

test plan

First Version



Bilger yahov

oLEKSANDR SUPRUNENKO

ILIA NIKUSHEV

GEORGI CHISHIRKOV

LYUBOMIR DIMOV

MENGCHUAN LIU

Traffic Lights System

Table of Contents

[1. Introduction 1](#_Toc446009893)

[1.1. Purpose 1](#_Toc446009894)

[1.2. Project Overview 1](#_Toc446009895)

[2. Test Strategy 1](#_Toc446009896)

[2.1. Test objectives 1](#_Toc446009897)

[2.2. Test Principles 1](#_Toc446009898)

[2.3. Functional Test 1](#_Toc446009899)

[2.3.1. Select a crossing to place 2](#_Toc446009900)

[2.3.2. Place a crossing 2](#_Toc446009901)

[2.3.3. Remove a crossing 3](#_Toc446009902)

[2.3.4. Create a simulation 3](#_Toc446009903)

[2.3.5. Save a simulation 4](#_Toc446009904)

[2.3.6. Load a simulation 4](#_Toc446009905)

[2.3.7. Edit a road traffic flow 5](#_Toc446009906)

[2.3.8. Start a simulation 5](#_Toc446009907)

[2.3.9. Stop a simulation 6](#_Toc446009908)

[2.3.10. Pause a simulation 6](#_Toc446009909)

[2.3.11. Restart a simulation 6](#_Toc446009910)

[2.3.12. Undo an action 6](#_Toc446009911)

[2.3.13. Redo an action 7](#_Toc446009912)

[2.3.14. Save simulation results 7](#_Toc446009913)

[2.3.15. Show the help window 8](#_Toc446009914)

[2.3.16. Exit application 8](#_Toc446009915)

[2.3.17. Override simulation (Add police, ambulance, firetruck cars). 8](#_Toc446009916)

[2.3.18. Relocate crossing 9](#_Toc446009917)

[2.3.19. Startup the application 10](#_Toc446009918)

[2.3.20. Show simulation result 10](#_Toc446009919)

[2.3.21. Select crossing’s component to make changes 11](#_Toc446009920)

[1.4.22. Set current active crossing 12](#_Toc446009946)

[1.4.23. Start simulating pedestrian 12](#_Toc446009947)

[2.4.24. Access “Saved” crossings 13](#_Toc446009977)

[2.4.25. Access “Removed” crossings 13](#_Toc446009978)

[2.4.26. Empty the recycle bin 14](#_Toc446009979)

# **Introduction**

## **Purpose**

This test plan describes the testing approach and overall framework that will drive the testing of the Traffic Lights System. The document introduces:

* Test Strategy: rules the test will be based on, including the givens of the project; description of the process to set up a valid test.

## **Project Overview**

Traffic Lights System is a tool provided with the necessary means to regulate the traffic within Csharp city in order to prevent traffic accidents. By having control over the traffic system, the mayor’s desire of safe city will be satisfied.

The traffic simulation program can simulate different traffic situations within the city. The simulations can be adjusted for different scenarios and will provide with accurate results that can be related to real life.

# **Test Strategy**

## **Test objectives**

The objective of the test is to verify that the functionality of Traffic Lights System works according to the specifications.

## **Test Principles**

* Testing will be focused on meeting the business objectives, cost efficiency, and quality.
* There will be common, consistent procedures for all teams supporting testing activities.
* Testing processes will be well defined, yet flexible, with the ability to change as needed.
* Testing will be a repeatable, quantifiable, and measurable activity.

## **Functional Test**

Functional testing will be performed to check the functions of application. The functional testing is carried out by feeding the input and validates the output from the application.

