Software Requirement Specification

Louw, Matthew Jason
u16217498

Bresler, Mathilda Anna
u16313382

Braak, Pieter Albert
u16009917

Reva, Kateryna u17035989

Li, Xiao Jian
u16099860

April 2019

1 Introduction

1.1 Purpose

This document is written with the COS 301 lectures, the CS department, and epi use as clients in mind. This SRS will provide all parties concerned with details concerning the Real time fire escape routes project. The SRS contains information relevant to the current scope of the project, and is thus not a complete representation of the final project since there are expected to be changes to the system as the project takes shape. In this document we hope to convey our current understanding of the project and its implementation to the previously mentioned clients. This will guide the developers and the clients to obtain a mutual understanding of what is expect from the system.

1.2 Scope

The Real-time Fire Escape Route (Real-time FER) System is a new approach to solve an age old problem. The interface is in the form of mobile application as well as desktop application. The main goal of the Real-time FER is to indicate to the agent using the application what the most efficient route would be to take in the case of an emergency. The system will perform this functions by anonymously tracking the building populations to be aware of the building's current state. It will be done with the use of a heatmap, which is generated using the Bluetooth sensors installed throughout the monitored location. This will show the quantity and distribution of the population throughout the building.

The escape routes are pre-planned by the building designers, since there are special considerations that need to be made, including which walls have fire proofing, and which areas should be avoided for safety reasons. The heatmap is then to be consumed by an AI algorithm to assign a fire escape route to each agent in the building. The building's population distribution and available routes must be taken into consideration when assigning a route to each individual.

With the building heatmap and time sensitive fire escape route in place, each agent in the building will be sent a push notification to his/her phone indicating what escape path to follow in case of emergency.

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
FER	Fire Escape Route
WebUI	Web User Interface
RFC	Request for Comments
AI	Artificial intelligence
CS	Computer science
COS 301	Abbreviation for the module named "Software Engineering"

1.4 References

Department of CSE. (2015). Software Requirements Specification ATM. In: Department of CSE Software Engineering and Project Management Lab Manual. Indore: Department of CSE. 1-17.

1.5 Overview

The sensors will be placed in various locations, the server will then iterate through the sensors fetching data from each sensor, the server will map the data from various sensors into relevant locations. The server will then process the Map data generating various routes depending on agent locations and send it to the applications of the various agents. The application will sync the latest version from the server upon opening. The WebUI will be used to view a live view of the map as well as allow new agents to register into the system.

- **Section 1:** Describes the main purpose of the system, it's scope, and relevant information to interpret the rest of the document.
- **Section 2:** Specifies the different users that will make use of the system and it's sub-systems, and their expected adaptability to the new system.
- **Section 3:** Specifies the components and the interactions among them that are required for the system to be functional.
- **Section 4:** Outlines the Quality requirements that the system needs to fulfill, and a means of testing whether the system adheres to these requirements.
- **Section 5:** Shows the requirements of the system, and whether they are met by the Use-cases.

2 User characteristics

The two user group interacting with the application are:

Administrative users: these users will have full access to the system. They can add locations, edit escape routes, and manage the user account on the system.

Agent/regular users: these users will interact with the application interface primarily in the event of emergencies. These users are temporary or permanent occupants of the building in which the system is installed.

3 Functional requirements

The Mobile Application will:

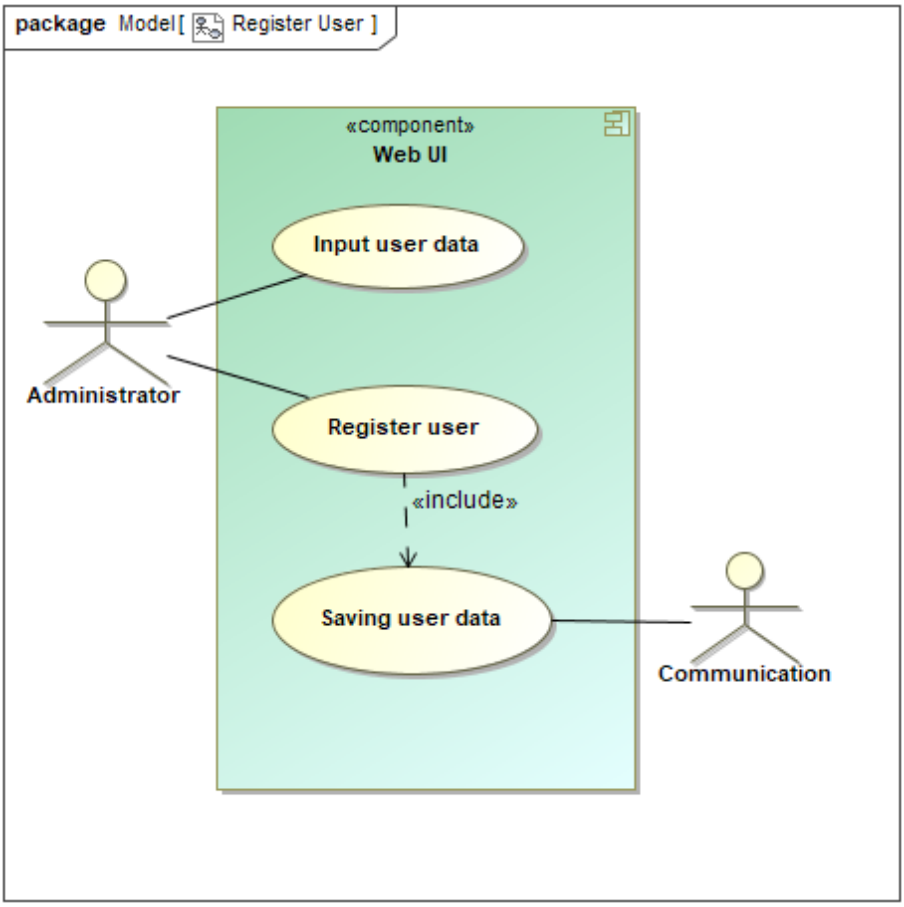
- Provide push notications to the 'agents' in the case of an emergency.
- The notication will provide a visual indication of the route to be followed.
- The application will broadcast the 'agents' position on the network for processing by the system.
- Provide additional notications upon changes in routes.

The system will:

- Track agents using their devices in real-time.
- Generate Heatmaps based on population density in areas.
- Heatmaps will be generated using Wi/bluetooth sensors placed throughout the building.
- Assigned Routes depending on various variables.
  - Routes will be assigned depending on the amount of people already assigned to the route.
  - Routes should if possible avoid potential bottlenecks if the population density for an area is high.e.
  - The total distance of the route.
  - Routes will be dierent for each agent depending on their location.
  - Routes will be updated if there are signicant changes in data.

