Test Plan

First Version



Group B

Todor Tsekov (2228777)

Wen Zhang (2351420)

Jiaqi Ni (2217774)

Yidi W (2197301)

Kalina Petrova (2221667)

Jianfei Feng (2204529)

Abdulla Al Tuhami (289035)

Class: EI7s1/s2

Course: Project C-phase

Mentor: Bert Van Gestel

Contents

[1.Introduction 3](#_Toc368839600)

2.1[Fig: Test Image A 4](#_Toc368839601)

2.2[Fig: Test Image B 5](#_Toc368839602)

2.3[Fig: Test Image C 6](#_Toc368839603)

3.[Additional Information 7](#_Toc368839604)

3.1Crossings [7](#_Toc368839605)

3.2[States 7](#_Toc368839606)

3.3Traffic light states [7](#_Toc368839606)

[4.Test Action 9](#_Toc368839607)

[5.Test Tables 10](#_Toc368839608)

# 

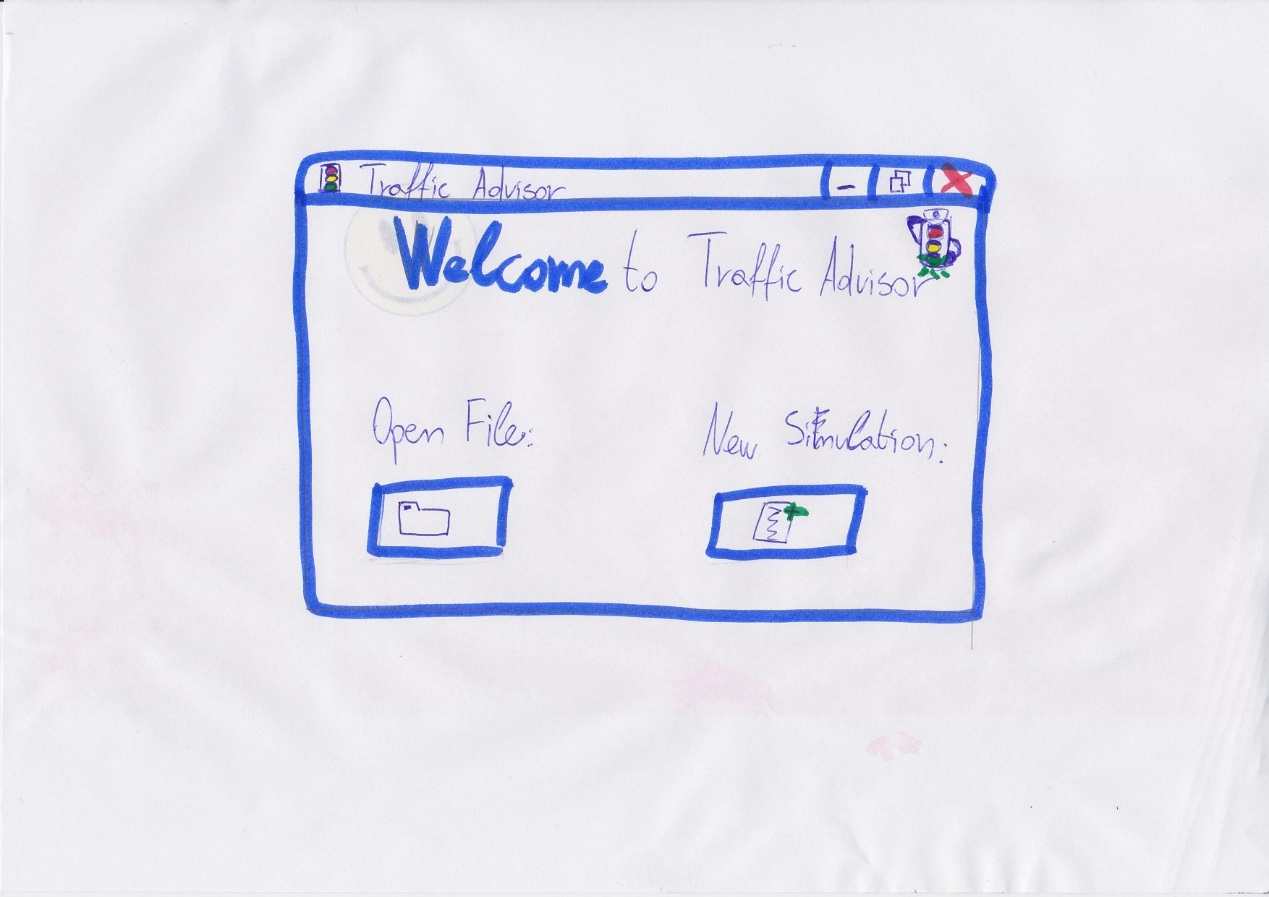
# Introduction

The purpose of this test-plan is to see whether all functionalities described in the use-cases are working correctly and whether certain undesired actions affect the program’s workflow. Instead of constant warning pop-up when a certain action is not permitted we focused on preventing the user from creating mistakes as much as possible, meaning certain functions will be disabled when they’re not supposed to be accessed.

We will conduct a small test with a handful of testers and conduct a final reconfiguration before the final acceptance test with the client.

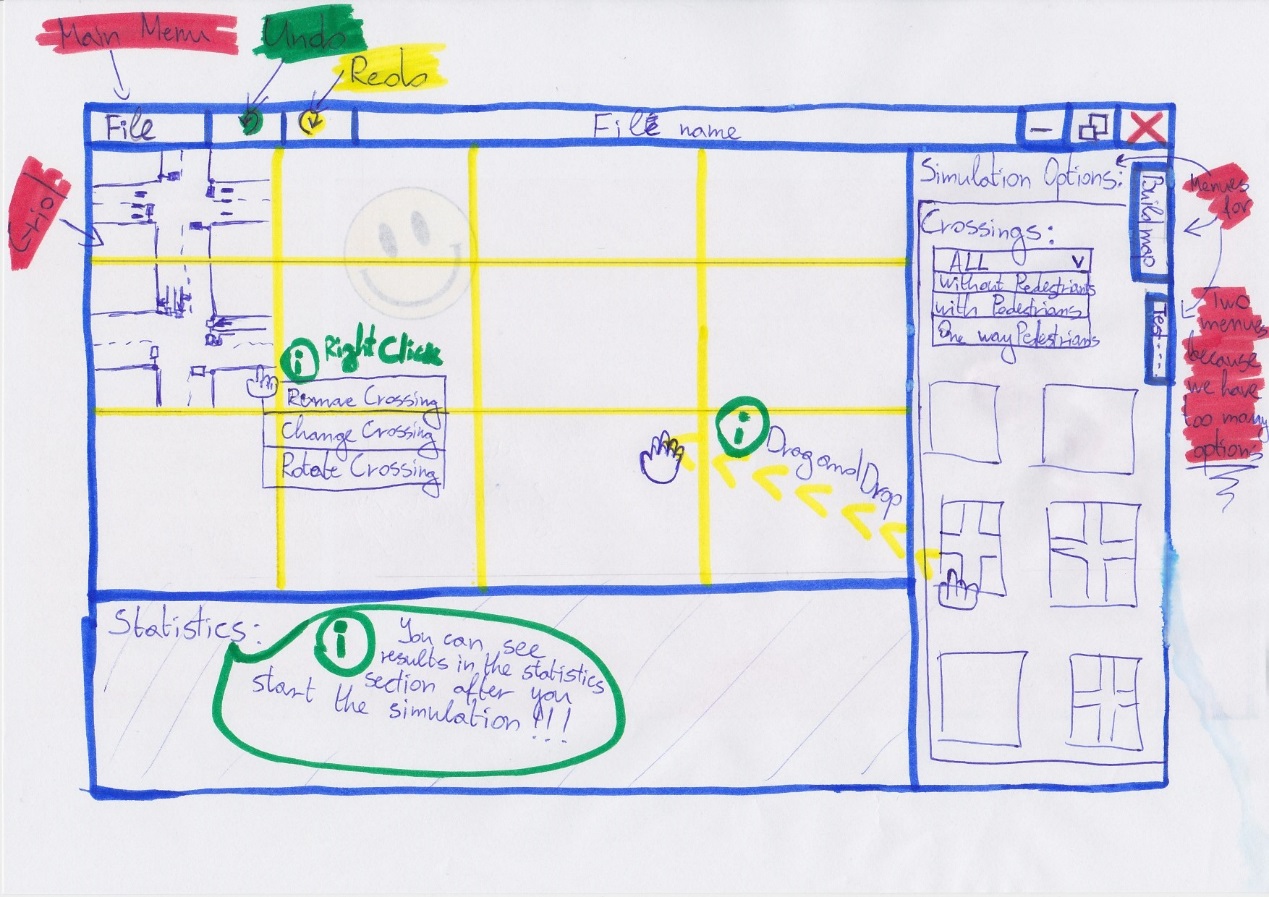
# Fig: Test Image A

Main screen:



