

# USER REQUIREMENTS SPECIFICATIONS

*Hyena Crossing*

Updated On: 20-May-20



---

Version 1.4

# INTRODUCTION

---

This document is going to cover all the required features that the client has requested from our application.

We are going to represent those requirements in a form of use cases.

## TABLE OF CONTENTS

---

<i>Introduction</i>	2
<b>Non-Functional Requirements</b>	3
<b>Functional Requirements</b>	4
<b>User Interface Sketches</b>	5
<i>Use cases</i>	9
<b>Iteration 1</b>	9
Start Simulation	9
Stop Simulation	10
Pause Simulation	10
Generate a report	11
Reset Road Map Layout	12
Add intersection without zebra crossings	13
Remove intersection without zebra crossings	14
Add intersections with zebra crossing	15
Remove intersections with zebra crossing	16
Save a file	16
Save as a file	17
Specify number of cars generated	18
<b>Iteration 2</b>	19
Add roundabouts	19
Remove intersections with roundabout	20
Change simulation speed	20

Simulation timer	21
Simulation speed	22
Add Surprise Elements	23
Enable Special (Emergency) Vehicles	24
Disable Special (Emergency) Vehicles	25
Current Flow	26
Adjust Number of Pedestrians	28
Enable Road Sensors	28
Disable Road Sensors	30

## NON-FUNCTIONAL REQUIREMENTS

---

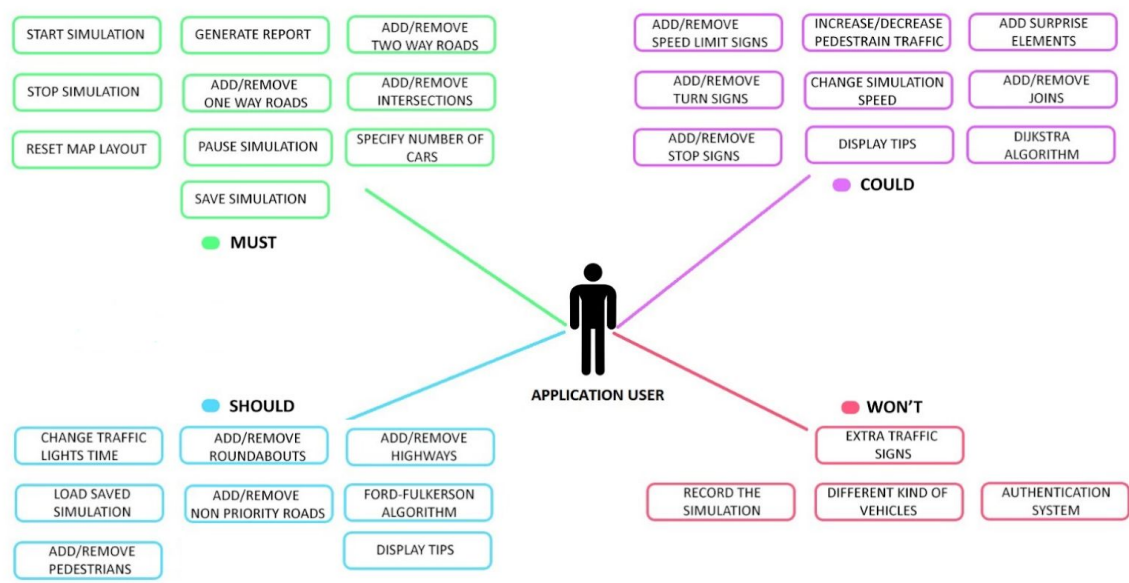
- **Robustness** - the application has to work without errors when executed.
- **Efficiency** - the application has to work smoothly with small response time
- **Usability** - the application needs to have user-friendly ui
- **Readability** - the code of the application has to be structured and the solutions justified
- **Testability** - the application has to be tested with test cases
- **Accessibility** - the application has to work on Windows

# FUNCTIONAL REQUIREMENTS

The functional requirements for our application are shown in this use case diagram.

