Group B

User Requirements Specifications

Contents

*Use Cases:*

[Exit 1](#_Toc414275465)

[New File 1](#_Toc414275466)

[Open 2](#_Toc414275467)

[Save crossing 3](#_Toc414275468)

[Increase traffic (sidebar) 3](#_Toc414275469)

[Decrease traffic (sidebar) 4](#_Toc414275470)

[Change State of Traffic light 5](#_Toc414275471)

[Insert/Add Crossing 5](#_Toc414275472)

[Remove Crossing 6](#_Toc414275473)

Requirements and Rules (functional and non-functional):

[Requirements 7](#_Toc414275474)

[Rules 7](#_Toc414275475)

1. Use Cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | 1.1 | | |
| **Use Case Name:** | Exit | | |
| **Date Created:** | 7.3.2015 | |
| **Actors:** | | User | |
| **Description:** | | Closing the program | |
| **Pre-conditions:** | | The program must be open | |
| **Post-conditions:** | | Program is closed | |
| **Normal Flow:** | | * 1. The User clicks on the *X* icon on the program’s window   2. Program is terminated | |
| **Exceptions:**  1.1 The user previously had opened a file and had clicked on the grid area (activated it).  a. System asks the User if he wants to Save the opened file. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | 1.2 | | |
| **Use Case Name:** | New File | | |
| **Date Created:** | 7.3.2015 | |
| **Actors:** | | User | |
| **Description:** | | To create a new file | |
| **Pre-conditions:** | | The program must be open | |
| **Post-conditions:** | | System creates new file | |
| **Normal Flow:** | | 1. User selects option new  2. System creates new file. | |
| **Exceptions:**  None | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | 1.3 | | |
| **Use Case Name:** | Open/Load | | |
| **Date Created:** | 7.3.2015 | |
| **Actors:** | | User | |
| **Description:** | | Open/Load an existing crossing file | |
| **Pre-conditions:** | | The program must be open | |
| **Post-conditions:** | | New crossing file is open | |
| **Normal Flow:** | | 1. The user click on the Open/Load File button 2. System shows a file dialog. 3. The user browses and DoubleClick’s the file of choice 4. The system opens the chosen file | |
| **Exceptions:**  1.1 The user had opened a file already  a. System asks the user if he/she wants to Save the opened file.  3.1 The selected file is not a crossing file.  a. System displays error message and returns to step 2. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | 1.4 | | |
| **Use Case Name:** | Save crossing | | |
| **Date Created:** | 7.3.2015 | |
| **Actors:** | | User | |
| **Description:** | | Save crossing | |
| **Pre-conditions:** | | The program must be open | |
| **Post-conditions:** | | File is saved | |
| **Normal Flow:** | | * + - 1. The user clicks on the Save button       2. System shows a file dialog.       3. The User selects the save destination and names the file. | |
| **Exceptions:**  None | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | 1.5 | | |
| **Use Case Name:** | Increase traffic (sidebar) | | |
| **Date Created:** | 7.3.2015 | |
| **Actors:** | | User | |
| **Description:** | | Increase traffic of all crossings. | |
| **Pre-conditions:** | | Program is running.  At least one crossing must be added.  Traffic simulation must be running. | |
| **Post-conditions:** | | Traffic is increased. | |
| **Normal Flow:** | | 1. User slides option bar to the right.  2. System increases traffic. | |
| **Exceptions:**  1.1 Maximum traffic capacity has been reached.  a. System displays error message returns to step 1. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | 1.6 | | |
| **Use Case Name:** | Decrease traffic (sidebar) | | |
| **Date Created:** | 7.3.2015 | |
| **Actors:** | | User | |
| **Description:** | | Decrease traffic of all crossings | |
| **Pre-conditions:** | | Program is running.  At least one crossing must be added.  Traffic simulation must be running. | |
| **Post-conditions:** | | Traffic is increased. | |
| **Normal Flow:** | | 1. User slides option bar to the left.  2. System decreases traffic. | |
| **Exceptions:**  2.1. When minimum traffic capacity is reached no cars will be visible. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | 1.7 | | |
| **Use Case Name:** | Change State of Traffic light | | |
| **Date Created:** | 7.3.2015 | |
| **Actors:** | | User | |
| **Description:** | | Change the state of the traffic light | |
| **Pre-conditions:** | | Program is running  Traffic simulation must be running | |
| **Post-conditions:** | | Traffic light changed from red/green to green/red | |
| **Normal Flow:** | | 1. User clicks on the sensor  2. System places action in queue  3. System changes state of traffic light | |
| **Exceptions:**  None | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | 1.8 | | |
| **Use Case Name:** | Insert/Add Crossing | | |
| **Date Created:** | 7.3.2015 | |
| **Actors:** | | User | |
| **Description:** | | Insert or add a crossing | |
| **Pre-conditions:** | | Program is running | |
| **Post-conditions:** | | System loads the selected crossing | |
| **Normal Flow:** | | 1. The User clicks on the Crossing icon  2. System shows a window with possible types of crossings  3. User selects the desired crossing and clicks OK | |
| **Exceptions:**  3.1 System shows error message if the maximum number of crossings is reached.  a. User clicks OK( return to step 1) | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | 1.9 | | |
| **Use Case Name:** | Remove Crossing | | |
| **Date Created:** | 7.3.2015 | |
| **Actors:** | | User | |
| **Description:** | | Remove crossing from the opened simulator | |
| **Pre-conditions:** | | Program is running | |
| **Post-conditions:** | | System removes the selected crossing | |
| **Normal Flow:** | | 1. The User clicks on the Remove Crossing button  2. System shows a list with all the current Crossings, sorted by ID  3. User selects the desired crossing and clicks OK | |
| **Exceptions:**  3.1 System shows empty list if no crossings are inserted/added  a. User clicks CANCEL( **Extend:** “Insert/Add Crossing” use case)  3.3 System shows error message if no crossing was selected from the list.   a. User clicks OK( return to step 1). | | | |