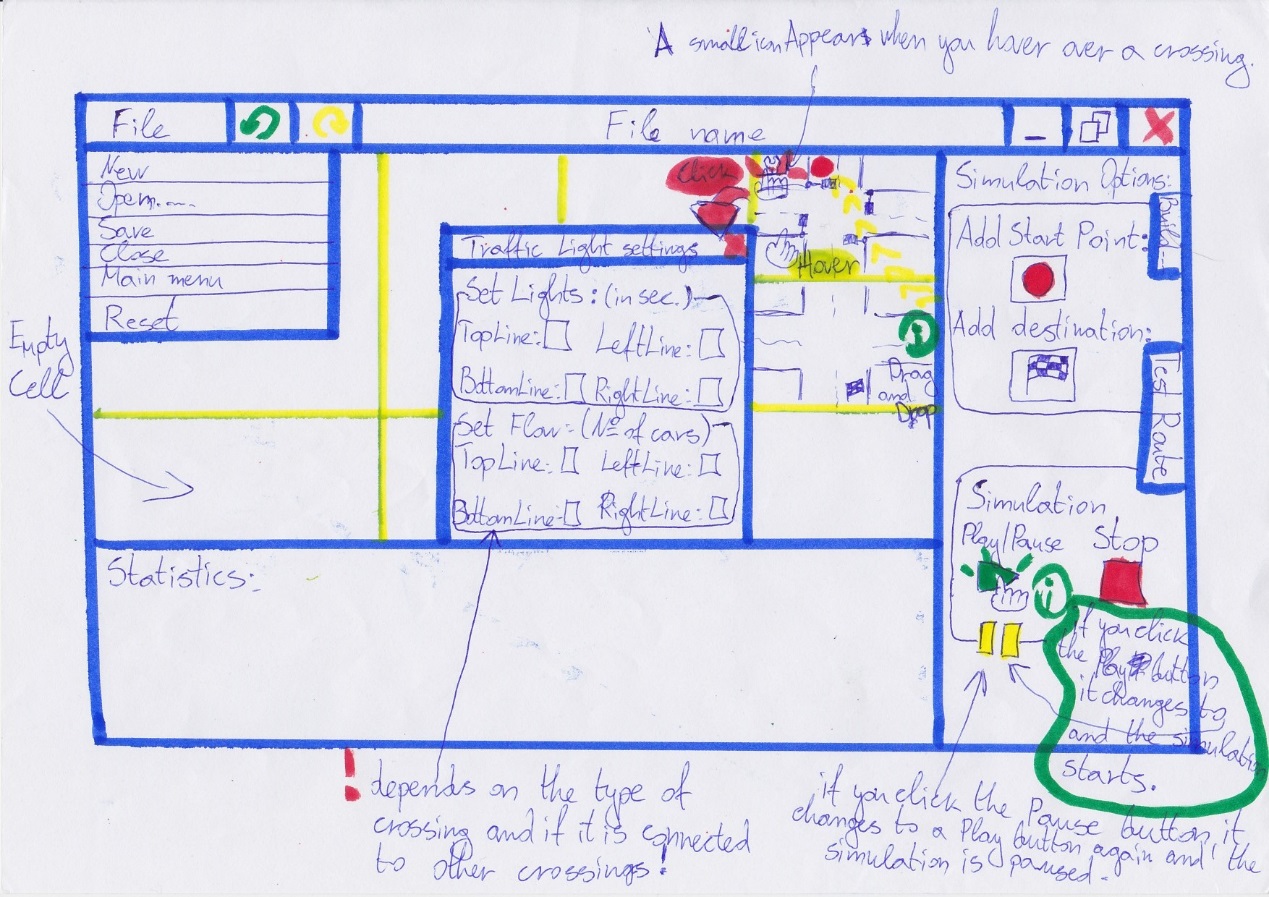
# Fig: Test Image B

## Project-grid screen



# Fig: Test Image C

Project-grid Screen



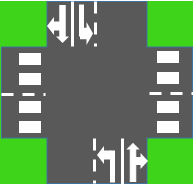
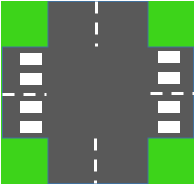
State 1:\* State 2:

State 3:\* State 4:

# Additional information:

# Crossings:

This is our proposal for suitable crossing for the traffic advisor application.



