

User REquirements Specification



Traffic Lights System

Bilger yahov

oLEKSANDR SUPRUNENKO

ILIA NIKUSHEV

GEORGI CHISHIRKOV

LYUBOMIR DIMOV

MENGCHUAN LIU

Table of Contents

[Introduction 2](#_Toc444808656)

[Functional requirements (use-cases) 2](#_Toc444808657)

[1. Select a crossing to place 2](#_Toc444808658)

[2. Place a crossing 2](#_Toc444808659)

[3. Remove a crossing 3](#_Toc444808660)

[4. Create a simulation 3](#_Toc444808661)

[5. Save a simulation 4](#_Toc444808662)

[6. Load a simulation 4](#_Toc444808663)

[7. Edit a road traffic flow 5](#_Toc444808664)

[8. Start a simulation 5](#_Toc444808665)

[9. Stop a simulation 5](#_Toc444808666)

[10. Pause a simulation 5](#_Toc444808667)

[11. Restart a simulation 6](#_Toc444808668)

[12. Undo an action 6](#_Toc444808669)

[13. Redo an action 6](#_Toc444808670)

[14. Save simulation results 7](#_Toc444808671)

[15. Show the help window 7](#_Toc444808672)

[16. Exit application 7](#_Toc444808673)

[17. Override simulation (Add police, ambulance, firetruck cars) 7](#_Toc444808674)

[18. Relocate crossing 8](#_Toc444808675)

[19. Startup the application 8](#_Toc444808676)

[20. Show simulation result 8](#_Toc444808677)

[21. Select crossing’s component to make changes 9](#_Toc444808678)

[22. Set current active crossing 9](#_Toc444808679)

[23. Edit a crosswalk’s pedestrian flow 9](#_Toc444808680)

[User interface 10](#_Toc444808681)

[Non-functional requirements 10](#_Toc444808682)

[Appendix A: Definitions 11](#_Toc444808683)

[Appendix B: MOSCOW 12](#_Toc444808684)

[Must-s: 12](#_Toc444808685)

[Should-s: 12](#_Toc444808686)

[Could-s: 12](#_Toc444808687)

# Introduction

The User Requirements Specification document contains requirements which the application will meet and the features it will possess. Every software application which is to be developed needs to meet two kinds of requirements – functional and non-functional.

Use-cases are tasks that describe the flow of actions between the user and the system. Each use case has a goal, Main Success Scenario and Extensions. The main success scenario shows the fastest way to achieve the goal of the use case, while Extensions cover situations which the system might come across if a criterial is met. Inside the document there are 23 use-cases.

Non-functional requirements of an application have to deal with its quality aspects such as how difficult it is to use it, how efficient, is it reliable and how maintainable is it.

# Functional requirements (use-cases)

## Select a crossing to place

Goal level: Sea level

Actor: User

Pre-condition: Simulation is not running.

Main Success Scenario:

1. User selects crossing from the available crossing types.

2. System updates the current selected crossing type.

3. System enters a mode for placing a crossing (see next use-case).

## Place a crossing

Goal level: Sea level

Actor: User

Pre-condition: Simulation is not running and a crossing type is selected.

Main Success Scenario:

1. User positions mouse over a grid slot.

2. System updates the GUI to show the availability on that position on the grid.

3. User clicks to place the crossing on the grid.

4. System changes the specified grid slot to be with the specified crossing type.

Extensions:

3a. Position contains a crossing.

3a1. System prompts the user to confirm the placement.

3a1a. User agrees to remove the current crossing.  
 3a1a1. Case continues with 4.

3a1b. User disagrees to remove the current crossing.

3a1b1. Case continues with 1.

3b. User moves the mouse away from the grid.

3b1. System updates the GUI.

3b2. System waits for action from the user.

## Remove a crossing

Goal level: Sea level

Actor: User

Pre-condition: The simulation is not running.

Main Success Scenario:

1. User drags a crossing from the grid into the recycle bin.

2. System removes it from the grid and places it into the recycle bin.

Extensions:

1a. User ends drag before on top of recycle bin.

1a1. System does nothing, crossing remains in place.

## Create a simulation

Goal level: Sea level

Actor: User

Pre-condition: The application is running.

Main Success Scenario:

1. User prompts the system to create a new simulation.
2. System empties the grid.

Extensions:

1a.Current grid is not empty.

1a1. System prompts the user to save their changes.  
 1a2. Case continues with 2.

1b.Current grid is empty.  
 1b1. System does nothing.

## Save a simulation

Goal level: Sea level

Actor: User

Pre-condition: There is at least one crossing on the grid.

Main Success Scenario:

1. User prompts the system to save the current simulation.
2. System provides with a dialog to save the simulation.
3. User specifies file name and location.
4. System saves the simulation.

Extensions:

3а. User does not specify name or location.

3а1. The system shows an error message informing the user.

3b. File with such name already exists.

3b1. The system shows message informing the user.

