

**ProCp**

Design document

Version: 5.0

**Group members**:

Redzhep Molaahmed – 3058212

Son Duong – 3471381

Imran Touqi – 3475379

Dean Selivestrov – 3021270

Huy Bui – 2790106

Abraham Ackom-Mensah – 2983214

Contents

Revision History 1

Introduction 2

Entities needed for coding 2

UML 3

Sequence diagram 4

Functional and non-functional requirements 4

Interface design 4

# Revision History

|  |  |  |
| --- | --- | --- |
| Version | Date | Description |
| 1.0 | 21-10-2019 | First concept. |
| 2.0 | 25-10-2019 | Update sequence diagram & UML iteration 1. |
| 3.0  4.0 | 30-11-2019  08-12-2019 | Update document for iteration 2.  Updated document for iteration 2 and iteration 3 preparation. |
| 5.0  6.0 | 05-01-2020  17-01-2020 | Updated for iteration 3 UML, design, entities, sequence diagrams.  Updated Sequence diagram, Design |

Introduction

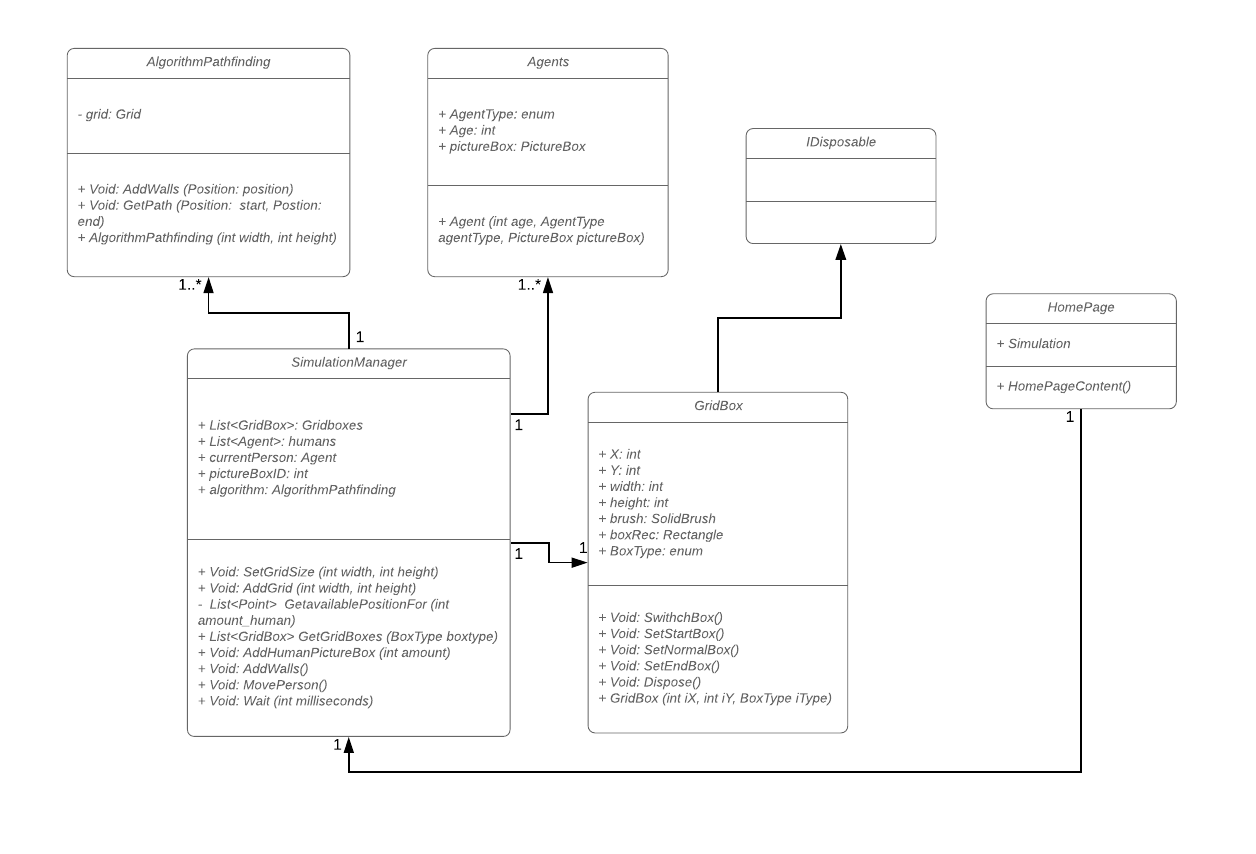
We are a team contains 6 ICT students who are building a 2D- simulation of people autonomously studying and working on a floor of a building where they can travel between different locations such as classroom, office or bathroom. The movement of those people are given randomly, once the application starts simulating the scenario of a fire, people should all stop what they are doing and head toward the exist.