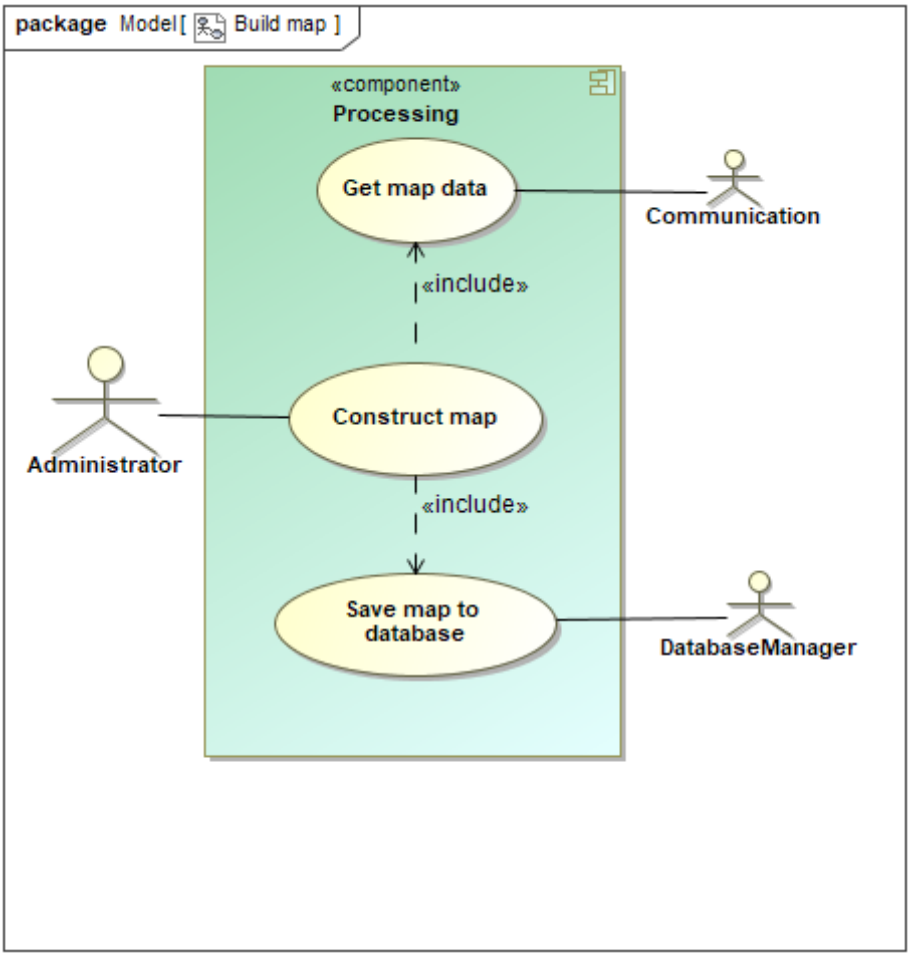
3.1 Use Cases

3.1.1 Register Users Use Case - UC1



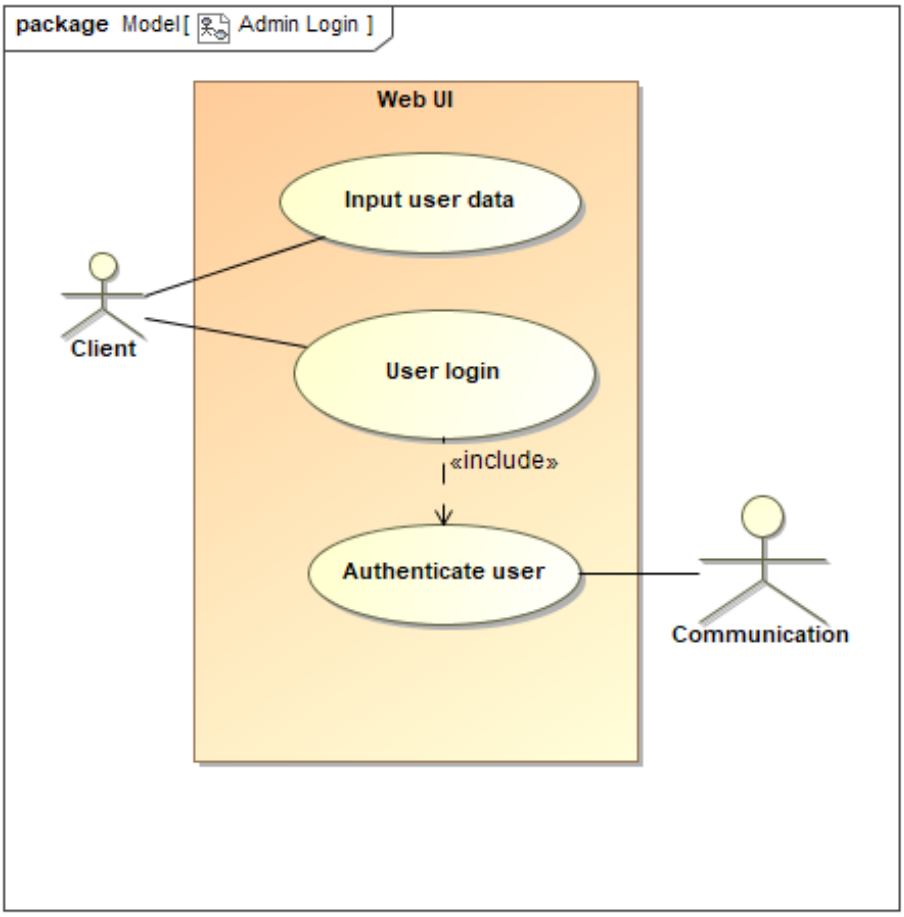
The actors in this use case are the Clients and the Communication subsystem. The new user to be registered's data is input through the The Web UI which is implemented with HTML, CSS, Javascript and Bootstrap which is run on a traditional web browser. The data is then used to register the the communication subsystem, which mainly consists of saving the user data to the database subsystem.

3.1.2 Build Map Use Case - UC2



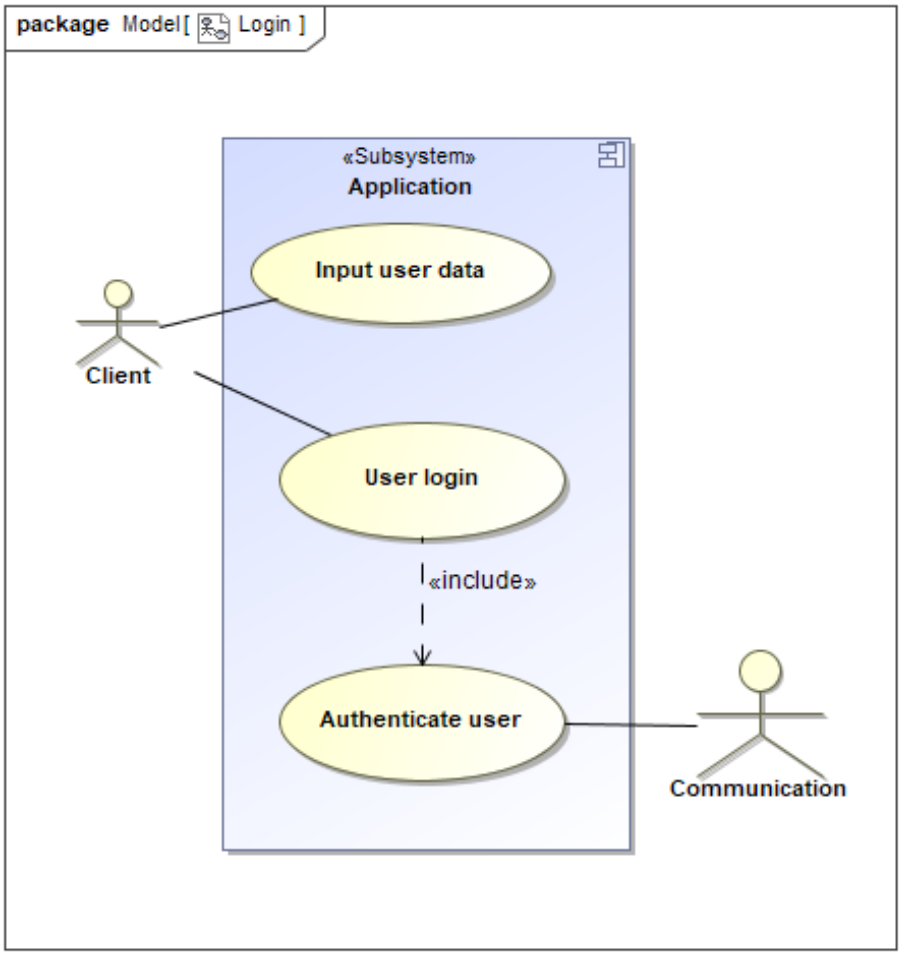
The actors involved in this use case is the adminstrative users (to provide the building data), the communication subsystem, and the database subsystem (through the DatabaseManager). The administrator initiates the construction of the map, this happens through input facilitated by the communication subsystem. The generated map data is then saved to the system through the databaseManager.

