Concordia University

Smart Home System Simulator

Team 2

Francesco Benvenuto 40019845, section H HC
Ashraf Khalil 40066289 H HB
Martin Marcos 40041398, section H HA
Adam Richard 27059329, section H HA
Derek Ruiz-Cigana 40096268, section H HC

T.A:
Yashika Khurana

Professessor:
Dr. Rodrigo Morales

Delivery 1 due date: October 20, 2020

Table of contents

1.0 Introduction

The smart home simulator allows you to simulate a smart home system on a 2D representation of the house. The automatic functionality of this system includes: home security, home heating system and lighting that allows you to open doors and windows. Also an easy to use user interface allows the user to interact with the simulator, with profile features and authorization. Furthermore, it allows the user to add more modules to the system easily thanks to the modular approach of coding.

2.0 Delivery 1 Scope

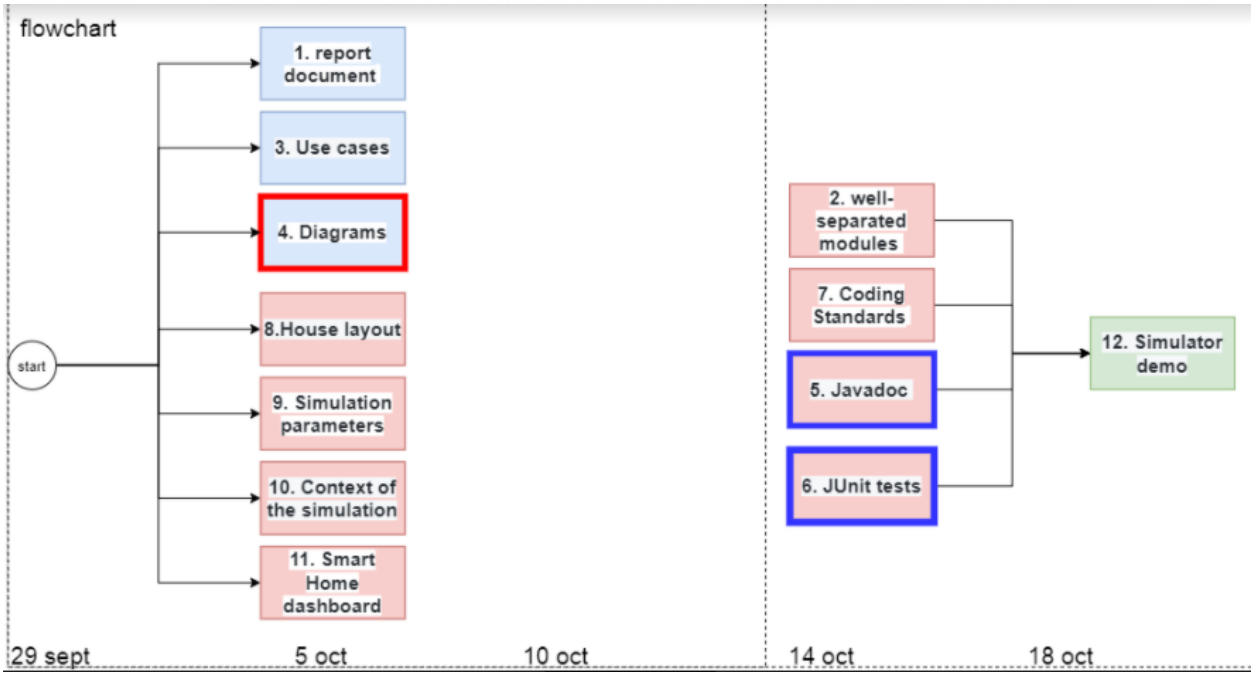

Figure 1: Deliverable 1 scope flow chart

As you can see in Fig1, the scope of this project comprises: documentation, 1 formal document with uses cases and diagrams (1,2,3,4), a working user interface dashboard that users can use to interact with the simulation (turn on lights, temperature change) (8,11), the modules SHC,SHH,SHS and SHP that are the core components that makes the simulation work(9,11,2), Javadoc and Junit tests along with respecting coding standards (7,5,6) and a video demo showing the functionality of the simulator. The project should be done by 18 of October.