Entities need for coding

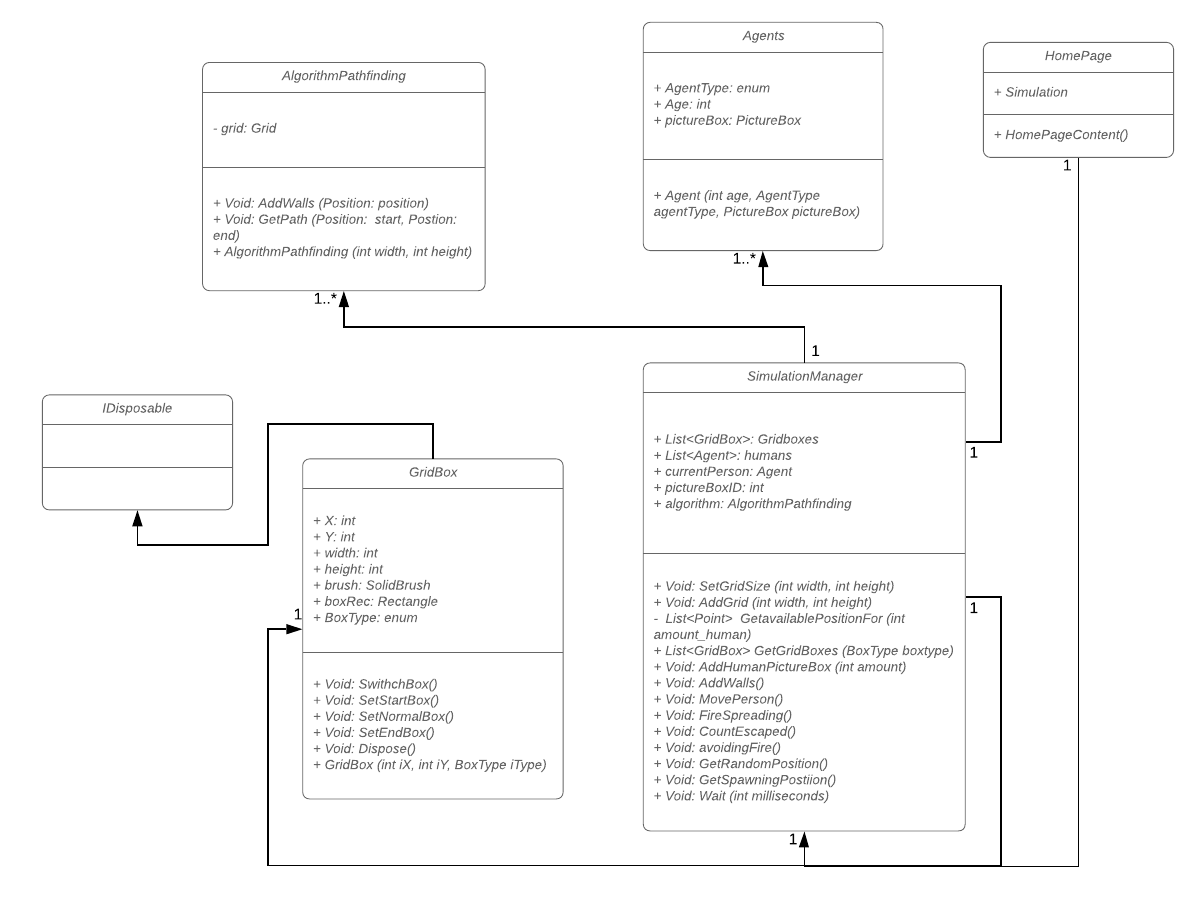
* Agents Class
* This class contains personal information of all people who are working or studying on the floor (Name, status, type,..etc.).
* SimulationManager Class
* This class is responsible for all methods that running the simulation.
* Grid Class
* This class is in charge of forming the grid and store all coordinates of entities on that grid.
* All Database classes
* This class allows the user to save all data of the simulation and responsible for showing this information.
* AlgorithmPathfinding
* This class contains the algorithm that running in the back-end calculates the shortest path a person could use to escape the floor plan via stored coordinates and A\* algorithm
* GroupPeople
* This class in charge of creating a group of selected number of people on the floor plan.
* Fire Class
* This class creates Fire on the floor plan as well as stores fire location, intensity,...
* MutipleSimulation class
* This class in charge of generating a vast amount of simulation for statistics and analysis purpose by covering all aspects of the simulation such as: injured people, escape chance throughout each exit, casualties,...
* MultipleResult and MultipleStatistic classes
* These classes stored data of multiple simulations and in charge of showing it as overall statistics.