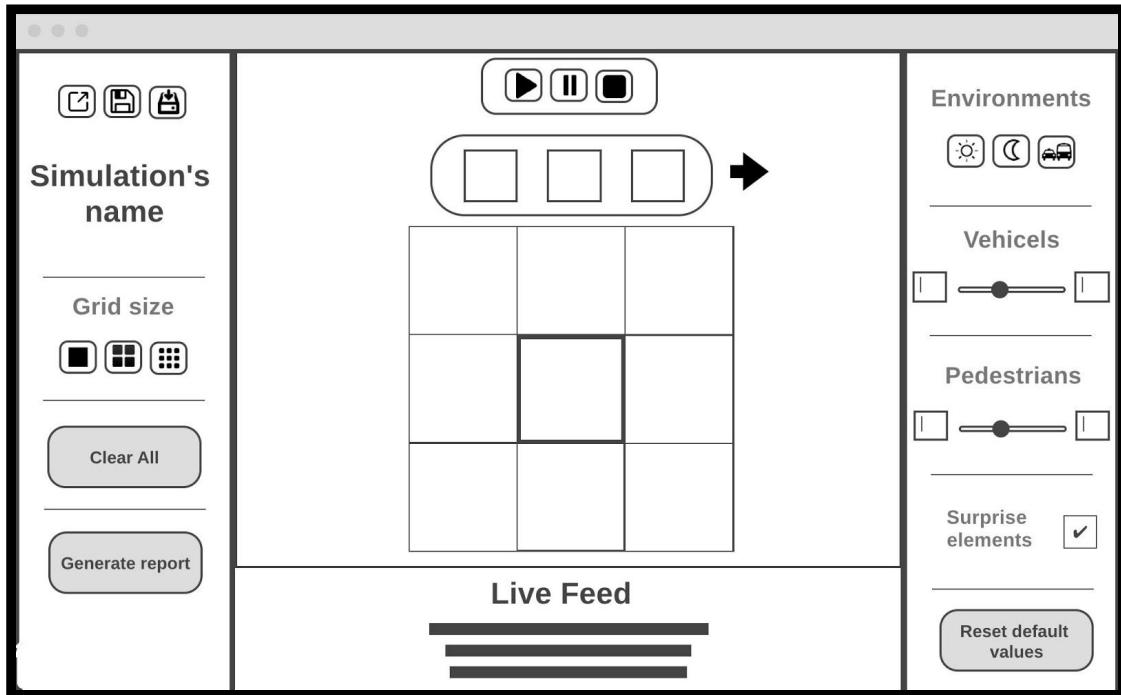
The must features will be explored further in the use cases section.

## USE CASE DIAGRAM



# USER INTERFACE SKETCHES

Our application will have the following layout for the main screen:



This window has 4 panels:

1- File & Grid options (Left panel): From top towards bottom you can see the file options (Open - Save - Save as). Underneath that you can find the simulation's name that the user can edit. Followed by grid size options (Small - Medium - Large). Next you can see a "Clear All" button which removes all elements from the grid. After that you can see the "Generate report" button which displays a report about the simulation that just took place, here's an example of such window:

# Simulation's Name

## Report

---

# Completed

Planned runtime	Actual runtime:
<b>00:45</b>	<b>00:45</b>
Cars entered:	Cars left:
<b>12</b>	<b>10</b>
Pedestrians entered:	Pedestrians left:
<b>8</b>	<b>7</b>
Surprise elements:	Accidents:
<b>0</b>	<b>1</b>

Save

Compare

Close

2- Simulation's main screen (Top-center panel): This is the screen where the actual simulation is going to take place, on top you have "Play - Pause - Stop" buttons. Underneath you have the pieces list window, this window is visible whenever a user clicks on an empty piece of the grid. After that you have the grid on which the user can have the pieces on.

By double clicking on a specific piece, a window form for modifying the settings of the selected piece will be shown on the screen.

The image shows a software window with a title bar at the top. Inside the window, on the left, is a large square grid with an 'X' drawn across it from corner to corner. To the right of the grid, there are two checkboxes: 'Add pedestrian sensors' and 'Add traffict light sensors', both of which are currently unchecked. Below these checkboxes is a section titled 'Traffic light time'. This section contains two small square icons, each with an 'X' inside, positioned to the left of two empty text input boxes. Below the text boxes is an 'Apply' button. At the bottom of the window, there are two buttons: 'Save' and 'Cancel'.