3.0 Use Case Tables

3.1 Use Case 1

Use case	Manage house layout files
Level	subfunction
Description	- System can read imputed house layout file from user and load it to the dashboard for to user to interact with - User can add/remove/modify profiles
Preconditions	User has inputted a valid layout file to the system
Triggering event	System reads layout file
Main flow	1. System reads layout file 2. System generates correct number of rooms, doors, windows and lights. 3. System displays visible house layout to the dashboard
Extensions	- User is denied because of wrong authorization - System malfunctions and does not read house layout - Invalid house layout introduced - System outputs incorrect house layout
Postconditions	System recognizes valid layout file imputed and loads it to the dashboard for the user to interact with

3.2 Use Case 2

Use case	Add/remove edit user profiles
Level	User level
Description	User can add/remove profiles
Preconditions	System is correctly installed
Triggering event	The user chooses the option to edit profile
Main flow	1. User powers on the simulation 2. User edits user profiles 3. System receives input and updates accordingly
Extensions	– User is denied authorization – System malfunctions and does not output inputs accordingly – User inputs invalid settings – User is able to access system even though he does not have authorization (authorization failure)
Postconditions	User edits profiles and the system updates accordingly

3.3 Use Case 3

Use case	Set date and time
Level	User level
Description	User can set date and time
Preconditions	System is correctly installed
Triggering event	The user chooses the option to set date and time
Main flow	1. User powers on the simulation 2. User sets date and time 3. System receives input and updates accordingly
Extensions	– User is denied authorization – System malfunctions and does not output inputs accordingly – User inputs invalid settings – User is able to access system even though he does not have authorization (authorization failure)
Postconditions	User sets date and time and the system updates accordingly

3.4 Use Case 4

Use case	Log in using an existing user profile and set house location
Level	User level
Description	User can Log in using an existing user profile and set house location
Preconditions	System is correctly installed
Triggering event	User selects a profile to Log in with and chooses the option to set house location
Main flow	1. User powers on the simulation 2. User logs in using an existing user profile 3. User sets house location 4. System receives input and updates accordingly
Extensions	– User is denied authorization – System malfunctions and does not output inputs accordingly – User inputs invalid settings – User is able to access system even though he does not have authorization (authorization failure)
Postconditions	User logs in and sets house location and the system updates accordingly

3.5 Use Case 5

Use case	Start/stop simulation
Level	User level
Description	User can start/stop the simulation
Preconditions	System is correctly installed
Triggering event	User selects the option to start or stop the simulation
Main flow	1. User powers on the simulation 2. System receives input and updates accordingly 3. User powers off the simulation 4. System receives input and updates accordingly
Extensions	– User is denied authorization – System malfunctions and does not output inputs accordingly – User inputs invalid settings – User is able to access system even though he does not have authorization (authorization failure)
Postconditions	User turns on the simulation to simulate scenarios and turns off the simulation

3.6 Use Case 6

Use case	Modify date and time
Level	User level
Description	User can modify date and time
Preconditions	System has valid parameters set up
Triggering event	User chooses the option to modify date and time
Main flow	1. User powers on the simulation 2. User modify date and time 3. System receives input and updates accordingly
Extension	– User is denied authorization – System malfunctions and does not output inputs accordingly – User inputs invalid settings – User is able to access system even though he does not have authorization (authorization failure)
Postconditions	User modifies date and time and the system updates accordingly

3.7 Use Case 7

Use case	Move the logged user to different room
Level	User level
Description	User can move to a different room
Preconditions	System had valid parameters set up
Triggering event	User chooses the option to move to a different room
Main flow	1. User powers on the simulation 2. Move the logged user to different room 3. System receives input and updates accordingly
Extensions	– User is denied authorization – System malfunctions and does not output inputs accordingly – User inputs invalid settings – User is able to access system even though he does not have
Postconditions	User was moved to a different room, the system receives input and reacts accordingly

3.8 Use Case 8

Use case	Place people in specific rooms
Level	User level
Description	User can place people in specific rooms
Preconditions	System had valid parameters set up
Triggering event	User chooses option to place people in specific rooms
Main flow	1. User powers on the simulation 2. Move a person or people to a specific room 3. System receives input and updates accordingly
Extensions	– User is denied authorization – System malfunctions and does not output inputs accordingly – User inputs invalid settings – User is able to access system even though he does not have
Postconditions	Person or people are moved to a specific room, the system receives input and reacts accordingly

3.9 Use Case 9

Use case	Modify temperature outside home
Level	User level
Description	The user can modify the temperature outside the home
Preconditions	System has valid parameters set up
Triggering event	User chooses option to modify temperature outside home
Main flow	1. User powers on the simulation 2. Modify temperature outside home 3. System receives input and updates accordingly
Extensions	– User is denied authorization – System malfunctions and does not output inputs accordingly – User inputs invalid settings – User is able to access system even though he does not have
Postconditions	User modifies the temperature outside, the system receives input and reacts accordingly