UML

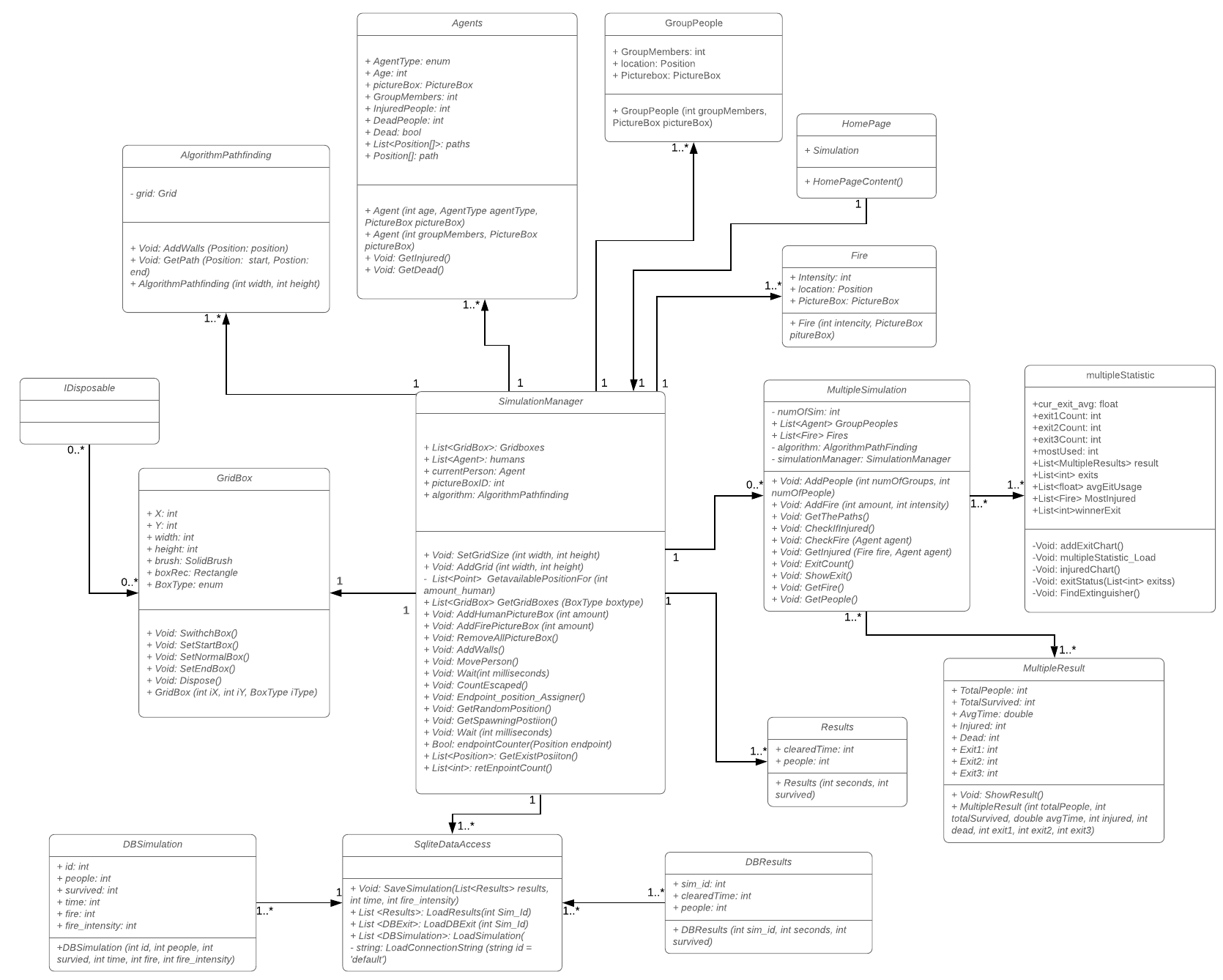
* UML diagram iteration 1



* UML diagram iteration 2

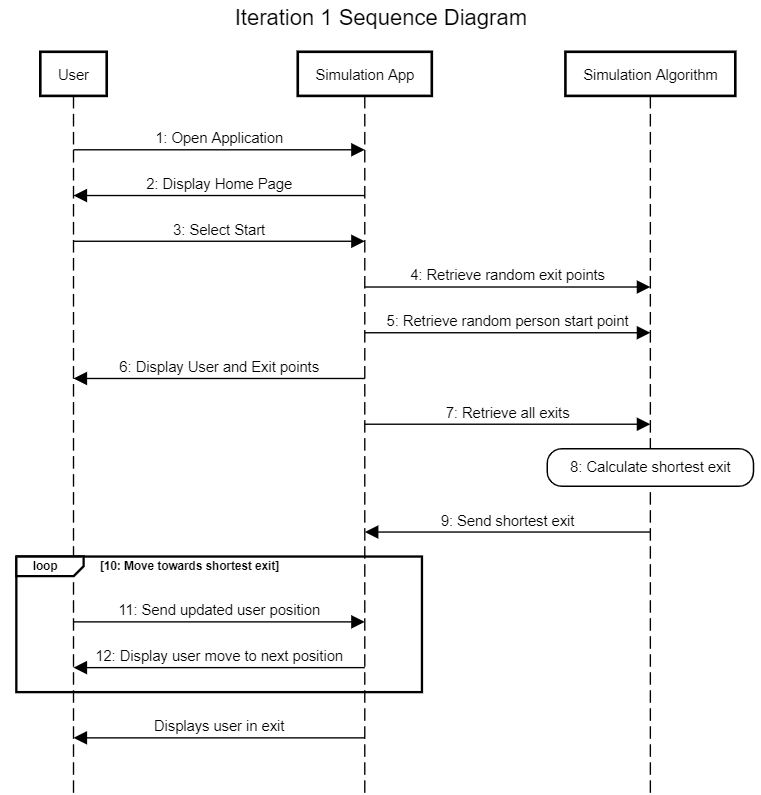


* UML diagram iteration 3



Sequence diagram

* Sequence diagram iteration 1

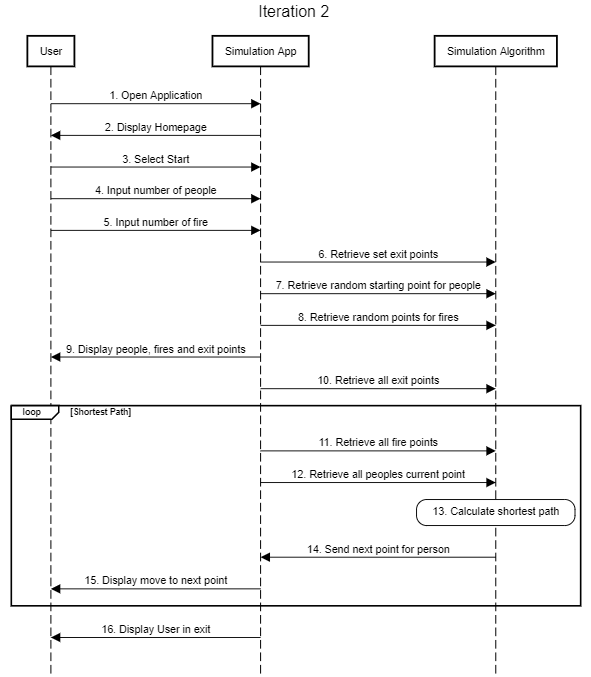


## 

## Breakdown:

1. Application is opened by the user
2. The app displays the simulation interface
3. The user selects start to run the application
4. User inputs the number of agents for the simulation
5. User clicks run button
6. Application Displays User at start point and exits.
7. Application algorithm takes all exit points
8. Application algorithm calculates the shortest path
9. User clicks on the Agent
10. Application displays movement of the Agent to the exit
11. Application Displays user in the exit