The user will be able to set the green and the red light time per traffic light and add / remove sensors. To change the green or the red time of a traffic light, the user will select the traffic light by clicking on it and inserting in the text boxes the desired amount of seconds for each traffic color. The settings are saved per traffic light by clicking the "Apply" button.

The user can also choose to add pedestrian or traffic light sensors. To save the changes, the user can click the "Save button" on the bottom of the form. The user will be informed if the changes have been successfully applied.

3- Live feed (Bottom-center panel): This window will provide updates/live feed about the simulation while it's taking place.

4- Dynamic elements (Right panel): This panel is for adding/removing/modifying live elements (pedestrians - vehicles - emergency vehicles) to the simulation. On top you can find Pre-set environments with 3 pre-defined environments (Normal day traffic - Normal night traffic - Rush hour), in which the user can click on one and it would modify the parameters to emulate that environment. Afterwards you have the pedestrians and vehicles sliders, which the user can use to specify how many cars/people this simulation is going to have. Next you have the "Surprise elements" checkbox, which adds a percentage of random events taking place on the map to simulate real-life traffic (e.g. An accident happening & ambulance/police cars showing up). Lastly we have the "Reset default values" button, which restores the default parameters of this panel. This panel is also inaccessible while the simulation is taking place.



# USE CASES

---

In this section we're going to list all the required use cases that our application is going to have:

## ITERATION 1

---

### Start Simulation

**ID:** 1

**Name:** Start Simulation

**Goal:** Start the simulation

**Actors:** User

**Description:** The user clicks a button that starts the flow of cars hence the simulation

**Pre-conditions:** The user has constructed a road system by placing roads on the map

**Trigger:** Click of the button "Start Simulation"

**Main Success Scenario:**

1. User click button "Start Simulation"
2. System starts the simulation

**Post-condition:**

1. User can stop the simulation or exit
2. User can pause the simulation

**Extensions:**

**2A: *User restarts the program:***

- Use case ends
-

## Stop Simulation

**ID:** 2

**Name:** Stop Simulation

**Goal:** Stop the simulation

**Actors:** User

**Description:** The user clicks a button that stops the simulation and clears the currently build road system

**Pre-conditions:** The simulation is running

**Trigger:** Clicking the button "Stop Simulation"

**Main Success Scenario:**

1. User clicks button "Stop Simulation"
2. System stops the simulation
3. System resets the road map

**Post-condition:**

1. The simulation is resetted.

**Extensions:**

-3A: *User restarts the program:*

- Use case ends.

---

## Pause Simulation

**ID:** 3

**Name:** Pause Simulation

**Goal:** Stop the simulation for a while/pause the simulation

**Actors:** User

**Description:** The user clicks a button that stops the simulation but does not end it

**Pre-conditions:** The simulation is running

**Trigger:** Clicking the button "Pause Simulation"

**Main Success Scenario:**

1. User clicks button "Pause Simulation"
2. System pauses the simulation without resetting the map

**Post-condition:**

1. User can continue the simulation or stop it

**Extensions:**

-2A: *User restarts the program:*

- Use case ends.

---

## Generate a report

**ID:** 4

**Name:** Generate a report

**Goal:** Generate a report for the user of a simulation that already took place.

**Actors:** User

**Description:** The user is indicating to display a report that contains the stats of the simulation that just took place. The system displays the result.

**Pre-condition:**

1. A simulation has just taken place.
2. The simulation is stopped
3. The simulation had at least one dynamic element.

**Trigger:** Simulation has just ended, and the user indicates to display a report.

**Main success scenario:**

1. User indicates to display a report for the simulation.
2. System displays the report.

**Post-condition:**

- The report is shown, and the user can choose to continue.

**Extensions:**

-1A: *The simulation had no dynamic elements (cars, pedestrians):*

- System displays an error message.
- The use case is over.

---

## Reset Road Map Layout

**ID:** 5

**Name:** Reset road map layout

**Goal:** Remove all the elements off the grid.

**Actors:** User

**Description:** A user can choose to reset a layout displayed on the screen.

**Pre-condition:**

1. A map is already loaded on the grid.
2. The simulation is stopped.

**Trigger:** User wants to remove all elements off the grid.

**Main success scenario:**

- 1- User clicks on “Reset map”
- 2- System displays a verification message
- 3- User chooses to proceed.
- 4- All elements are removed off the grid.