3.10 Use Case 10

Use case	Block windows movement by putting an arbitrary object
Level	User level
Description	User can block windows movement by placing an arbitrary object
Preconditions	System has valid parameters set up
Triggering event	User places an arbitrary object which will block window
Main flow	1. User powers on the simulation 2. Block windows movement by putting an arbitrary object 3. System receives input and updates accordingly
Extensions	– User is denied authorization – System malfunctions and does not output inputs accordingly – User inputs invalid settings – User is able to access system even though he does not have
Postconditions	User blocks window by putting an arbitrary object, the system receives input and reacts accordingly

4.0 Cross-referenced Diagrams

4.1 Domain Model Diagram

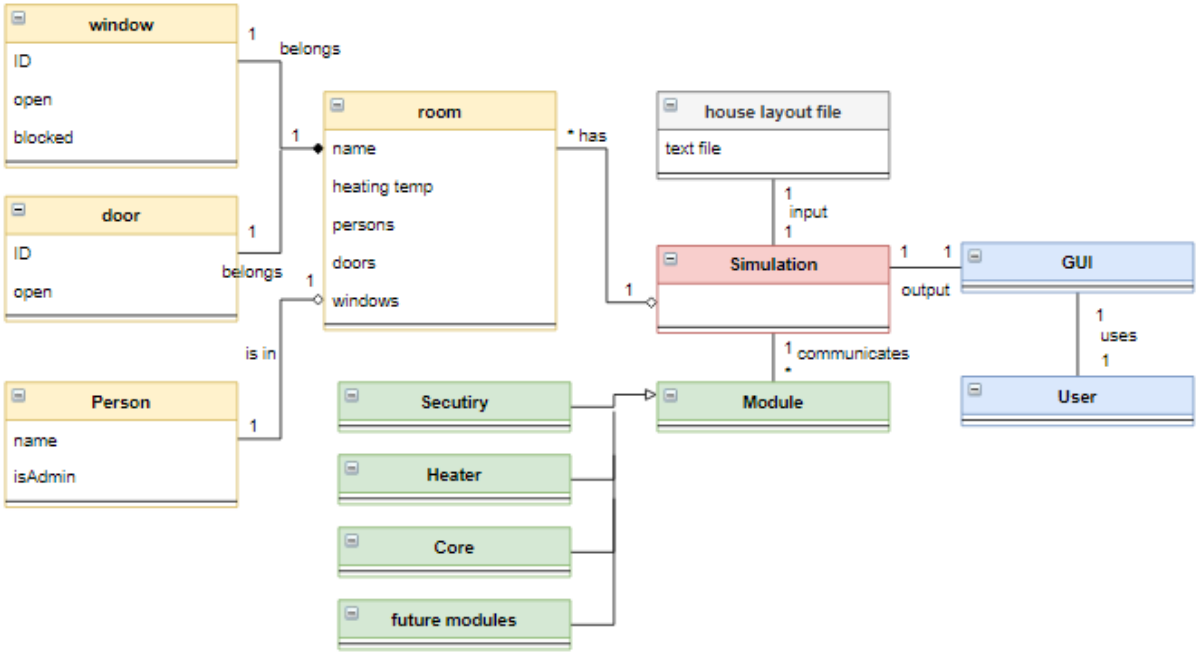

Figure 2: Domain model diagram

4.2 UML Class Diagram

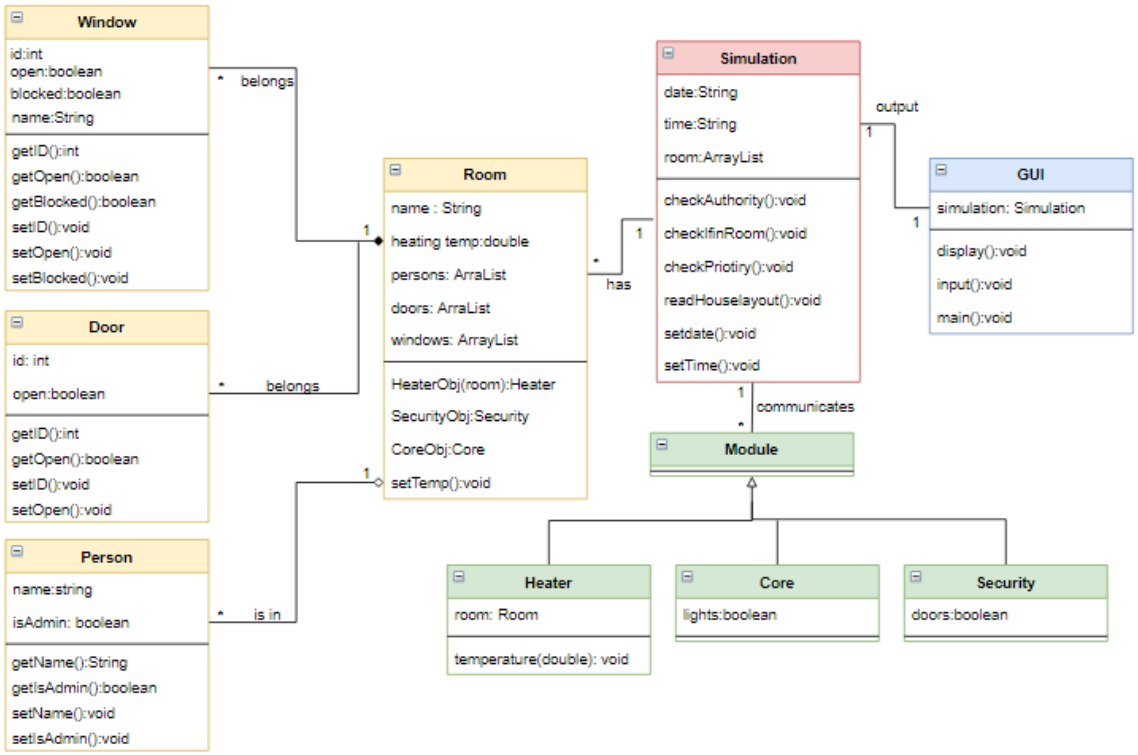

Figure 3: Class diagram

4.3 Sequence Diagram

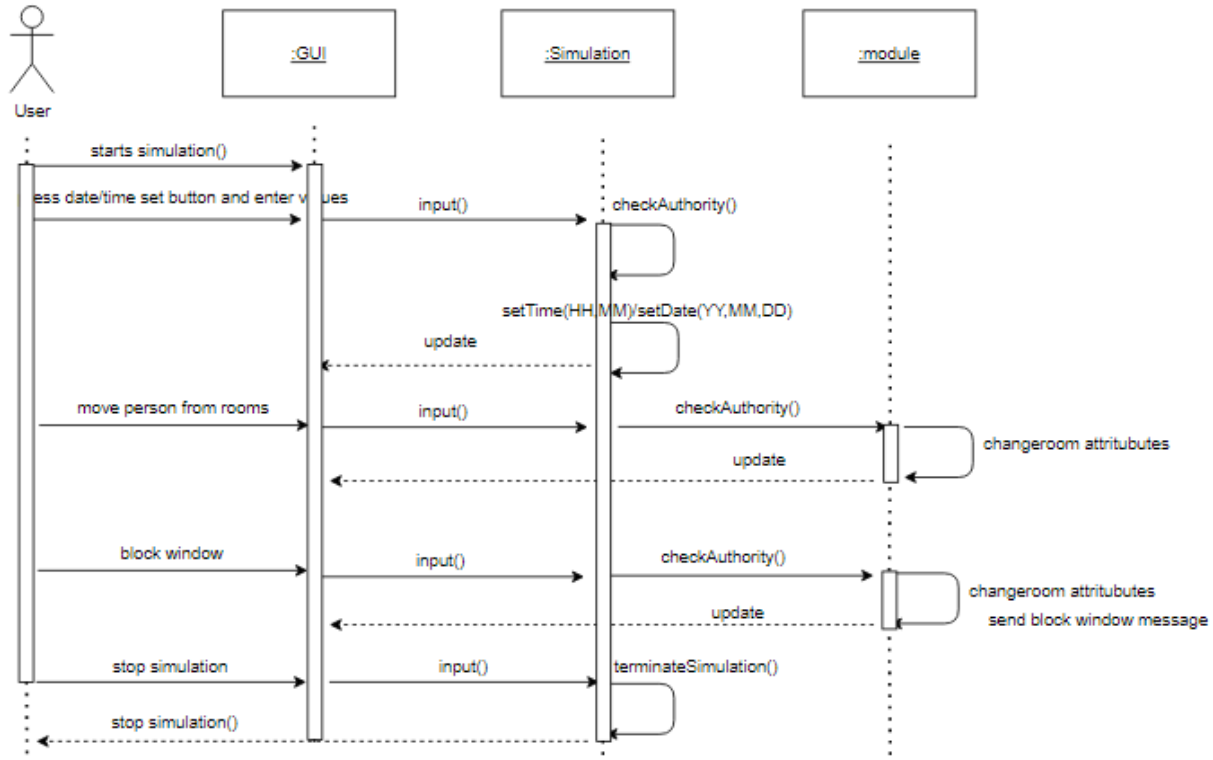

Figure 4: Sequence diagram