

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](#_Toc445236041)

[**1.1.** **Purpose** 1](#_Toc445236042)

[**1.2.** **Project Overview** 1](#_Toc445236043)

[**1.3.** **Audience** 1](#_Toc445236044)

[**2.** **Test Strategy** 2](#_Toc445236045)

[**2.1.** **Test objectives** 2](#_Toc445236046)

[**2.2.** **Test Assumptions** 2](#_Toc445236047)

[**2.3.** **Test Principles** 2](#_Toc445236048)

[**2.4.** **Functional Test** 2](#_Toc445236049)

[**2.4.1.** **Select a crossing to place** 2](#_Toc445236050)

[**2.4.2.** **Place a crossing** 3](#_Toc445236051)

[**2.4.3.** **Remove a crossing** 3](#_Toc445236052)

[**2.4.4.** **Create a simulation** 4](#_Toc445236053)

[**2.4.5.** **Save a simulation** 4](#_Toc445236054)

[**2.4.6.** **Load a simulation** 5](#_Toc445236055)

[**2.4.7.** **Edit a road traffic flow** 5](#_Toc445236056)

[**2.4.8.** **Start a simulation** 6](#_Toc445236057)

[**2.4.9.** **Stop a simulation** 6](#_Toc445236058)

[**2.4.10.** **Pause a simulation** 7](#_Toc445236059)

[**2.4.11.** **Restart a simulation** 7](#_Toc445236060)

[**2.4.12.** **Undo an action** 8](#_Toc445236061)

[**2.4.13.** **Redo an action** 8](#_Toc445236062)

[**2.4.14.** **Save simulation results** 9](#_Toc445236063)

[**2.4.15.** **Show the help window** 9](#_Toc445236064)

[**2.4.16.** **Exit application** 10](#_Toc445236065)

[**2.4.17.** **Override simulation (Add police, ambulance, firetruck cars)** 10](#_Toc445236066)

[**2.4.18.** **Relocate crossing** 11](#_Toc445236067)

[**2.4.19.** **Startup the application** 11](#_Toc445236068)

[**2.4.20.** **Show simulation** 12](#_Toc445236069)

[**2.4.21.** **Select crossing’s component to make changes** 12](#_Toc445236070)

[**2.4.22.** **Set current active crossing** 13](#_Toc445236071)

[**2.4.23.** **Edit a crosswalk’s pedestrian flow** 13](#_Toc445236072)

[3. Test Environment 14](#_Toc445236073)

# 

# **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 provide with the necessary tools to regulate the traffic within Csharp city to prevent traffic accidents. By having control over the traffic system the mayor’s desire of safe city will be satisfied.

The functionality of the traffic stimulation program that 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.

## **Audience**

* Project team members perform tasks specified in this document, and provide input on this document.
* Project Manager Plans for the testing activities in the overall project schedule, reviews the document, tracks the performance of the test according to the task herein specified, approves the document and is accountable for the results.

# **Test Strategy**

## **Test objectives**

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

The final product of the test is twofold:

* A production-ready software;
* A set of stable test scripts that can be reused for Functional test execution.

## **Test Assumptions**

**Key Assumptions**

* Production like data required and be available in the system prior to start of Functional Testing.

**General**

* Exploratory Testing would be carried out once the build is ready for testing.
* Performance testing is not considered for this estimation.
* The project will provide test planning, test design and test execution support.
* There is no environment downtime during test due to outages or defect fixes.
* The system will be treated as a black box; if the information shows correctly online and in the reports, it will be assumed that the database is working properly.

**Functional Testing**

* During Functional testing, tester will use preloaded data which is available on the system at the time of execution.