**Postcondition:**

- Map grid is now empty.

**Extensions:**

-3A: *User chooses to cancel:*

- Use case ends.
- 

## **Add intersection without zebra crossings**

**ID:** 10

**Name:** Add intersection

**Goal:** Add an intersection to the map on the grid.

**Actor:** User

**Description:** A user can choose to place an intersection on an empty space on the grid.

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

**Trigger:** User wants to add an intersection to the grid.

**Main success scenario:**

1. System shows the available intersections.
2. User chooses one and clicks on it.
3. User clicks on the grid where he wants to put the intersection.
4. System draws the intersection on the grid.

**Post-condition:** an intersection is placed on a grid.

**Extensions:**

-3A: *The place on the map is already taken by another object:*

- the system triggers an error sound.
- Nothing changes on the map.
- Back to step 1.

---

## Remove intersection without zebra crossings

**ID:** 11

**Name:** Remove intersection

**Goal:** Remove an intersection from the current map.

**Actor:** User

**Pre-condition:** The software should be running

**Trigger:** User wants to remove an intersection from the current map

**Main success scenario:**

1. User clicks on the intersections that he wants to remove
2. User chooses from the menu the remove option
3. System removes the selected intersection

**Post-condition:** Intersection is removed from the screen.

**Extensions:** *None*

---

## **Add intersections with zebra crossing**

**ID:** 11

**Name:** Add road with a traffic light & zebra crossing

**Actor:** User

**Goal:** To add a road with a traffic light to the current map.

**Pre-condition:** The Software should be running

**Trigger:** User wants to add a road with a traffic light & zebra crossing to the map

**Main success scenario:**

1. System shows the available roads with traffic lights & zebra crossings
2. Actor chooses one and clicks on it
3. Actor click on the map where he wants to put the road with traffic lights & zebra crossings
4. System draws the road with traffic lights & zebra crossings on the map

**Post-condition:** Road with traffic lights & zebra crossings is drawn on screen.

**Extensions:**

-3A: *If the place on the map is already taken by another object:*

- the system triggers an error sound
  - Map stays unchanged..
  - Back to step 1.
- 

## Remove intersections with zebra crossing

**ID:** 12

**Name:** Remove road with traffic light & zebra crossing

**Actor:** User

**Goal:** To remove a road which has a traffic light & zebra crossing

**Pre-condition:** The Software should be running

**Trigger:** User wants to remove a road with a traffic light & zebra crossing

**Main success scenario:**

1. Actor clicks on the road with traffic lights & zebra crossings that he wants to remove
2. Actor chooses from the menu the remove option
3. System removes the selected road with traffic lights & zebra crossings

**Post-condition:** Road with traffic lights & zebra crossings is removed from the screen.

**Extensions:** *None*

---

## Save a file

**ID :** 13



**Name:** Save a file

**Goal:** Save the contents of the currently open application

**Actor:** User

**Trigger:** User chooses option "Save"

**Main Success Scenario:**

1. The application saves the contents using the current name and location of the file.
2. The system displays the time and date of the last save in a message.

**Post-condition:** The content of the application is stored in a file.

**Extensions:**

-1A: *Current application has not been given a name and location*

1. System goes to "Save As" use case
- 

## Save as a file

**ID:** 14

**Name:** Save As file

**Goal:** Saving contents of currently open file

**Actor:** User

**Trigger:**

- User chooses option "Save As"
- User tried to perform "Save a file" use case but did not specify location nor name for the file.

**Main Success Scenario:**

1. Application asks for name and location of file to be saved
2. Actor provides name and location of file
3. Actor confirms by clicking the "Save" button
4. Application saves contents using given name and location of file

**Extensions:**

-2A: *Actor presses "Cancel" button:*

1. Use case ends

-3A: *There is already a traffic file with the same name and path:*

1. The system displays appropriate messages and offers choices to replace existing files.
  2. If yes, the use case continues.
  3. If not, the actor is returned to step 4.
- 

## **Specify number of cars generated**

**ID:** 15

**Name:** Specify number of cars generated

**Goal:** Specify the number of cars that will take part in the traffic flow

**Actor:** User

**Trigger:** User signals to change the default value of cars generated

**Main Success Scenario:**

1. User types in text box the desired number of cars to be shown

**Extensions:**

-1A: *The value entered is not a number type/ exceeds the maximum value allowed/ is not a positive value:*

- The system shows an appropriate message.
- Return to step 1 of MSS.