### **Select a crossing to place**

**Purpose:** The purpose of this test is to select a crossing.

**Pre-condition:** Simulation is not running.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Select a crossing to place. | 1. User enters selecting mode 2. user selects a crossing type 1, type 2 or type 3. | System updates the current selected crossing type and the GUI. |  |

**Pre-condition:** Simulation is not running.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Select a crossing to place. | 1. User try to select the crossing without entering selecting mode | System does nothing. |  |

### **Place a crossing**

**Purpose:** The purpose of this test is to place the selected crossing.

**Pre-condition:** Simulation is not running and user just finishes selecting type 1.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Place the selected crossing. | 1. User enters placing mode. 2. User positions mouse over the 1\*1 slot. 3. User clicks to place the crossing on the grid. | System changes the 1\*1 grid slot to be with the type 1 crossing. |  |

**Pre-condition:** Simulation is not running and user just finishes selecting type 1 crossing.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Place the selected crossing. | 1. User enters placing mode. 2. User positions mouse outside a grid slot. 3. User clicks to place the crossing. | User is still on the mode to place the type 1 crossing, nothing changed. |  |

**Pre-condition:** Simulation is not running, user just finishes selecting a crossing and the 1\*1 slot has type 2 crossing.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Place the selected crossing. | 1. User enters placing mode. 2. User positions mouse over the 1\*1 slot. 3. User clicks to place the crossing on the 1\*1 slot. | System asks user whether to replace the type 2 crossing or not.  If user chooses yes, System changes the 1\*1 slot to be with the type 1 crossing.  If user chooses no, nothing changed. |  |
|  |

### **Remove a crossing**

**Purpose:** The purpose of this test is to remove a crossing.

**Pre-condition:** Simulation is not running and there is a type 1 crossing on 1\*1 slot.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Remove a crossing. | 1. User right clicks on the type 1 crossing from 1\*1 slot. 2. User prompts system to delete the crossing | System removes the type 1 crossing from the 1\*1 slot and places it into the recycle bin. |  |

**Pre-condition:** Simulation is not running and there is a type 1 crossing on 1\*1 slot.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Remove a crossing. | 1. User left clicks or double clicks on the type 1 crossing from 1\*1 slot. | There is no effort. |  |

**Pre-condition:** Simulation is not running and there is a type 1 crossing on 1\*1 slot.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Remove a crossing. | 1. User try to use del from keyboard, or drags the crossing from 1\*1 slot to recycle bin. | There is no effort. |  |

### **Create a simulation**

**Purpose:** The purpose of this test is to create a simulation.

**Pre-condition:** The application is running and all the changes have saved.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Create a simulation. | 1. User clicks File -> New Simulation to create a new simulation. | System empties the grid and creates a new one. |  |
| System does nothing if the current grid is empty. |  |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Create a simulation. | 1. clicks File -> New Simulation to create a new simulation with some changes not saved. | System asks user whether to save or not.  If yes, system saves and creates a new one.  If no, system creates a new one. |  |
|  |

### **Save a simulation**

**Purpose:** The purpose of this test is to save a simulation.

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Save a simulation. | 1. User clicks on File -> Save to save the current simulation. 2. User specifies file name which is never used before and location. | System saves the simulation. |  |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Save a simulation. | 1. User prompts the system to save the current simulation. 2. User try to save the simulation with an existing name in the selected folder. | System asks user whether to replace the existing file or not.  If yes, system replaces the existing simulation.  If no, system asks user to enter a new name. |  |
|  |

### **Load a simulation**

**Purpose:** The purpose of this test is to load a simulation.

**Pre-condition:** The application is running and current changes have saved.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Load a simulation. | 1. User prompts the system to load an exist simulation. 2. User specifies file name and location. | System loads the simulation. |  |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Load a simulation. | 1. User prompts the system to load a simulation without saving their changes. | System asks user whether to save or not.  If yes, system saves and loads an existing simulation.  If no, system and loads an existing simulation, |  |
|  |

### **Edit a road traffic flow**

**Purpose:** The purpose of this test is to edit the flow of a crossroad.

**Pre-condition:** A crossing is selected as an active component (in an active mode) and the simulation is not running.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Edit a road traffic flow. | 1. User inputs a positive value for the car flow. 2. User clicks Update. | System sets the number as the current flow. |  |

**Pre-condition:** A road is selected as an active component (in an active mode) and the simulation is not running.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Edit a road traffic flow. | 1. User inputs a negative value or non-digit value for car flow 2. User clicks Update. | System gives an error message and asks user to re-enter a positive value. |  |