* Sequence diagram iteration 2



## 

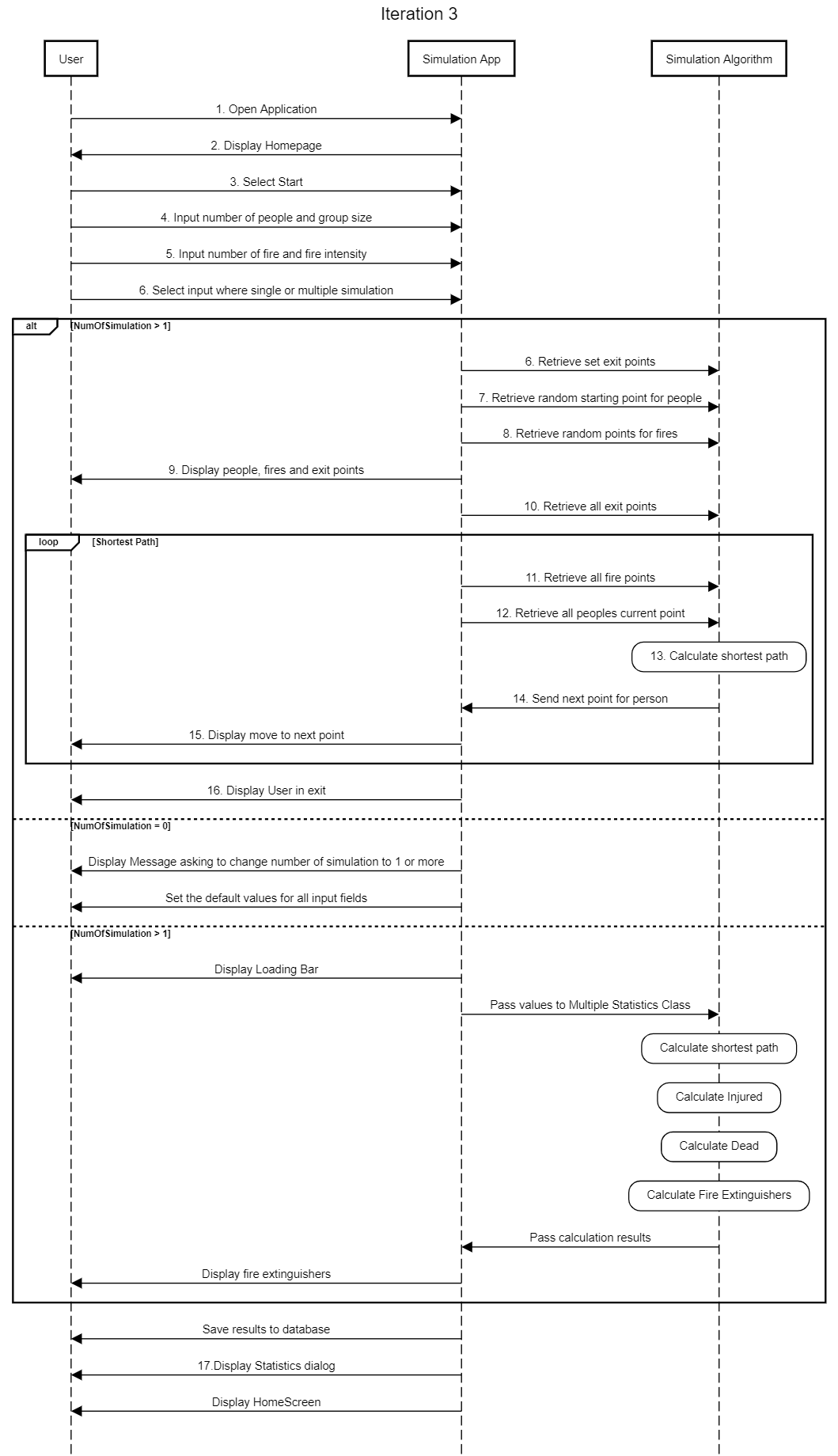
## 

## 

## Breakdown:

1. Application is opened by the user
2. The app displays the simulation interface
3. The user selects start to run the application
4. User inputs the number of human agents for the simulation
5. User input the number of fire for the simulation
6. Application displays agents and fire at random locations
7. User clicks run button
8. Application algorithm calculates the shortest path for each human agent
9. The application displays simultaneous movements of the Agents to the shortest exit while avoiding
10. Application Displays user in the exit
11. Application stops Displaying user.

* Sequence diagram iteration 3



## Breakdown:

1. Application is opened by the user.
2. The app displays the main interface
3. The user selects start to run the application
4. User inputs the number of human agents and number of agent group for the simulation
5. User input the number of fire and its intensity for the simulation.
6. User decided which type of simulation is going to be ran (multiple or single).
7. The application retrieve all exit points.
8. The application generate random starting points for people.
9. The application retrieve available point for fire.
10. The application displays all the information in the interface for user.
11. The application retrieve all fire and people coordinates.
12. The algorithm calculates the shortest path for each human agent.
13. The algorithm sends the path to each agent and they move based on that data.
14. The application displaying that movement in the interface.
15. The application displaying people escaped the floor plan.
16. If the number of Simulation set equal 0, the application shows a message that remind the user.
17. The application set all fields to default values.
18. If the number of Simulation set to more than 1, the application displaying the loading bar.
19. The application passing values to multiple Statistic class.
20. The algorithm calculates the shortest path for each human agent.
21. The algorithm calculates the number of people got injured.
22. The algorithm calculates the number of casualties.
23. The algorithm calculates the location of fire extinguishers.
24. The algorithm passing the results to the application.
25. The application displays suggested location of fire extinguishers.
26. The application saves results to the database.
27. The application displays statistics dialog.
28. The application displays home screen.

Functional and non-functional requirements

## Deliverables

* An application for a fire escape simulation which will emulate a real-life fire in a building with people and give the best possible escape routes and a correct number of fire extinguishers for a successful scenario.
* (URS) User requirements specification for the application.
* Success report of previous simulation tests.
* Design document for the application.

## Non-Deliverables

* User manuals
* Expected maintenance after project goal is reached
* Once the presentation has been given, and workshops are completed our company will have then finished the project and therefore will no longer help in the continuation or development of the app.
* Financial support of any kind
* Any money or financial matter needed for the development of the project will be provided by “” and Co, therefore under no circumstance will our company spend money in any way for the growth of the project, this includes:
  + - Payment for the employees.
    - Money for the technology needed.
    - Money to fix any risk or problem which occurs.
* Solutions for any matter not related to the project such as personal matters.