# States:

**Initial state**- where you build the simulation (crossings,etc.)

**Simulation running state**- when you test the simulation

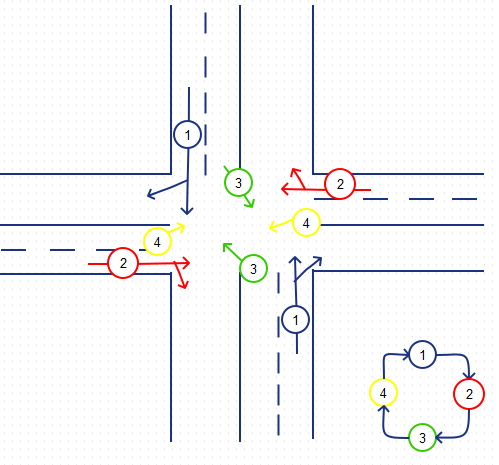
**Pause state**-where you stop to see the progress of the simulation till this point

# Traffic light states:

When we create a crossing, we will automatically create a traffic light system. Each lane of the crossing will have its own traffic light. We will combine traffic lights into groups or states as we call them. Each group will be green at a specific time of the cycle. The user will be able to change only the green time of a specified cycle and not a traffic light by itself. That way we make sure that the user doesn’t make a mistake with his inputs. Refer to the Figure bellow for an example of a traffic light system setup.

**“Change traffic light system setup” specifications:**

When a user wants to change the type of a traffic light system he’d like to use on the specified crossing. For now we have limited the user to a set of traffic light setups for different crossings. Later in the implementation we may be able to let the user create his own groups of traffic lights and create his own states.



So as you can see lanes numbered 1 will go together then they’ll switch with lanes numbered 2 and so on.

# Test Action:

**This test action are based on URS document we already made.**

Scenarios:

* Add crossing.
* Delete crossing.
* Change crossing.
* Rotate crossing.
* Modify traffic Light System.
* Change traffic Light Setup.
* Alter Flow.
* Navigate.
* Play Simulation.
* Pause Simulation.
* Stop Simulation.
* Create New Project.
* Load Project and Statistics.
* Save Project and Statistics.
* Exit Application.
* Go to Main Screen.
* Undo.
* Redo.
* Reset.

# Table content:

## Purpose:

Explains what the purpose of the taken test is. In our case we want to check if we can redirect our user to the main menu screen under certain conditions.

## Target on screen:

The actual screen commands the user will interact with.

## Test Data/Simulation:

Test our actions under different kinds of conditions and with different kinds of data to check if we have captured all the exceptions and if we take necessary precautions to prevent the action from crashing.

## Expected Result:

What is the expected result in each different case we ran the test.

## Actual Result:

The actual result that occurred during the test.

## Outcome and actions required:

Compare the Expected results and the actual results to come to conclusions what kind of actions are to be taken to fix the inaccuracies.