3.1.3 Admin Login Use Case - UC3



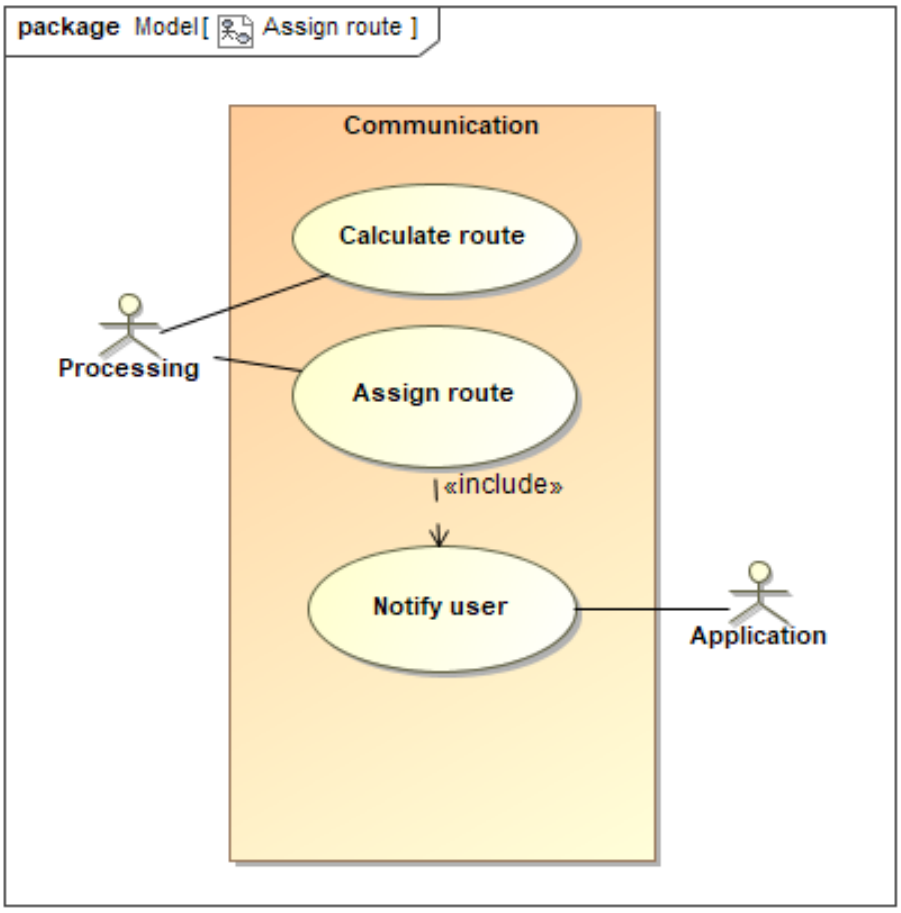
The actors in this use case are the Administrator users and the Communication subsystem. The new user to be registered's data is input through the The Web UI which is implemented with HTML, CSS, Javascript and Bootstrap which is run on a traditional web browser. The data is then used to register the communication subsystem, which mainly consists of saving the user data to the database subsystem.

3.1.4 Login Use Case - UC4



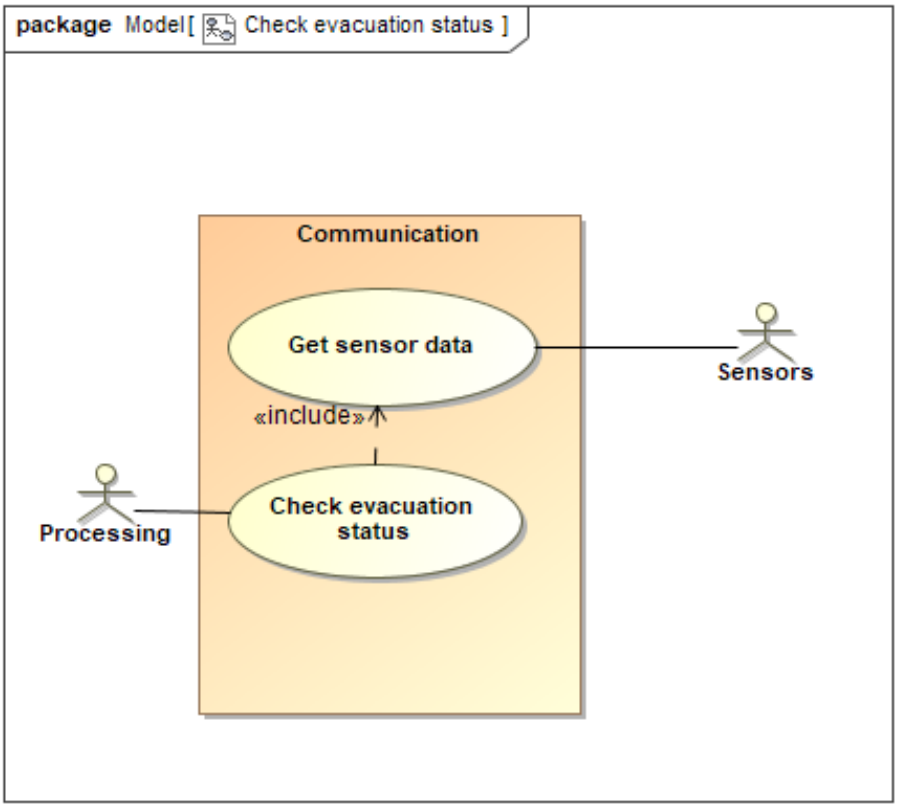
The actors in the system are The Client and the communication subsystem. The login is done through an application, which is still to be implemented, and which is currently mocked by the web UI. This is used to capture the client's data, which is then verified through the communication subsystem, which is a server implemented with Java, which makes use of the database subsystem to authenticate the user. Which then completes this use case.