3b2. User selects to overwrite file or change the name of the file.

## Load a simulation

Goal level: Sea level

Actor: User

Pre-condition: None.

Main Success Scenario:

1. User chooses “Load a simulation”.
2. System provides with a dialog to load a simulation.
3. User chooses simulation.
4. System loads the simulation.

Extensions:

3а. Something goes wrong while loading the simulation

3а1. The system shows an error message informing the user.

## Edit a road traffic flow

Goal level: Sea level

Actor: User

Pre-condition: A road is selected as an active component (in an active mode).

Main Success Scenario:

1. User inputs the new flow
2. User confirms new value.
3. System sets the number as the current flow.

Extensions:

1a. User sets a negative or a value that is too high.

1a1. System sets the value to the closest possible number.  
1a2. Case continues with 2.

## Start a simulation

Goal level: Sea level

Actor: User

Pre-condition: The application is running and the simulation isn’t running.

Main Success Scenario:

1. User accesses the start functionality.
2. System starts the execution of the simulation.

Extensions:

1a. There are crossings on the grid which are not connected between.

1a1. System notifies the user that the start-up cannot be performed because of

not connected components.

## Stop a simulation

Goal level: Sea level

Actor: User

Pre-condition: Simulation is started.

Main Success Scenario:

* 1. User accesses the stop functionality.
  2. System stops the execution of the simulation.

## Pause a simulation

Goal level: Sea level

Actor: User

Pre-condition: The application is running and the simulation is running.

Main Success Scenario:

* 1. User accesses the pause functionality.
  2. System pauses the simulation at a certain position.

## Restart a simulation

Goal level: Sea level

Actor: User

Pre-condition: The simulation is running.

Main Success Scenario:

1. User accesses the restart functionality.
2. System stops the simulation.
3. System resets the simulation to its initial state.
4. System starts the simulation.

## Undo an action

Goal level: Sea level

Actor: User

Pre-condition: The simulation is not currently running.

Main Success Scenario:

1. User accesses the undo functionality
2. System restores the previous state of the application before the action was performed.

Extensions:

1. The simulation is still running.

2a1. System will display an error message and ask the use to stop the simulation.

## Redo an action

Goal level: Sea level

Actor: User

Pre-conditions: User performed some actions on the grid.

Main Success Scenario:

1. User accesses the redo functionality
2. System restores the previous state of the application before the action was undone.

## Save simulation results

Goal level: Sea level

Actor: User

Pre-condition: Simulation has been completed at least once.

Main Success Scenario:

1. User accesses the save simulation functionality.
2. System provides with possible formats in which the statistics can be saved.
3. User chooses a format.
4. System uses the operating system’s store a file feature to specify a file path and save the statistics.

## Show the help window

Goal level: Sea level

Actor: User

Pre-condition: The application is running.

Main Success Scenario:

1. User accesses the Help functionality.
2. System presents the manual of the application.

## Exit application

Goal level: Sea level

Actor: User

Pre-condition: The application is running.

Main Success Scenario:

1. User accesses the close functionality.
2. System closes the application.

Extensions:

1. The currently simulation is not saved.

2a1. System will ask the user whether to save or not save the current simulation.

## Override simulation (Add police, ambulance, firetruck cars)

Goal level: Sea level

Actor: User

Pre-condition: User started simulation.

Main Success Scenario:

1. User selects the ‘override’ functionality.
2. User selects start and end points of “special” cars route.
3. User selects button to start moving of “special” cars.
4. System overrides simulation according to selected route.
5. System displays changed simulation.

Extensions:

1a. It is not possible to create selected moving route.

1a1. System displays message that selected route cannot be created.

1a2. User changes the route or cancels it.

## Relocate crossing

Goal level: Sea level

Actor: User

Pre-condition: Application is at drag state (mode).

Main Success Scenario:

1. User holds their left mouse button over a crossing.
2. User moves their mouse towards a grid slot.
3. User releases the mouse button.
4. System changes the crossing’s position.

## Startup the application

Goal level: Sea level

Actor: User

Pre-condition: None.

Main Success Scenario:

1. User starts the application.
2. System provides the user with the initial state of the application.

## Show simulation result

Goal level: Sea level

Actor: User

Pre-condition: The simulation is completed at least once.

Main Success Scenario:

1. User prompts the system to show the simulation result.
2. System provides with the most recent results of the simulation.

## Select crossing’s component to make changes

Goal level: Sea level

Actor: User

Pre-condition: There is an active crossing.

Main Success Scenario:

1. User clicks on a component on the current active crossing
2. System sets the current active component to the newly selected

## Set current active crossing

Goal level: Sea level

Actor: User

Pre-condition: There is a crossing on the grid.

Main Success Scenario:

1. The user clicks on the editing properties tool from the toolbox.
2. The system highlights the tool.
3. User hovers over a crossing over the grid
4. The system highlights the crossing.
5. User clicks on a crossing from the grid.
6. The system assigns the current active crossing to the newly selected.
7. The system updates the current active component.