# ITERATION 2

---

## Add roundabouts

**ID:** 16

**Name:** Add road with a roundabout

**Actor:** User

**Goal:** To add a road with a roundabout to the current map.

**Pre-condition:** The Software should be running

**Trigger:** User wants to add a road with a roundabout to the map

**Main success scenario:**

5. System shows the available roads with traffic lights & zebra crossings
6. Actor chooses one and clicks on it
7. Actor click on the map where he wants to put the road with a roundabout
8. System draws the road with a roundabout on the map

**Post-condition:** Road with a roundabout is drawn on screen.

**Extensions:**

-3A: *If the place on the map is already taken by another object:*

- the system triggers an error sound
  - Map stays unchanged..
  - Back to step 1.
-

## Remove intersections with roundabout

**ID:** 17

**Name:** Remove road with roundabout

**Actor:** User

**Goal:** To remove a road which has a roundabout

**Pre-condition:** The Software should be running

**Trigger:** User wants to remove a road with a roundabout

**Main success scenario:**

1. Actor clicks on the road with roundabout that he wants to remove
2. Actor chooses from the menu the remove option
4. System removes the selected road with roundabout

**Post-condition:** Road with roundabout is removed from the screen.

**Extensions:** *None*

---

## Change simulation speed

**ID:** 18

**Name:** Change simulation speed

**Actor:** User

**Goal:** To increase or decrease the speed of the simulation

**Pre-condition:** The Software should be running

**Trigger:** User wants to increase or decrease the speed of the simulation

**Main success scenario:**

1. Actor clicks the button for changing the speed of the simulation

2. Actor chooses how much to increase or decrease the speed of the simulation
5. System changes the speed of the simulation

**Post-condition:** The speed of the simulation is changed.

**Extensions:** *None*

---

## Simulation timer

**ID:** 19

**Name:** Add timer for the simulation

**Goal:** Simulation stops when time runs out

**Actor:** User

**Description:** A user can choose a number of seconds for the duration of the simulation

**Pre-condition:** The software is running

**Trigger:** User wants to specify the duration of the simulation.

### **Main success scenario:**

1. System shows the main screen of the application.
2. User enters number of seconds.
3. User chooses other options
4. User starts simulation.
5. System runs simulation.
6. System stops simulation after the time has passed.

**Post-condition:**

- The main screen of the application is shown.

**Extensions:**

2A: The user doesn't specify the number of seconds

- system shows a message
  - Back to step 1.
- 

## **Simulation speed**

**ID:** 20

**Name:** Add slider for the speed of the simulation

**Goal:** Simulation runs faster or slower depending on the user's choice

**Actor:** User

**Description:** A user can choose a speed for the simulation

**Pre-condition:** The software is running, a grid has not yet been chosen

**Trigger:** User wants to specify the speed of the simulation.

**Main success scenario:**

1. System shows the main screen of the application.
2. User adjusts the slider.
3. User chooses other options

4. User starts simulation.
5. System runs simulation with correct speed.

**Post-condition:**

- The simulation's speed has increased or decreased

**Extensions:**

- 2A: The user chooses a grid before adjusting slider
- system shows a message
  - Back to step 1.
- 

## **Add Surprise Elements**

**ID:** 21

**Name:** Enable Surprise elements.

**Goal:** Allowing random car crashes to take place within the simulation.

**Actor:** User

**Description:** The user can choose to enable random car crashes within the simulation (to have more of realism).

**Pre-condition:**

1. The simulation is not running.
2. There is at least one road placed on the grid, and at least 1 car to be generated.

**Trigger:** User Clicks on "Surprise Elements" checkbox.

**Main success scenario:**

1. 1.User checks the "Surprise Elements" checkbox.