3.1.5 Assign route - UC5

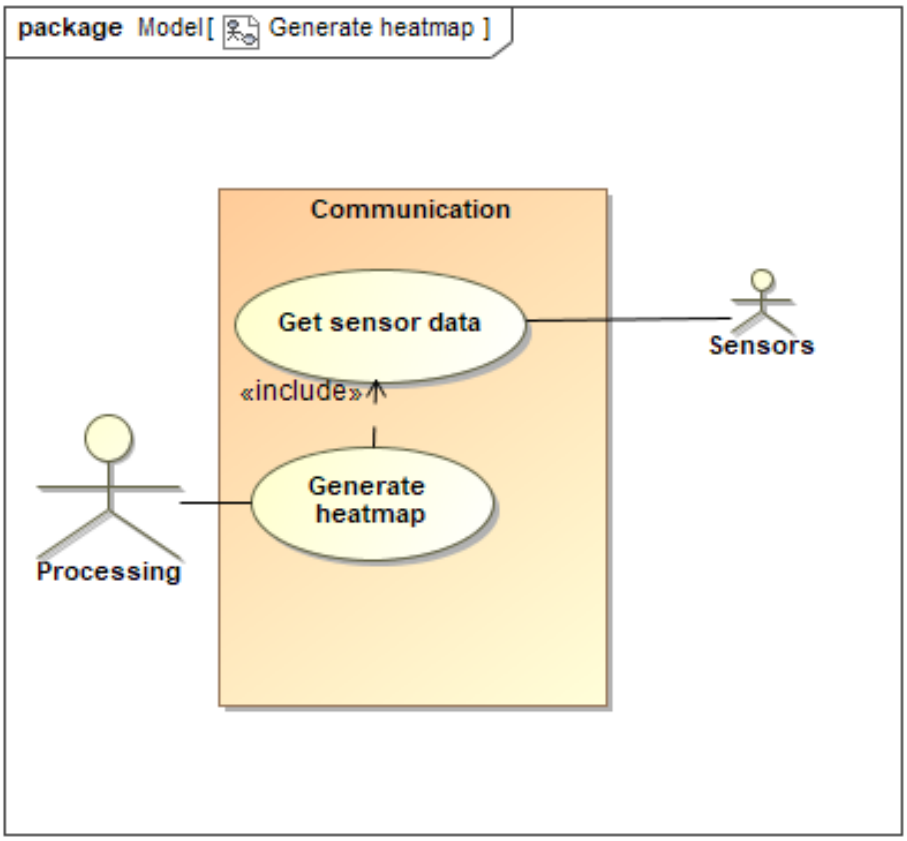


The actors involved is the Processing subsystem as well as the Application subsystem. To assign a route the Communication subsystem uses data available to it through other subsystems to calculate the route most efficient for the 'Agents' to be assigned to. The communication subsystem then communicates with the application subsystem to notify the 'Agents', telling them which route they should take to optimize the evacuation process.

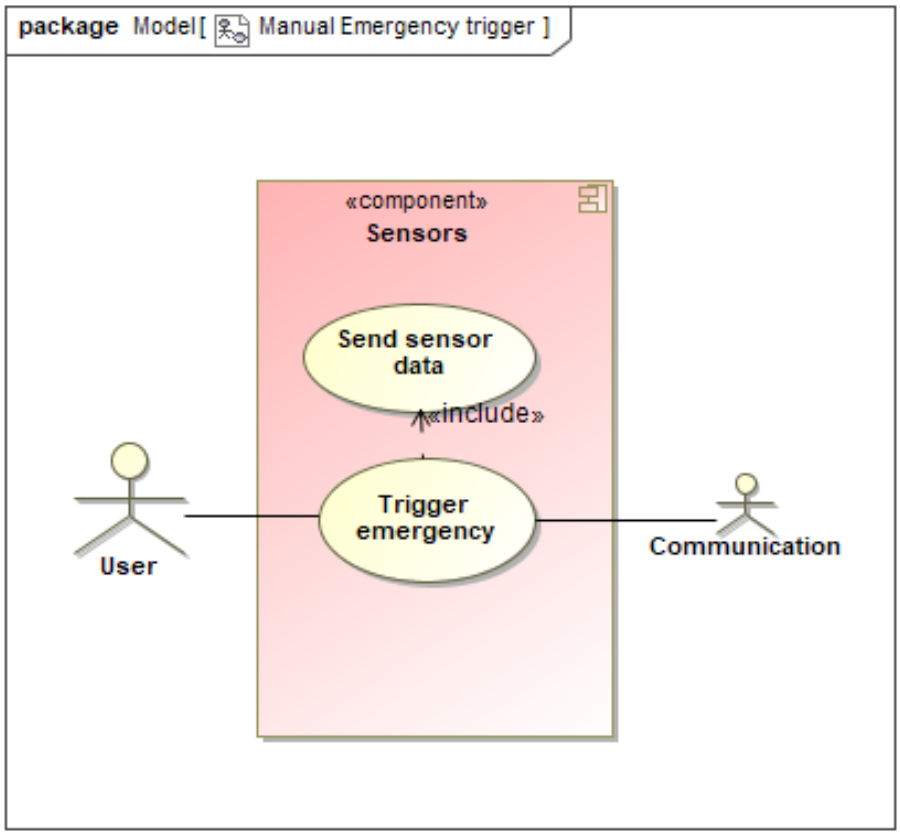
3.1.6 Check Evacuation status Use Case - UC5