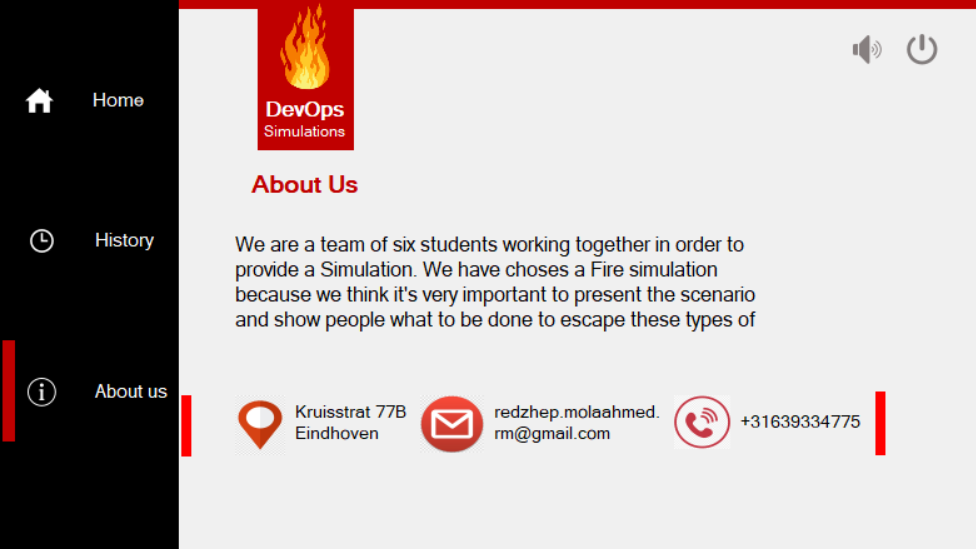
Interface design

In our application, we have 3 forms. The first one is the Home page which users could easily start the simulation.

Home Page

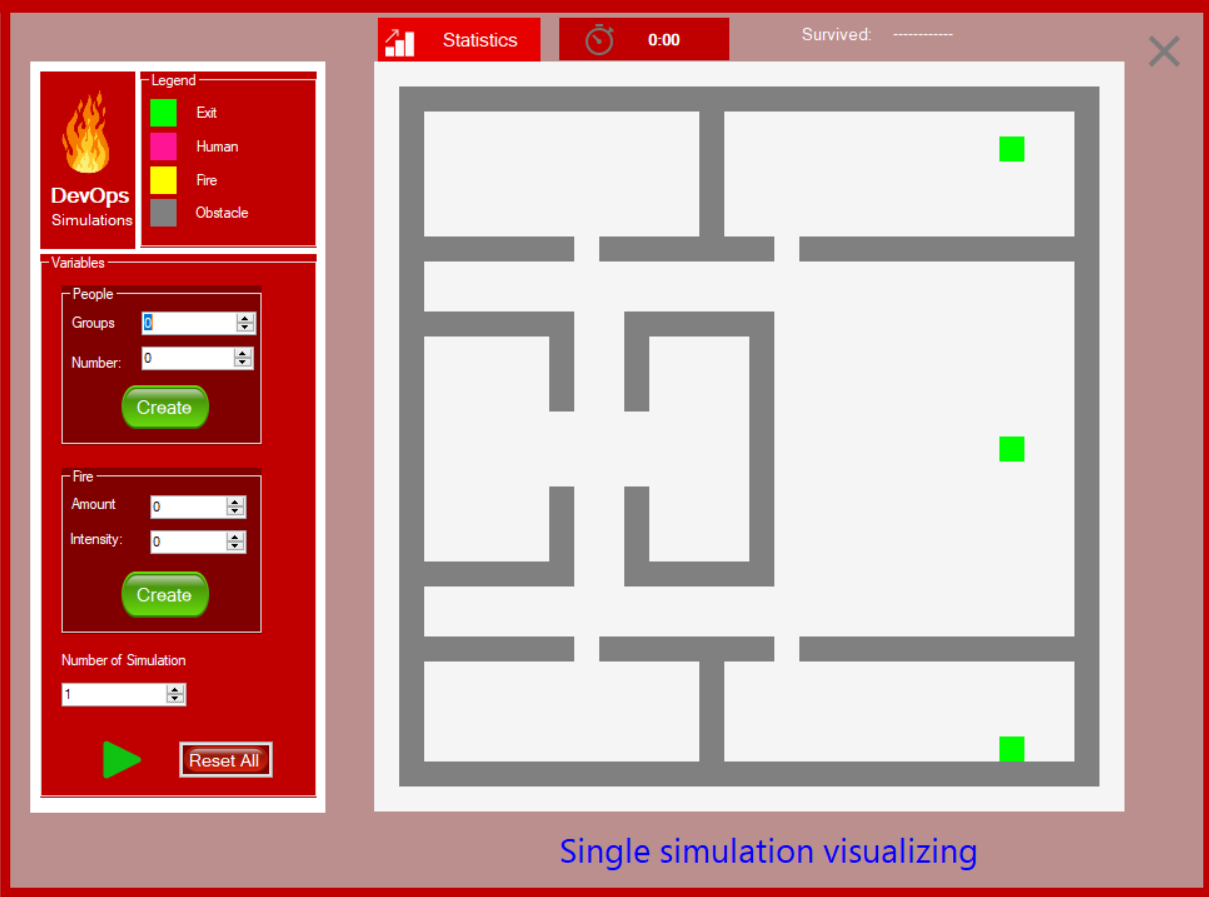


About us



* About us tab where users could find relevant information.