2. User runs the simulation.

**Post-condition:**

1. Cars will have a random chance to crash into each other during the simulation.

**Extensions:**

1A. "Surprise Elements" checkbox is already checked:

1. Continue with Step 2.

2A. User unchecks the "Surprise Elements" checkbox

1. Use case ends.
2. Back to step 1.

3B. User stops/pauses the simulation:

1. Use case ends.
  2. Back to step 1.
- 

## **Enable Special (Emergency) Vehicles**

**ID:** 22

**Name:** Add Special Emergency Vehicles

**Goal:** Generate Emergency vehicle to pick up crashed cars

**Actor:** User

**Description:** User enables emergency vehicles that pick up crashed cars, these vehicles follow the shortest-path-possible algorithm.

**Pre-condition:**

1. Simulation is not running.
2. There is at least one road placed on the map, and at least 2 car to be generated.



**Trigger:** User checks the “Emergency vehicles” checkbox.

**Main success scenario:**

1. User checks the “Emergency Vehicles” checkbox.
2. System asks the user to choose the start road of the vehicles.
3. User chooses a road piece.
4. System asks the user to choose the destination road of the vehicles.
5. User chooses another road piece.
6. User runs the simulation.

**Post-condition:**

1. Whenever two cars crash into each other during the simulation, an emergency vehicle will spawn from the starting road and will follow the shortest path to the destination road piece.

**Extensions:**

1A. “Emergency Vehicles” is already checked:

1. Proceed to step 2.

3A, 5A. User chooses “Cancel” instead

1. Use case ends.
2. Back to step 1.

6A. User resets the simulation.

1. Use case ends.
2. Back to step 1.

---

## Disable Special (Emergency) Vehicles

**ID:** 23

**Name:** Remove Special Emergency Vehicles

**Goal:** Disable emergency vehicles from spawning during the next simulation.

**Actor:** User

**Description:** User disables previously enabled emergency vehicles from spawning during the simulation runtime.

**Pre-condition:**

1. Simulation is not running.
2. "Emergency Vehicles" checkbox is enabled.

**Trigger:** User unchecks the "Emergency vehicles" checkbox.

**Main success scenario:**

1. User unchecks the "Emergency Vehicles" checkbox.
2. User runs the simulation.

**Post-condition:**

1. Whenever two cars crash into each other during the simulation, no emergency vehicles will spawn.

**Extensions:**

1A. "Emergency Vehicles" is already unchecked:

1. Proceed to step 2.

2A. User resets the simulation.

1. Use case ends.
2. Back to step 1.

---

## Current Flow

**Name:** Calculate current flow of cars

**Goal:** System shows the current number of cars on the road network

**Actor:** User

**Description:** User can see the number of cars in the simulation

**Pre-condition:** The software is running

**Trigger:** User wants to see the number of cars in the simulation.

**Main success scenario:**

1. System shows the main screen of the application.
2. User starts simulation
  3. User clicks the “Current flow” button.
  4. System calculates the number
  5. System shows the number

**Post-condition:**

- The simulation has shown the current number of vehicles in the simulation

**Extensions:**

2A User resets the simulation

1. Use case ends
2. Back to step 1

## Adjust Number of Pedestrians

**Id:** 26

**Name:** Adjusting Number of Pedestrians

**Goal:** Increasing and Decreasing numbers of pedestrians

**Actor:** User

**Description:** User can add pedestrians to map

**Pre-condition:** The software is running

**Main success scenario:**

1. User adjust number of pedestrian using trackbar
2. System shows the number of pedestrians user can add

## Enable Road Sensors

**Id:** 27

**Name:** Enabling Road Sensors

**Goal:** Pedestrian can walk in zebra cross

**Actor:** User

**Description:** User can make pedestrian walk in zebra cross by activating enable road sensors

**Pre-condition:** The software is running

**Main success scenario:**

1. User clicks the checkbox “Enable Road Sensors”
2. System informs user

**Extensions:**

1A. There are no pedestrians:

1. System informs user
2. Proceed to use case “Add Pedestrians”

## Disable Road Sensors

**Id:** 28

**Name:** Disabling Road Sensors

**Goal:** Pedestrian walks on the sidewalk

**Actor:** User

**Description:** User can make pedestrian walks only in sidewalk

**Pre-condition:** The software is running

**Main success scenario:**

1. User unchecks the checkbox "Enable Road Sensors"
2. System informs user