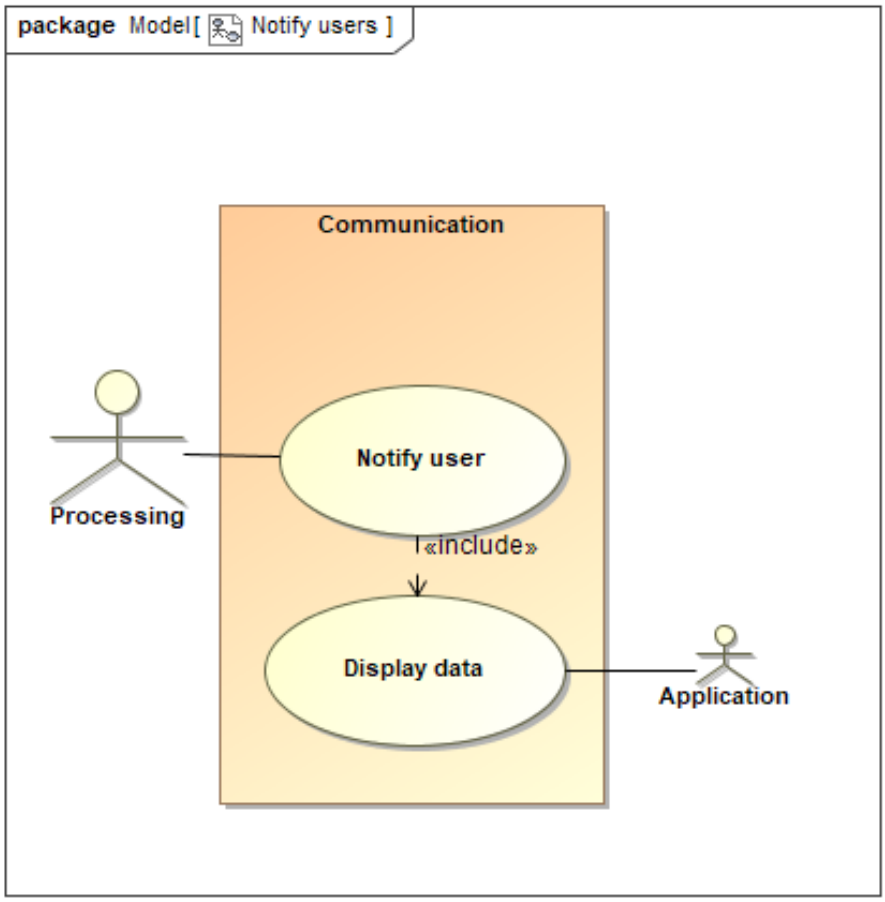
3.1.7 Generate Heatmap Use Case - UC6



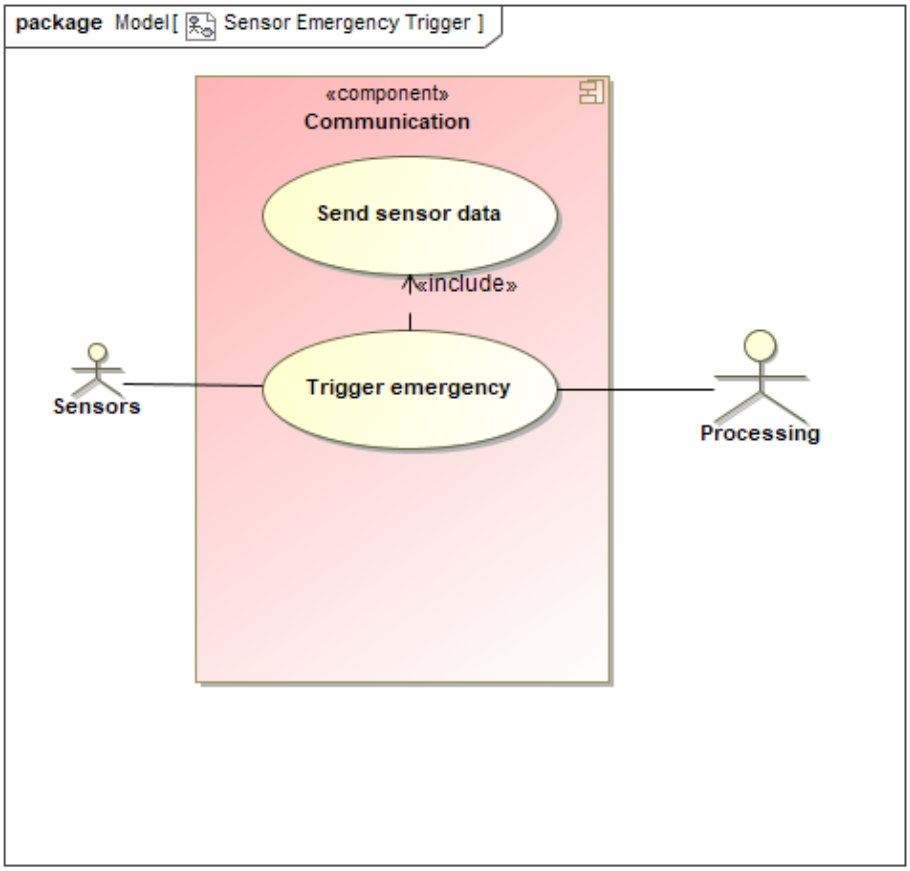
3.1.8 Manual Emergency Trigger Use Case - UC7