## Edit a crosswalk’s pedestrian flow

Goal level: Sea level

Actor: User

Pre-condition: Crosswalk is selected, the application is running and the simulation is not running.

Main Success Scenario:

1. User inputs a new value for the flow.
2. User confirms the new value.
3. System sets the value as the current pedestrian flow.

Extensions:

1a. User sets a negative or a value that is too high.

1a1.System sets the value to the closest possible number.  
 1a2.Case continues with 2.

## Start simulating pedestrians

Goal level: Sea level

Actor: User

Pre-condition: The simulation is running.

Main Success Scenario:

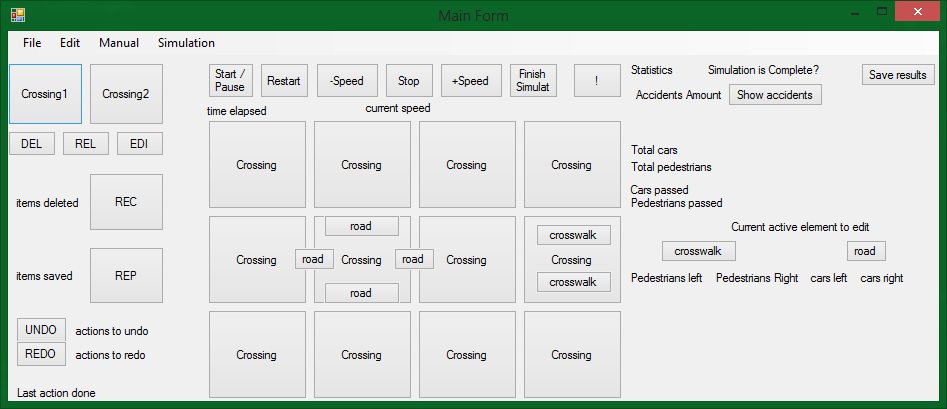
1. User selects to start the pedestrian flow.
2. System shows up moving pedestrians where specified beforehand.

Extensions:

1a. User sets a negative or a value that is too high.

1a1.System sets the value to the closest possible number.  
 1a2.Case continues with 2.

# User interface - initial version, structure and overview



# Non-functional requirements

Of course when using an application the things that can bother us or make us happy are not always related to the product’s functionality. What about Usability, Reliability, Performance and Maintainability?

1. Usability – The application will not be intrusive and will have an easy to understand GUI. The application’s interaction will be intuitive.
2. Performance – This is an important aspect in any Simulation. The application will not take up unnecessary computer memory or computing time. It will match with efficiency standards allowing the simulations to run smoothly.
3. Reliability – All the simulations from the software will be accurate and precise. They will provide with an ideal situation of traffic control that is not influenced by forces not stated by the client. Any outside tempering with the system will be prevented and prohibited.
4. Maintainability – The software will be provided with error protection protocols. The software will be written in proper standards, making it easy to be accessed from third parties.

# Appendix A: Definitions

1. Crossing:

* Representation of real crossroad, displayed on the grid in the application. Crossings are of two types.

1. Road:

* Representation of real road, which is a connection between crossings or connection to crossing.

1. Simulation:

* Simulation is a representation of real situation on the crossroad. Including the cars, the pedestrians and traffic lights. It is represented using the grid.

1. Traffic Flow:

* The amount of car objects which are present on the roads in a certain stimulation (in the application).

1. Pedestrian Flow:

* The amount of pedestrian objects, which are present on a certain stimulation (in the application).

1. Simulation results:

* Graphical representation of statistics. Information about traffic statistics on a certain stimulation.

1. Help menu:

* Option menu, which is present in the application. It gives assisting information to the user about how to use the application

1. Crossing component:  
   - A part of a crossing. Road/Crosswalk
2. Crossing element:

* Crossing component

1. GUI:

* Graphical user interface consisting of all buttons, text and images present in the application

# Appendix B: MOSCOW

Below there is a clear explanation of the priorities of the software application. Provided is a MOSCOW list which includes Must-s, Should-s and Could-s that we will implement. Use-case names are used to point the functionality aspects described in the list.

### Must-s:

* + 1. Select a crossing to place
    2. Place a crossing
    3. Remove a crossing
    4. Create a simulation
    5. Edit a road traffic flow
    6. Start a simulation
    7. Stop a simulation
    8. Exit application
    9. Startup the application
    10. Select crossing’s component to make changes
    11. Set current active crossing
    12. Edit a crosswalk’s pedestrian flow
    13. Save a simulation
    14. Load a simulation
    15. Pause a simulation
    16. Show simulation results

### Should-s:

1. Save simulation results
2. Relocate crossing
3. Restart a simulation

### Could-s:

1. Undo an action
2. Redo an action
3. Show the help window
4. Override simulation (Add police, ambulance, firetruck cars)