## **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 the crossing.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Success?** |
| Select a crossing to place | 1. Select a crossing (type 1, type 2) | System updates the current selected crossing type and the GUI |  |

### **Place a crossing**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Success?** |
| Place a crossing | 1. User positions mouse over a grid slot 2. User clicks to place the crossing on the grid. | System changes the specified grid slot to be with the specified crossing type. |  |
| 1. User positions mouse over a grid slot 2. User clicks to place the crossing on the occupied grid. 3. User removes the current crossing | System overrides the crossing. |  |
| 1. User positions mouse over a grid slot 2. User clicks to place the crossing on the occupied grid. 3. User moves the mouse away from the grid. | System updates the GUI. |  |

### 

### **Remove a crossing**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Success?** |
| Remove a crossing | 1. User drags a crossing from the grid into the recycle bin. | System removes it from the grid and places it into the recycle bin. |  |
| 1. User drags a crossing from the grid into the recycle bin. 2. User ends drag before on top of recycle bin. | System does nothing, crossing remains in place. |  |

### **Create a simulation**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Success?** |
| Create a simulation | 1. User prompts the system to create a new simulation. | System empties the grid. |  |
| 1. User prompts the system to create a new simulation. 2. User saves their changes. | System empties the grid. |  |

### **Save a simulation**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Success?** |
| Save a simulation | 1. User prompts the system to save the current simulation. 2. User specifies file name and location. | System saves the simulation. |  |
| 1. User prompts the system to save the current simulation. 2. User specifies file name and location. 3. User replace the existed file with same name, or user rename the current file | System saves the simulation. |  |

### **Load a simulation**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Success?** |
| Load a simulation | 1. User prompts the system to load an exist simulation. 2. User specifies file location. | System load the simulation. |  |
| 1. User save or close without save the current simulation. 2. User prompts the system to load an exist simulation. 3. User specifies file location. | System overrides the crossing. (pass) |  |

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

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Success?** |
| Edit a road traffic flow | 1. User inputs the new flow. 2. User confirms new value. | System sets the number as the current flow. |  |
| 1. User inputs the new flow 2. User confirms new value. 3. User reenter the value. 4. User confirms again. | System sets the number as the current flow. |  |

### **Start a simulation**

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

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

### **Stop a simulation**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Success?** |
| 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Success?** |
| 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Success?** |
| 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Success?** |
| Select a crossing to place | 1. User accesses the undo functionality | System restores the previous state of the application before the action was performed. |  |
| 1. User stops the current simulation 2. User accesses the undo functionality | System restores the previous state of the application before the action was performed. |  |

### **Redo an action**

**Purpose:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Success?** |
| Select a crossing to place | 1. Select a crossing (type 1 or type 2) 2. Drag crossing from interface to the cell (type 1 or type 2) 3. System adds crossing on that cell. | System adds a crossing to the selected cell. (pass) |  |
| 1. Select a crossing (type 1 or type 2) 2. Select the cell which is already occupied 3. System shows a message. 4. Tester selects yes. | System overrides the crossing. (pass) |  |

### **Save simulation results**

**Purpose:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Success?** |
| Select a crossing to place | 1. Select a crossing (type 1 or type 2) 2. Drag crossing from interface to the cell (type 1 or type 2) 3. System adds crossing on that cell. | System adds a crossing to the selected cell. (pass) |  |
| 1. Select a crossing (type 1 or type 2) 2. Select the cell which is already occupied 3. System shows a message. 4. Tester selects yes. | System overrides the crossing. (pass) |  |

### **Show the help window**

**Purpose:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Success?** |
| Select a crossing to place | 1. Select a crossing (type 1 or type 2) 2. Drag crossing from interface to the cell (type 1 or type 2) 3. System adds crossing on that cell. | System adds a crossing to the selected cell. (pass) |  |
| 1. Select a crossing (type 1 or type 2) 2. Select the cell which is already occupied 3. System shows a message. 4. Tester selects yes. | System overrides the crossing. (pass) |  |

### **Exit application**

**Purpose:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Success?** |
| Select a crossing to place | 1. Select a crossing (type 1 or type 2) 2. Drag crossing from interface to the cell (type 1 or type 2) 3. System adds crossing on that cell. | System adds a crossing to the selected cell. (pass) |  |
| 1. Select a crossing (type 1 or type 2) 2. Select the cell which is already occupied 3. System shows a message. 4. Tester selects yes. | System overrides the crossing. (pass) |  |