3.1.9 Notify Users Use Case - UC8

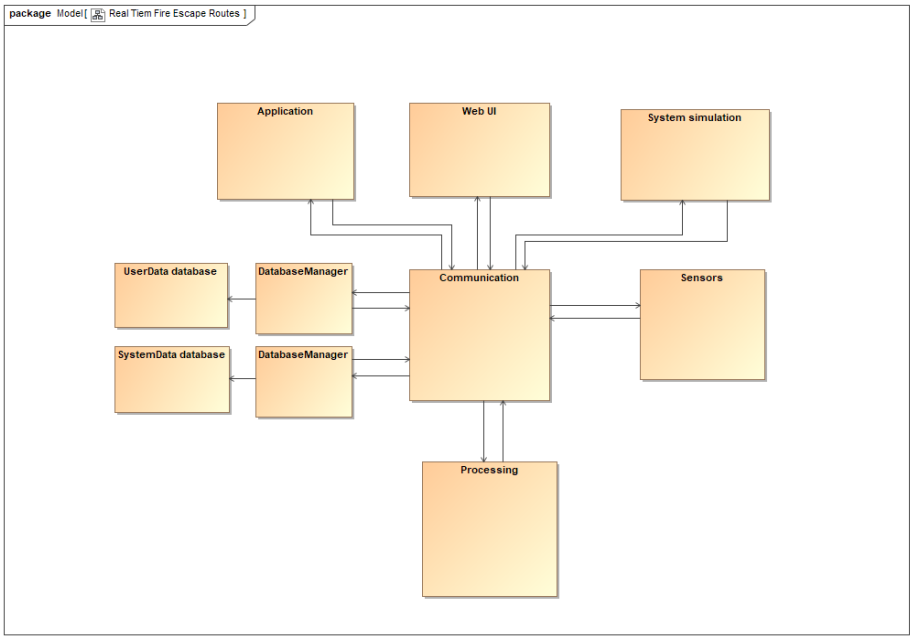


3.1.10 Sensor Emergency Trigger Use Case - UC10



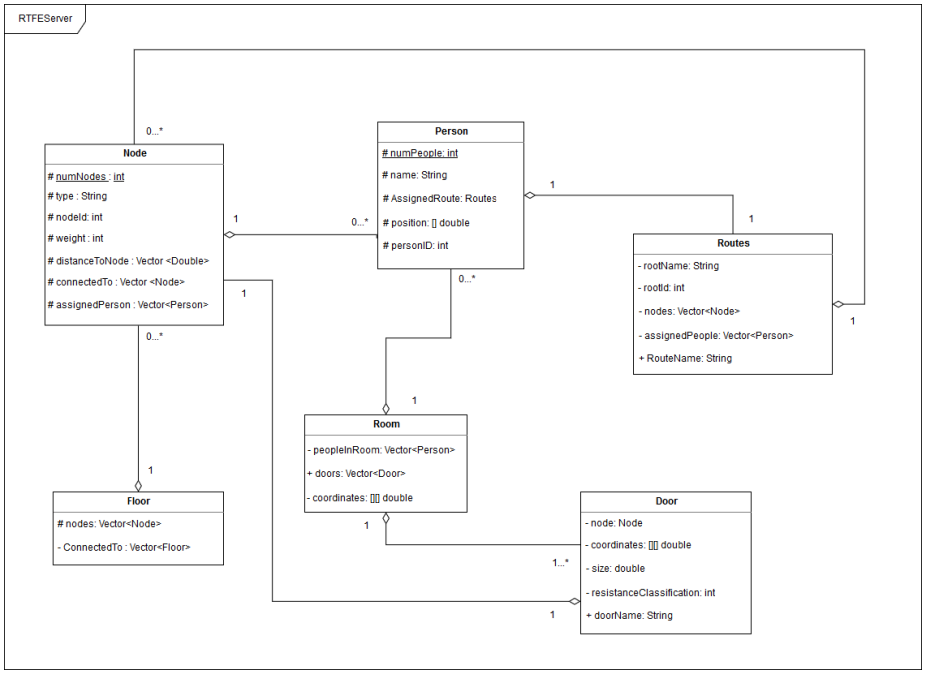
3.2 Subsystems

3.2.1 System overview and architectural design

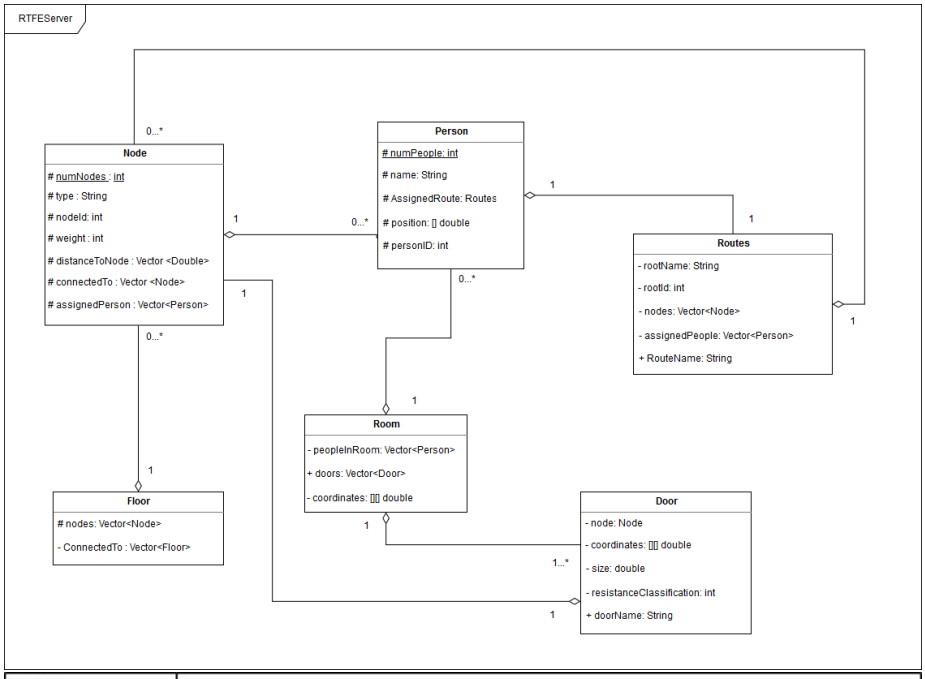


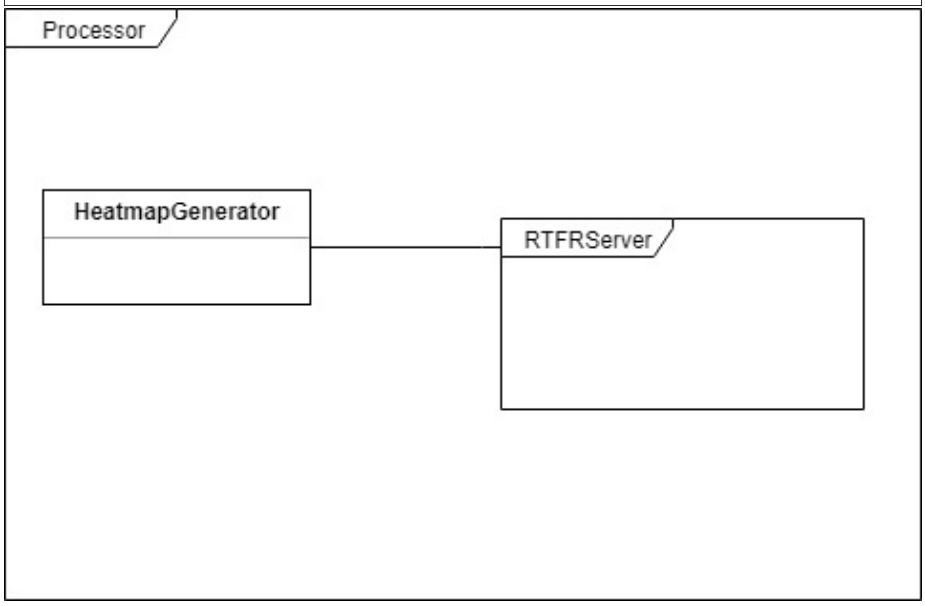
The main architectural pattern used by our System is the Event-driven architecture. This can be seen by the fact that the system uses a Controller subsystem (Called 'Communication' in our system). This subsystem is responsible for reacting to certain events depending on the current state (the heatmap layout and user to route assignment) of the system. There are also object persistent architectural styles present when it comes to the databases and their management.