1. Requirements and Rules (functional and non-functional)

# 

# Requirements

* Must be possible to add/remove crossings.
* Possibility to adjust the traffic.
* Traffic movement must be visualized real-time on the screen.
* Must be possible to adjust the green time of the traffic lights on each crossing.
* Must be possible to save and load a simulation.
* Must be possible to change the state of the traffic light via a sensor (button).

# 

# Rules

* Minimum number of crossings that can be put is 1.
* Maximum number of crossings that can be put is 12.
* Only two types of crossings will be available.
* The user can add/remove only one crossing at the same time.
* The application will not allow objects to overlap each other.
* Car objects must stop on red light and go on green.
* Car object must move along the crossings.
* Software should not have any major bugs on release.
* The code will have proper comments to improve the documentation and maintainability of the program.

The simplicity of the software’s design will help with responsiveness. In addition, it will enhance the user-friendliness and reduce the time needed to familiarize with the software. While the software itself isn’t portable, as it’s limited to Windows environments, the ability to save or load crossing layouts will at least supplement the portability of the product.

1. MOSCOW

|  |  |  |  |
| --- | --- | --- | --- |
| **Must** | **Should** | **Could** | **Won’t** |
| Add/Remove traffic | Add/Remove desired crossing | Cars simulate crashes | Pedestrians |
| Adjust traffic | Adjust green time | Average waiting time(info) | Be able to change the direction of cars |
| Cars moving | Adjust traffic of desired crossing | Chance to cause jam(info) |  |
| Cars not overlapping | Total cars on crossing(info) | Adjust the speed of cars |  |
| Start/Stop simulation | Save/Load | Adjust the position of a crossing |  |
| Cars stop/go according to the traffic light | Change state of traffic light via a sensor | Adjust the position of the cars |  |
|  |  | Resizing the simulation window |  |