# Test Tables:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Test Name | Purpose | Target on screen | Test Data/Simulation | Expected Result | Actual Result | Outcome and Actions required |
| 1. | Add crossing  \*If user wants to modify the traffic light system, go to the test for use case “modify traffic light system”. | Check if the user can successfully add a crossing without losing any data. | User choose a crossing🡪user drags a crossing to a cell🡪user drop the crossing🡪user chooses one traffic light system🡪 user modify the traffic light system(optional)🡪 system place the crossing in the cell. | 1. When simulation is in “initial state”. 2. When simulation is NOT in “initial state”. 3. User drops the crossing in a blank cell. 4. User drops the crossing not in a cell but in somewhere else. 5. User drops the crossing in a cell which is not black. 6. User gives up the operation halfway. | 1. System will let the user choose one traffic light system. 2. System will show nothing. 3. System will let the user choose the one traffic light system. 4. System will give a proper error message. 5. System will give a proper error message. 6. System cancels the operation and gives a proper message. |  |  |
| 2. | Delete crossing | Check if the user can successfully delete a crossing. | User right click on a crossing 🡪choose the “delete crossing”🡪confirm deletion🡪system deletes the crossing. | 1. When simulation is in “initial state”. 2. When simulation is NOT in “initial state”. 3. User right clicks a cell with a crossing. (in “initial state”) 4. User right clicks a blank cell. (in “initial state”) 5. User cancels the operation when the system is asking confirmation. | 1. System will show the option panel. 2. System will show nothing. 3. System will show the option panel 4. System will show nothing. 5. System does not delete the crossing and give a proper message. |  |  |
| 3. | Change crossing  \* After the user chooses a new crossing, go to the test for use case “add crossing”. | Check if the user can successfully change the setting of a crossing. | User right click on a crossing 🡪choose the “change the crossing”🡪choose a crossing🡪set the traffic light🡪system changes the crossing. | 1. When simulation is in “initial state”. 2. When simulation is NOT in “initial state”. 3. User right clicks a cell with a crossing. (in “initial state”) 4. User right clicks a blank cell. (in “initial state”) 5. User gives up the operation halfway. | 1. System will show the option panel. 2. System will show nothing. 3. System will show the option panel 4. System will show nothing. 5. System cancels the operation and gives a proper message. |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Test Name | Purpose | Target on screen | Test Data/Simulation | Expected Result | Actual Result | Outcome and Actions required |
| 4. | Rotate crossing | 1. Crossing will be rotated 90 degrees counter clockwise 2. The properties of the 4 roads will be swapped.  3. Flow will be altered appropriately as well.  Test all above 3 requirements whether to reach under different simulation states. | Grid->Crossing->Right click menu->Rotate | 1.When simulation is in “Initial state”  2.When simulation is running  3.When simulation is paused | 1. User is able to rotate existing crossing on the grid cell. All 3 requirements will be fulfilled.  2. User is not allowed to rotate crossing. All 3 requirements stay same as previous setting.  3. User is not able to rotate existing crossing on the grid cell. All 3 requirements stay same as previous setting. |  |  |
| 5. | Modify traffic light | 1. Interval times of states will be changed follows rules.  2. Able to change input of state.  Test all above 2 requirements whether to reach under different simulation states. | Hover over a crossing->click left corner output icon->setting window | 1.When simulation is in “Initial state”  2.When simulation is running  3.When simulation is paused | 1. User is able to modify traffic light for certain crossing on the grid cell. All 2 requirements will be fulfilled.  2. User is not allowed to modify traffic light for crossing. All 2 requirements stay same as previous setting.  3. User is not allowed to modify traffic light for crossing. All 2 requirements stay same as previous setting. |  |  |
| 7. | Alter flow for crossing | Test modifying flow whether is available, under different simulation states. | Grid->Crossing->Right click menu->crossing setting->flow setting | 1.When simulation is in “Initial state”  2.When simulation is running  3.When simulation is paused | 1. User is allowed to alter flow for existing crossing on the grid cell.  2. User is not allowed to alter flow.  3. User is not allowed to alter flow. |  |  |
|  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Test Name | Purpose | Target on screen | Test Data/Simulation | Expected Result | Actual Result | Outcome and Actions required |
| 8. | Navigate | User require to change the cars flow in specific route. | Project-grid screen->Navigation button | 1. When simulation is in running state.  2.When simulation is in pause state  3.When simulation is in initial state | 1.Navigation button is unavailable for users  2.Navigation button is unavailable for users  3.User clicks the navigation start point button, then user selects start point ,then clicks the destination point button and selects end point and input the flow numbers, click confirm button and the data saved or back to default value. After save value, system calculates the new data. |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Test Name | Purpose | Target on screen | Test Data/Simulation | Expected Result | Actual Result | Outcome and Actions required |
