Use cases names –

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

***Remove a crossing***

Start the traffic

Pause the traffic

Stop traffic

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

***Remove an element***

Add pedestrian crossing

Set the flow –cars, pedestrians

Set timer for lights

***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

**MSS**:

1. User selects the crossing type
2. The user clicks on the screen on the board where he wants it placed
3. The system draws the crossing

Extensions: there is already a crossing there - error

The crossing is connected to another crossing –the flow is inherited

*Set the flow*

1. *User selects the lane that he wants to set the flow for*

*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 element
2. User clicks on a spot on the screen
3. The system checks if the click is in the right spot

3.The system adds the element to the crossing AND DRAWS IT ON THE SCREEN

4. System updates the internal structure

Ext: If the element is a traffic light for pedestrians or a sensor and there is no crossing, the system will display an error

There is already an element there the system displays an error

The user wants to remove an element.

**Actor**: User

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

**MSS**:

1. User press the remove button

2. User selects the element

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

**MSS**:

1. The actor presses the ‘Start’ button.
2. The system starts the timer for the lights and sensors
3. The System displays the flow of cars/pedestrians 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

**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

**Goal:** Set the flow –cars, pedestrians

**Actors**: User of the system

**MSS:**

1. The actor chooses a crossing.
2. The actor choses a lane
3. The actor presses the ‘change flow’ button.
4. The system displays a dialog box.
5. The actor chooses cars, or pedestrians.
6. The actor inputs the new flow.
7. The actor confirms the change.
8. The system closes the dialog box.
9. The system changes the flow

**Extensions:**

**5a**.The input of actor is not an integer.

1. The system displays a warning
2. The actor is return to MSS-step 2

**5b.** The input of actor is out of range

1. The system displays a warning
2. The actor is return to MSS-step 2

**5c**.The actor presses the ‘Cancel’ button and exits the use case

**Goal:** Clear the screen

**Actors**: User of the system

**MSS:**

1. The actor presses the ‘Clear’ button.
2. The system displays the dialog box asking if the user is sure.
3. The system deletes everything from the drawing board

**Extensions:**

**1a**.If there is something on the drawing board and not saved the system asks if the user wants to save it. Go to use case save as

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

Goal: Add pedestrian crossing

Change the green on the traffic light

# Nonfunctional requirements

* Microsoft Windows XP (Service pack 2) and further, Microsoft Vista, Microsoft Windows 7, Microsoft Windows 8, Microsoft Windows 10
* Desktop computer, notebook
* Monitor with a minimum screen resolution of 1024 x 768
* A mouse
* A keyboard

The final product will be an installation file and will have a user manual with it. The source code will be delivered. We are not providing service or updates. The program won’t have any auto save feature and won’t be able to repair corrupt save files.