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

**Purpose:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Success?** |
| Select a crossing to place | 1. Select a crossing (type 1 or type 2) 2. Drag crossing from interface to the cell (type 1 or type 2) 3. System adds crossing on that cell. | System adds a crossing to the selected cell. (pass) |  |
| 1. Select a crossing (type 1 or type 2) 2. Select the cell which is already occupied 3. System shows a message. 4. Tester selects yes. | System overrides the crossing. (pass) |  |

### **Relocate crossing**

**Purpose:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Success?** |
| Select a crossing to place | 1. Select a crossing (type 1 or type 2) 2. Drag crossing from interface to the cell (type 1 or type 2) 3. System adds crossing on that cell. | System adds a crossing to the selected cell. (pass) |  |
| 1. Select a crossing (type 1 or type 2) 2. Select the cell which is already occupied 3. System shows a message. 4. Tester selects yes. | System overrides the crossing. (pass) |  |

### **Startup the application**

**Purpose:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Success?** |
| Select a crossing to place | 1. Select a crossing (type 1 or type 2) 2. Drag crossing from interface to the cell (type 1 or type 2) 3. System adds crossing on that cell. | System adds a crossing to the selected cell. (pass) |  |
| 1. Select a crossing (type 1 or type 2) 2. Select the cell which is already occupied 3. System shows a message. 4. Tester selects yes. | System overrides the crossing. (pass) |  |

### **Show simulation**

**Purpose:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Success?** |
| Select a crossing to place | 1. Select a crossing (type 1 or type 2) 2. Drag crossing from interface to the cell (type 1 or type 2) 3. System adds crossing on that cell. | System adds a crossing to the selected cell. (pass) |  |
| 1. Select a crossing (type 1 or type 2) 2. Select the cell which is already occupied 3. System shows a message. 4. Tester selects yes. | System overrides the crossing. (pass) |  |

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

**Purpose:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Success?** |
| Select a crossing to place | 1. Select a crossing (type 1 or type 2) 2. Drag crossing from interface to the cell (type 1 or type 2) 3. System adds crossing on that cell. | System adds a crossing to the selected cell. (pass) |  |
| 1. Select a crossing (type 1 or type 2) 2. Select the cell which is already occupied 3. System shows a message. 4. Tester selects yes. | System overrides the crossing. (pass) |  |

### **Set current active crossing**

**Purpose:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Success?** |
| Select a crossing to place | 1. Select a crossing (type 1 or type 2) 2. Drag crossing from interface to the cell (type 1 or type 2) 3. System adds crossing on that cell. | System adds a crossing to the selected cell. (pass) |  |
| 1. Select a crossing (type 1 or type 2) 2. Select the cell which is already occupied 3. System shows a message. 4. Tester selects yes. | System overrides the crossing. (pass) |  |

### **Edit a crosswalk’s pedestrian flow**

**Purpose:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Success?** |
| Select a crossing to place | 1. Select a crossing (type 1 or type 2) 2. Drag crossing from interface to the cell (type 1 or type 2) 3. System adds crossing on that cell. | System adds a crossing to the selected cell. (pass) |  |
| 1. Select a crossing (type 1 or type 2) 2. Select the cell which is already occupied 3. System shows a message. 4. Tester selects yes. | System overrides the crossing. (pass) |  |

### **Start simulating pedestrian**

**Purpose:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Method** | **Expected result** | **Success?** |
| Select a crossing to place | 1. Select a crossing (type 1 or type 2) 2. Drag crossing from interface to the cell (type 1 or type 2) 3. System adds crossing on that cell. | System adds a crossing to the selected cell. (pass) |  |
| 1. Select a crossing (type 1 or type 2) 2. Select the cell which is already occupied 3. System shows a message. 4. Tester selects yes. | System overrides the crossing. (pass) |  |

# Test Environment

Traffic Lights System application will do the simulation.

A windows environment with .Net 3.0+ Framework.