### **Start a simulation**

**Purpose:** The purpose of this test is to start a simulation.

**Pre-condition:** The application is running, the simulation isn’t running and all the crossings are well connected with each other.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Start a simulation. | 1. User accesses the start functionality. | System starts the execution of the simulation. |  |

**Pre-condition:** The application is running, the simulation isn’t running and there are only two crossings on 1\*1 slot and 9\*9 slot without any connections.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Start a simulation. | 1. User accesses the start functionality. | System gives an error message asking user to connect all crossings. |  |

### **Stop a simulation**

**Purpose:** The purpose of this test is to stop a simulation.

**Pre-condition:** Simulation is running.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Stop a simulation. | 1. User accesses the stop functionality. | System stops the execution of the simulation. |  |

### **Pause a simulation**

**Purpose:** The purpose of this test is to pause a simulation.

**Pre-condition:** Simulation is running.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Pause a simulation. | 1. User accesses the pause functionality. | System pauses the execution of the simulation. |  |

### **Restart a simulation**

**Purpose:** The purpose of this test is to restart a simulation.

**Pre-condition:** Simulation is running.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Restart a simulation. | 1. User accesses the restart functionality. | System restarts the execution of the simulation. |  |

### **Undo an action**

**Purpose:** The purpose of this test is to undo the last action.

**Pre-condition:** Simulation is not running, there was a type 1 crossing on 1\*1 slot and user deleted it from the previous step.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Undo an action. | 1. User accesses the undo functionality. | System restores the deleted type 1 crossing on 1\*1 slot and the crossing is deleted from recycle bin. |  |

### **Redo an action**

**Purpose:** The purpose of this test is to redo the last action.

**Pre-condition:** The Undo functionality was used (there is action to redo)

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Redo an action. | 1. User accesses the redo functionality. | System restores the previous state of the application before the action was undone. |  |

**Pre-condition:** The Undo functionality was not used (there is no action to redo)

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Redo an action. | 1. User cannot accesses the redo functionality. | System does not react in any way |  |

### **Save simulation results**

**Purpose:** The purpose of this test is to save simulation results.

**Pre-condition:** The application is started

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Save simulation results. | 1. User accesses the save simulation functionality. 2. User selects format of the file with results and file path. | System saves results in selected format at selected file path. |  |

### **Show the help window**

**Purpose:** The purpose of this test is to view manual of the application.

**Pre-condition:** The application is started

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Show the help window. | 1. User accesses the Help functionality. | System presents the manual of the application. |  |

### **Exit application**

**Purpose:** The purpose of this test is to exit the application.

**Pre-condition:** There were no created simulations/ created simulations were saved

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Exit application. | 1. User accesses the close functionality. | System closes the application. |  |

**Pre-condition:** Created simulations were not saved

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Exit application. | 1. User accesses the close functionality. 2. User selects to save simulation. | System saves simulation and closes the application. |  |

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

**Purpose:** The purpose of this test is to override simulation.

**Pre-condition:** Simulation is not running

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Override simulation. | 1. User selects the ‘override’ functionality. 2. User selects start and end points of “special” cars route. 3. User prompts system to simulate moving of “special” cars. | System displays changed simulation. |  |

**Pre-condition:** Simulation is running

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Override simulation. | 1. User selects the ‘override’ functionality. 2. User selects start point of “special” cars route. | System notifies user that simulation must be paused |  |

### **Relocate crossing**

**Purpose:** The purpose of this test is to relocate crossing on the grid.

**Condition:** User selects available (free) slot on the grid

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Relocate a crossing. | 1. User holds their left mouse button over a crossing. 2. User moves their mouse towards a desired grid slot. | System changes the crossing’s position. |  |

**Condition:** User selects unavailable slot (there is crossing) on the grid

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Relocate a crossing. | 1. User holds their left mouse button over a crossing. 2. User moves their mouse towards a desired grid slot. | System notifies user that crossing cannot be relocated to selected slot |  |