| 9. | Play Simulation | System simulates the project | Project-grid screen ->Play simulation | 1. When simulation is in running state.  2.When simulation is in pause state  3.When simulation is in initial state | 1. During the running state, play simulation button change to stop simulation button, user can only stop simulation.  2. Play simulation button is available for user to resume system.  3. Play simulation button is available for user to start system. |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Test Name | Purpose | Target on screen | Test Data/Simulation | Expected Result | Actual Result | Outcome and Actions required |
| 10. | Pause Simulation | Pause the simulation played by the system. | Power-grid screen->Pause button. | 1. When simulation is in “Running state”.  2. When simulation is NOT in “Simulation running state”. | 1. The pause button appears on the place of the star button. The user clicks the pause button, the simulation is paused.  The paused button changes to play button again.  2. User will not see the pause button and cannot press it. |  |  |
| 11. | Stop Simulation | Stops the simulation played by the system. | Power-grid screen->Stop button. | 1. When simulation is in “Running state”.  2. When simulation is in “Paused state”.  3. When simulation is in “Initial state”. | 1. User is presses the Stop button .The simulation stops, The system goes to “Initial State”.  2. User is presses the Stop button .The simulation stops, The system goes to “Initial State”.  3. The stop button is disabled. |  |  |
| 12. | Create new project | Creates a new project, which the user can work on. | Main-Screen  ->Create a new project button. | 1. When we start the application.  2. When simulation is in “Running state”/ “Paused state”.  3. When simulation is in “Initial state”. | 1. User is clicks the Create a new project button, a pop-up menu appears, so he/she can choose where to save their file. After browsing the user clicks ok, the project is created, the system is at Project-grid screen and in initial state.  2. User needs to stop the simulation (clicks the stop button).Use point 3.  3. User clicks on the main menu icon. A pop-up menu appears-the user chooses the Create a new project option, a pop-up menu appears, so he/she can choose where to save their file. After browsing the user clicks ok, the project is created, the system is at Project-grid screen and in initial state. |  |  |
| 13. | Load project and statistics | Load a Project. | Main screen->Load button. | 1. When the user starts the application.  2. When simulation is in “Initial state” and saved.  3. When simulation is in “Running state”.  4. When simulation is in “Paused state”. | 1. The file is loaded, the system is at Project-grid screen and in “initial state”.  2. A message will be shown “There has existed a project”.  3. The target button is disabled.  4. A message will be shown “There has existed a project”. |  |  |
| 14. | Save project and statistics | Save to a file. | Power-grid screen->File->save button | 1. When simulation is in “Initial state”.  2. When simulation is in “Running state”.  3. When simulation is in “Paused state”. | 1. The file is automatically saved.  2. User cannot save the project when it is running. The button is disabled.  3. A message shows to ask the user if he’d like to stop the simulation. |  |  |
| 15. | Exit Application | Close a file. | Power-grid screen->File->close button | 1. When simulation is in “Initial state” and saved.  2. When simulation is in “Initial state” and not saved.  3. When simulation is in “Running state”.  4. When simulation is in “Paused state”. | 1. The project is closed.  2. A message shows to notify user that his project has not been saved. Asks user if he’d like to save.  3. The button is disabled.  4. A message shows to ask the user if he’d like to stop the simulation. |  |  |
| 16. | Go to main menu | Check if we can safely redirect the user to the main menu screen without losing data. | File -> Go to “main menu” | 1. When simulation is in “Initial state” and progress is saved.  2. When simulation is in “Initial state” and progress is not saved.  3. When simulation is in “Paused state”.  4. When simulation is in “Running state”. | 1. User is automatically redirected to the “Main Menu” screen.  2. A command prompt pops up to notify user that his project has not been saved. Asks user if he’d like to save.  3. A command prompt pops up asking the user if he’d like to stop the simulation.  4. The target button should be inactive/ inaccessible. |  |  |
| 17. | Undo | Undo the most recent change made on the grid. | Undo button | 1. Perform one change on the grid and undo it.  2. No actions have been performed yet.  3. Perform 1 action then try to undo 2 times.  4. Try undo when simulation is in “Running state”. | 1. When action is performed undo button is activated. When we press it, it deactivates and change has been reverted.  2. Button should be inactive.  3.After the first undo the button should deactivate  4. The button should be inactive/inaccessible. |  |  |
| 18. | Redo | Redoes previously undone change on the grid. | Redo button | 1 .Make a change undo and redo it.  2. Try to redo when nothing has been undone.  3. Undo 1 time and try to redo 2 times.  4. Try to redo when simulation is in “Running state”. | 1. After the redo the change we made is the same.  2. The button should be inactive.  3. After the first redo the button deactivates. The change we have undone is back on the grid.  4. The button should be inactive/inaccessible. |  |  |
| 19. | Reset | Resets the grid to its initial state.(before we start inserting) | Reset button | 1. Make a few changes and reset.  2. Try to reset when nothing was created.  3. Try to reset when simulation is in “Running state”. | 1. The grid reverts back to its initial state/ starting point.(has no crossings)  2. The button should be inactive.  3. The button should be inactive/inaccessible. |  |  |