animation will continue where it was paused

**Goal:** User want to pause the simulation.

**Actor**: User

**Precondition**: Program is running, the simulation is started

**MSS**:

1. The actor presses the ‘Pause’ button

2. The system will stop the animation of the flow with cars (objects?)

**Ext:**

1a. If there is no crossroad on the board the system will display a warning message

1b. If the simulation was not started the system will display a warning message