3.2.2 Communication Subsystem - S1

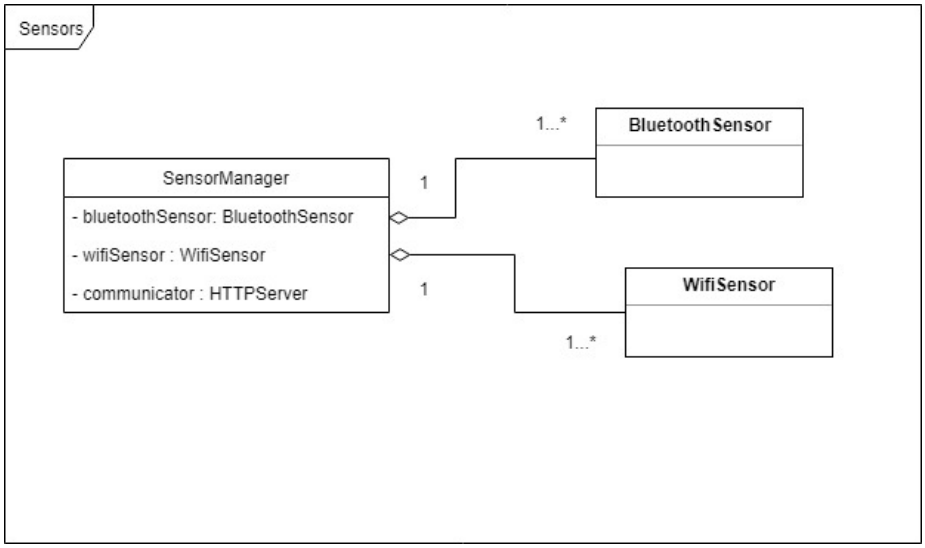


3.2.3 Processing Subsystem - S2

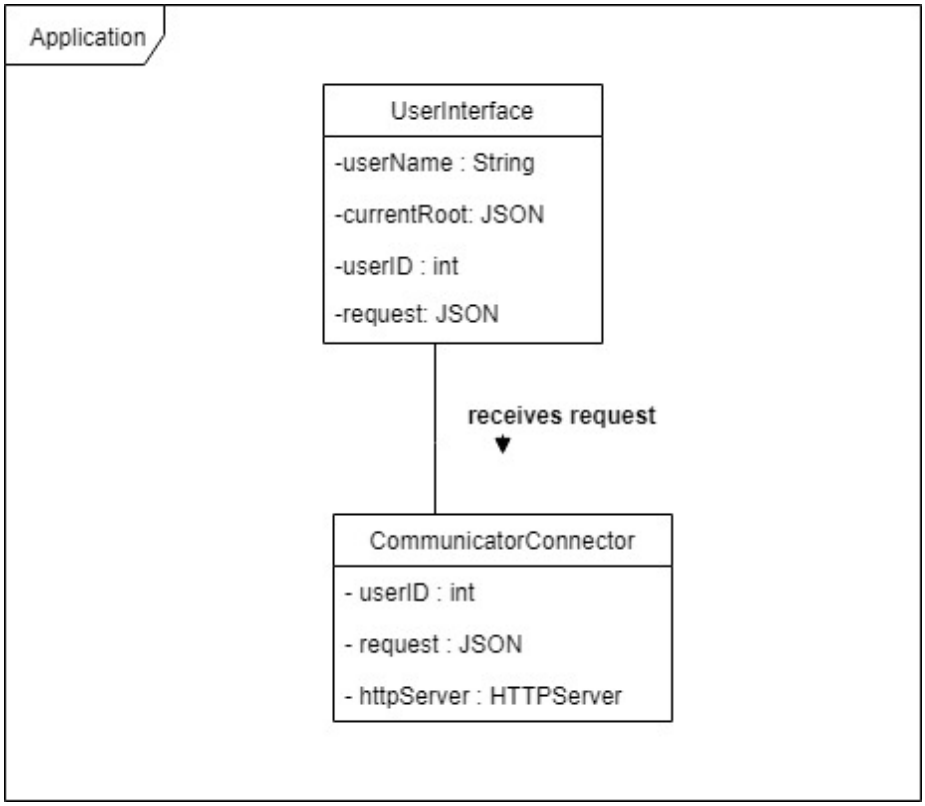




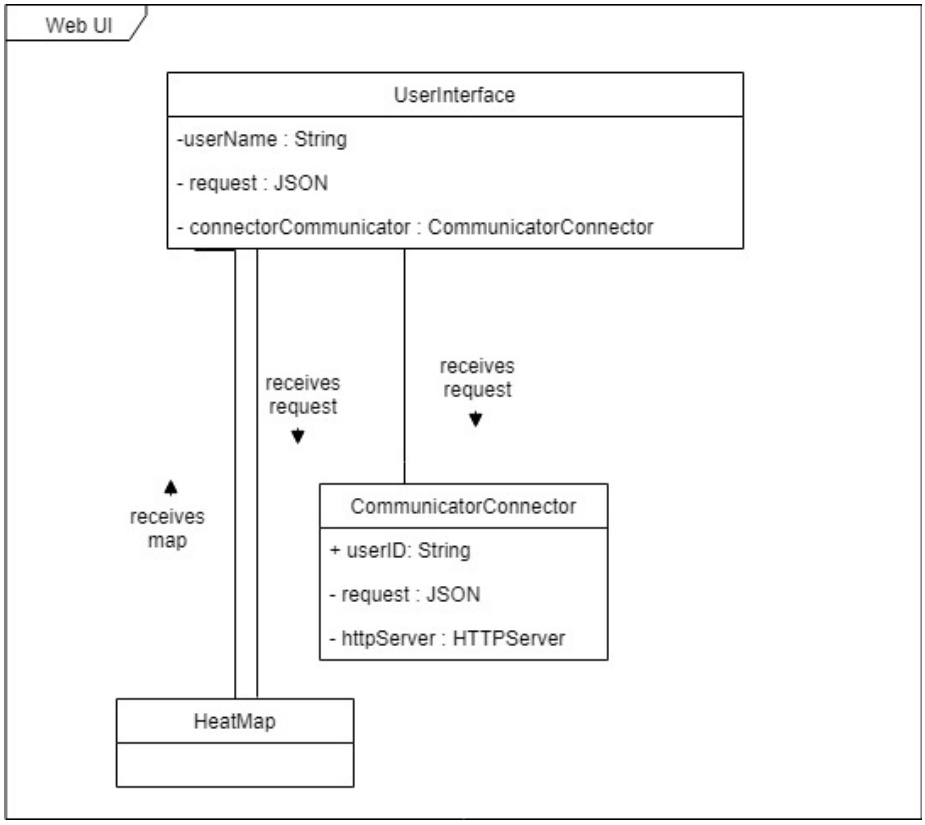
3.2.4 Sensors Subsystem - S3



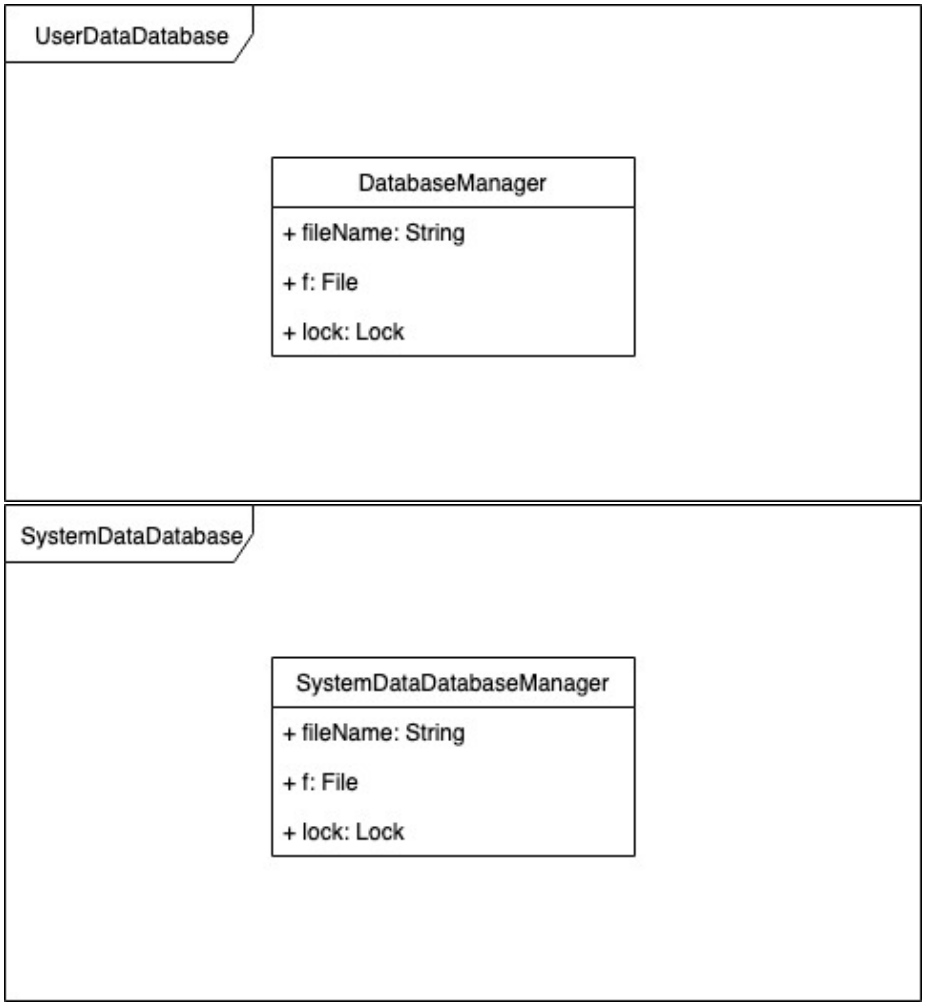
3.2.5 Application Subsystem - S4



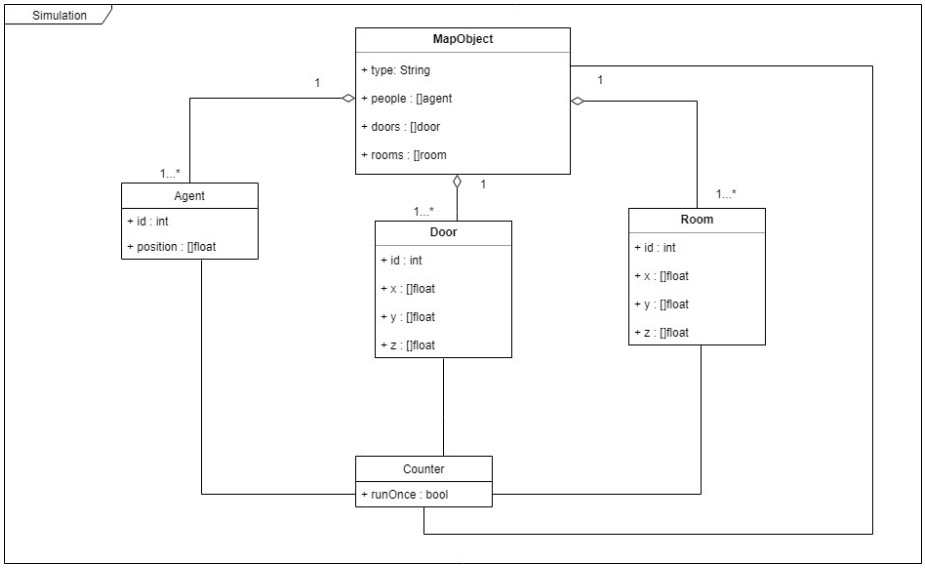
3.2.6 Web UI Subsystem - S5



3.2.7 Database Subsystem - S6



3.2.8 Simulation Subsystem - S7



4 Quality Requirments

4.1 Performance

Performance is a measure of how well the system works as well as the time it takes for operations to take place, under any circumstance(high volume usage, poor signal areas). Essential factors that needed to be tested to ensure that the performance is of a high enough standard are as follows: Testing the system under large population volumes, the speed at which route assignments are calculated and the assignment of the determined routes to the individual agents. A stress test can me formulated in order to see how well the system works under heavy load with the quantitative measure being the systems success: failure rate. However the main metric that will be used to measure the performance testing will be time. The speed at which the system responds and completes its intended task is the critical measurement.

- The system shall be capable to track the current state of the building as well as update the heatmap that represents the state in a real time . The maximum delay that the system is acceptable to have to perform this function would be 30 sec, ideally 5-15 sec, delay.
- The system shall be capable to automatically generate escape routes in maximum of 30 sec delay from the moment fire detectors detected a fire breakage.

- The system shall be able to push generated route to agents with maximum delay of 10 sec.
- The system shall be capable to automatically update pushed routes on each agent's phone in maximum delay of 30-50 sec from the moment the state in the building has been noticeably changed.
- In case if system was off and was only switched on during an emergency, the system shall be able to take no more than 125 sec to be fully loaded, that will include the performance of all the steps mentioned above.

4.2 Reliability

It is imperative that the system works as expected and maintains how it works over the course of its expected life time. The importance of reliability can not be overstated, since if the system fails it can result in the loss of life of the buildings occupants. In the case of the system failing to process data in a timely manner, the system needs to assign the users default routes to begin evacuation. We will check that the software can perform failure-free operation for a specified amount of time across a range of mobile devices. We can then simulate an unacceptable response time and ensure that the system does what has been previously described.

- The system shall generate 100% accurate map of the building with at least 95% accurate escape routes for each agent.
- The system shall be able to work in priority manner and push notification first to those agents that are in potentially more hazardous and dangerous places in the building.
- In case of failure to update the current state of the building, the system shall be able to generate static escape route based on the data that was lastly collected, before the system failed to collect any further data.