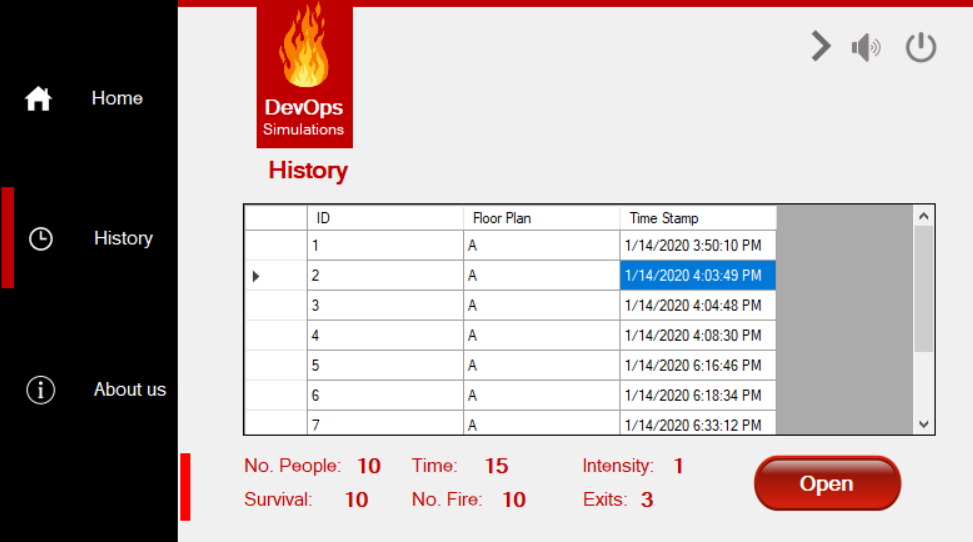
Simulation



* Simulation form where users could interact with the application and could visually observe the simulation in action.

History tab

* Showing all data about previous simulations.

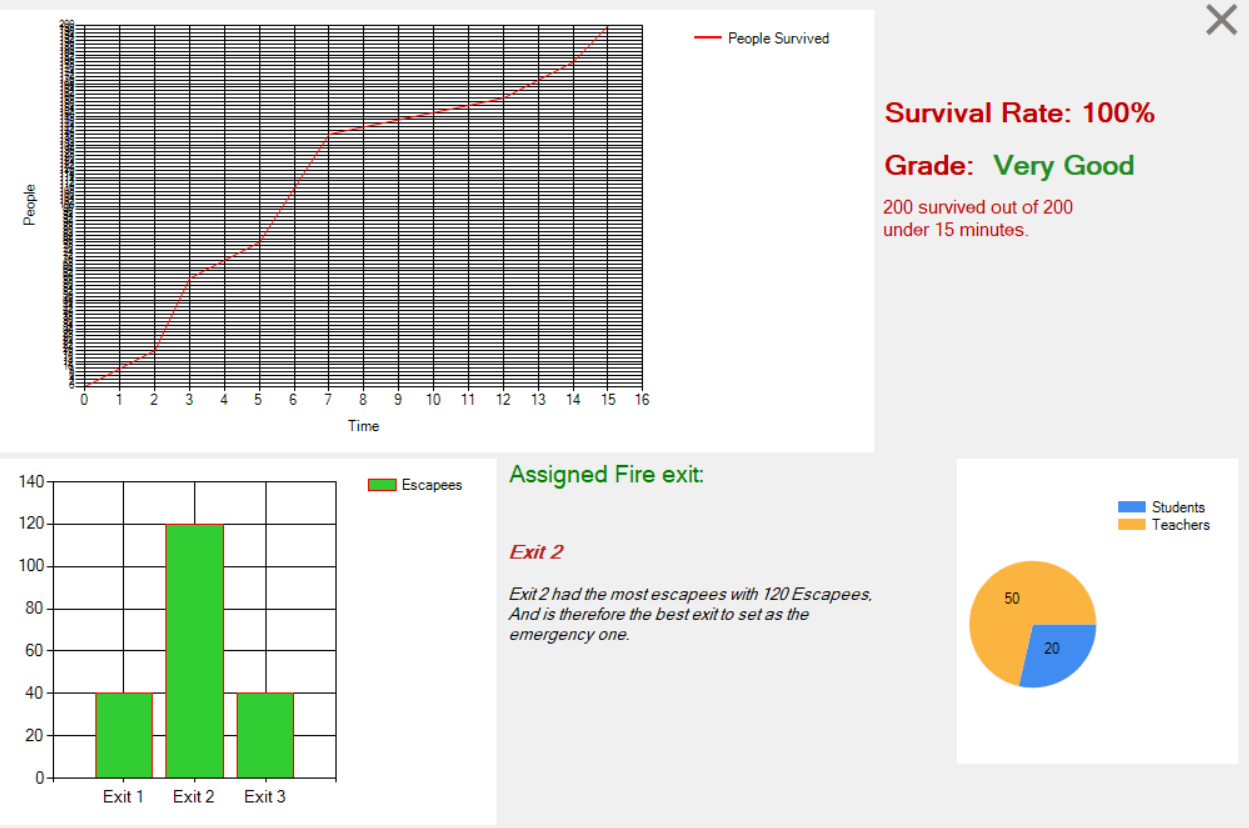


Statistic popup tab



* After the application finished single simulation statistic tab comes up.