**Condition:** User selects space outside the grid

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Relocate a crossing. | 1. User holds their left mouse button over a crossing. 2. User moves their mouse towards a desired grid slot. | System notifies user that crossing cannot be located outside the grid |  |

### **Startup the application**

**Purpose:** The purpose of this test is to startup the application and display initial

state

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Startup the application | 1. User makes double click on the application’s icon. | System provides the user with the initial state of the application. |  |
| 1. User right clicks on the application’s icon. 2. User prompts system to open application. | System provides the user with the initial state of the application. |  |

### **Show simulation result**

**Purpose:** The purpose of this test is to view results of the simulation.

**Pre-condition:** There is finished simulation and no currently active simulation

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Show simulation result. | 1. User prompts system to show results of the simulation. | System provides with the most recent results of the simulation. |  |

**Pre-condition:** There is no finished simulation and no currently active simulation

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Show simulation result. | 1. User prompts system to show results of the simulation. | System notifies user that there are no results to show |  |

**Pre-condition:** There is no finished simulation and there is currently active

simulation

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Show simulation result. | 1. User prompts system to show results of the simulation. | System notifies user that results cannot be shown before simulation is finished |  |

**Pre-condition:** There is finished simulation and there is currently active

simulation

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Show simulation result. | 1. User prompts system to show results of the simulation. | System notifies user that results cannot be shown before simulation is finished |  |

### **Select crossing’s component to make changes**

**Purpose:** The purpose of this test is to change component of the crossing.

**Pre-condition:** There is no selected crossing’s element

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Select crossing’s component to make changes. | 1. User selects crossing. 2. User selects component to change. | System applies the changes. |  |

**Pre-condition:** There is selected crossing’s element and user made some changes

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Select crossing’s component to make changes. | 1. User selects component to change. | System notifies user that other element was changed and displays which actions are possible |  |



### **Set current active crossing**

**Purpose:** The purpose of this test is to set current active crossing.

**Pre-condition:** There is (no) active crossing

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Set current active crossing. | 1. User selects the editing properties tool from the toolbox. 2. User hovers over a crossing over the grid 3. User clicks on a crossing from the grid. | System updates the current active component. |  |

### **Start simulating pedestrian**

**Purpose:** The purpose of this test is to simulate pedestrian’s flow.

**Pre-condition:** All crossings are located properly on the grid (there are no free

cells between crossings)

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Start simulating pedestrians. | 1. User prompts system to simulate pedestrian flow. | System starts moving the pedestrians on the places specified. |  |

**Pre-condition:** Crossings are located improperly on the grid (there are free

cells between crossings)

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Start simulating pedestrians. | 1. User prompts system to simulate pedestrian flow. | System notifies user that it is not possible to start simulations and informs where the error occurred |  |



### **Access “Saved” crossings**

**Purpose:** The purpose of this test is to view saved crossings

**Pre-condition:** There are saved crossings.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Access “Saved” crossings. | 1. User prompts system show saved crossings. | System shows all previously saved crossings. |  |

**Pre-condition:** There are no saved crossings.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Access “Saved” crossings. | 1. User prompts system show saved crossings. | System notifies user that there are no saved crossings |  |

### **Access “Removed” crossings**

**Purpose:** The purpose of this test is to view crossings which were removed.

**Pre-condition:** There are removed crossings.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Access “Removed” crossings. | 1. User prompts system show removed crossings. | System shows all previously removed crossings. |  |

**Pre-condition:** There are no removed crossings.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Access “Removed” crossings. | 1. User prompts system show removed crossings. | System notifies user that there are no removed crossings |  |

### **Empty the recycle bin**

**Purpose:** The purpose of this test is to view crossings which were removed.

**Pre-condition:** There are removed crossings in recycle bin.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Access “Removed” crossings. | 1. User prompts system to remove all crossings from the recycle bin. | System removes all items from the bin. |  |

**Pre-condition:** Recycle bin is empty.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Result** |
| Access “Removed” crossings. | 1. User cannot select feature to remove crossings from recycle bin. | System does not react |  |