- In case of connection failure (in case of poor or no connection or in case of bluetooth signals being lost), the system shall send an SMS to the agents with the short, bullet point descriptions of the escape routes they must take from their location. However, such functionality can only be applied to companies that store their employees phone numbers in their databases.

4.3 Scalability

Scalability refers to the expandability of the system, this is determined by how well it can be implemented in buildings which are densely populated and/or large locations with a large amount of different escape routes. This also extends to how the system will work in areas with perfect signal opposed to

limited signal areas. To test this quality requirement we will test the performance(failure/success rate) of the systems in the various specified areas which have been named above as well as the amount of people able to make use of the new system effectively.

- Ideally the system shall be able to broadcast newly generated route and its updated versions to all the agents all at once. However, realistically it is impossible to achieve. Therefore, the system shall be able to notify at least 50 agents at a time.

4.4 Security

- The system shall not collect any extra data from agents except their current location in the building.

4.5 Maintainability

- The system shall be simple to maintain. The system will be divided into subsystems that would be independent from each other. Therefore, when there is an update in one of the system's functionality, other subsystems will not be affected by that update and the system as a whole would not fail.

4.6 Usability

This quality requirement refers to how easily the system can be learned and effectively used. Any system components which the user interacts with must be designed in such a manner that it feels familiar thus making its operation easy. Since many people have embraced the use of mobile applications, it should not be that difficult to extend this functionality. The system should include the use of a help tab which will provide clear tips/instructions to assist in specific problems. Usability tests will be implemented to measure the effectiveness of the clients interaction with the system. These tests measure 5 performance based on the success rate at performing various tasks and are normally used in the form of questionnaires(after sufficient time using the new system the clients will be prompted to answer and can accept or decline).

- In most of the time the system shall be fully automated. It should perform all the required functions without any or with little guidance. However, in special case like: there is an ignition, but fire detectors failed to perform their functions , system shall provide a simple, one level deep, user interface with all the functions listed on intro page.
- There should be no more than one step (clicking) to evoke any of the main functions of the system.

- The system shall constantly notify a user about any event it performed, or even more important, failed to perform; like in case if system was unable to send newly generated escape routes to agents, the system will send notification to user which will be displayed on status bar (in case of smartphone) or on Windows Push Notification Service.

5 Trace-ability matrix

UsesCases vs susbsystems	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**
UC1	x				x	x	
UC2	x	x					x
UC3	x				x		
UC4			x		x		
UC5	x		x				
UC6 x		x					x
UC7	x			x			
UC8				x			
UC9	x			x		x	
UC10	x		x	x			
Total	7	2	3	4	3	2	2