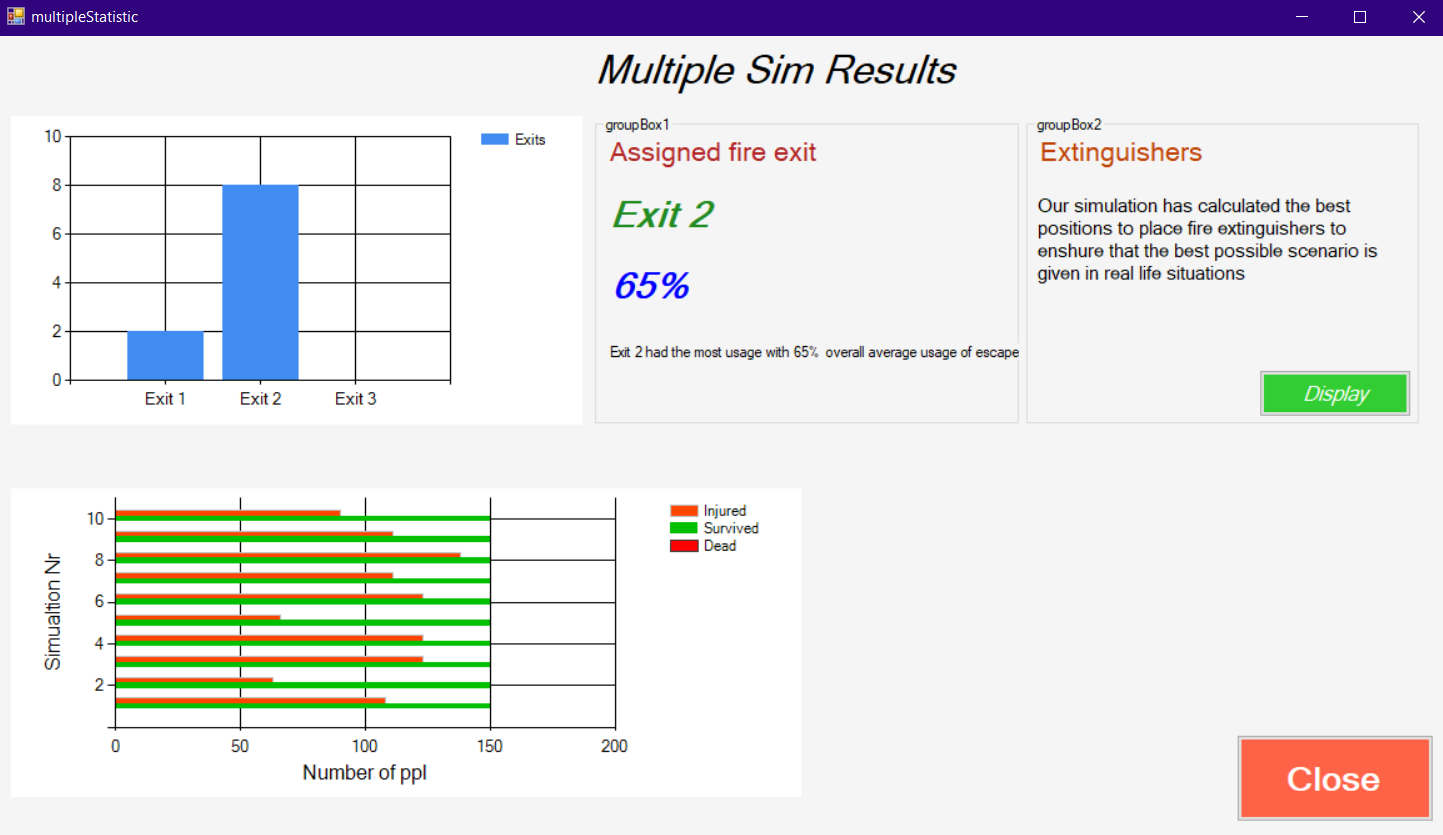
Single simulation statistics show



* This tab showing number of people survived through each exit and number of people escaped the floor plan throughout the time by graphs.

Multiple simulations statistics

* Statistics of multiple simulations shown in graphs.



Fire distinguisher location suggestion

* After analyzing all simulations the program process the data to suggest on which location the fire distinguisher should be placed.

