Concordia University

Smart Home System simulator

Team 2

Francesco Benvenuto 40019845, section HC

Julien Fadel

Ashraf Khalil

Martin Marcos 40041398, section H

Adam Richard

Derek Ruiz-Cigana 40096268, section H

T.A:

Yashika Khurana

Professessor:

Dr. Rodrigo Morales

Delivery 1 due date: oct 18 2020

Table of contents

Introduction………………………………………………………………………………………………..2

Scope…………………………………………………………………………………….........................3

Use Cases Tables………………………………………………………………………………………...4

Cross-referenced Diagrams……………………………………………………………………………..7

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 authorisation. Furthermore, it allows the user to add more modules to the system easily thanks to the modular approach of coding.

Deliverable 1 scope

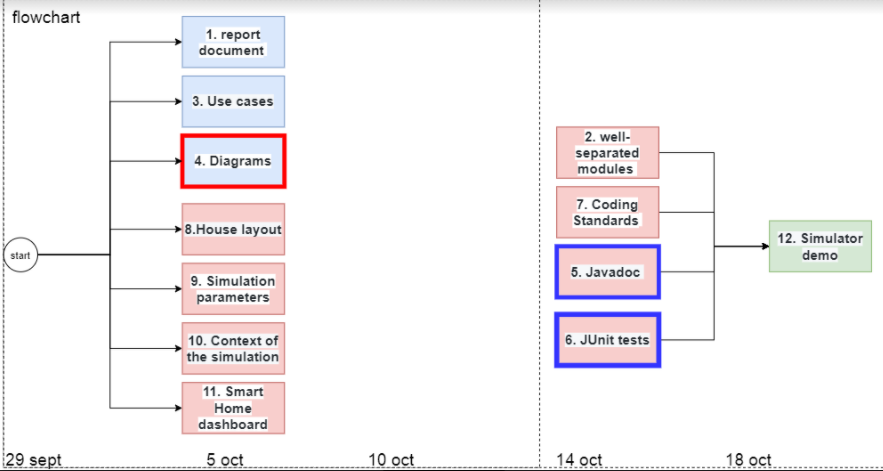


Fig1.Deliverable 1 scope flow chart

As you can see in Fig1, the scope of this project comprises: documentation, 1 formal document with uses cases and diagrams(1,3,4), 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), Javadocs 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 oct.

Use Cases table

|  |  |
| --- | --- |
| 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 authorisation * 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 |

|  |  |
| --- | --- |
| Use case | Manage simulation parameters |
| level | User level |
| description | * user can add/remove profiles * user can set date and time * user can Log in using an existing user profile and set house location |
| preconditions | System is correctly installed |
| Triggering event | User identifies himself and powers on the dashboard |
| Main flow | 1. User powers on the simulation 2. User edits user profiles 3. User sets date and time 4. User sets house location 5. System receives input and updates accordingly |
| extensions | * User is denied authorisation * System malfunctions and does not output inputs accordingly * User inputs invalid settings * User is able to access system even though he does not have authorisation(authorisation failure) |
| postconditions | User edits profiles, sets date and time and house location and the system updates accordingly |

|  |  |
| --- | --- |
| Use case | Performing simulations |
| level | User level |
| description | * user can simulate scenarios and modify parameters * User can modify simulation |
| preconditions | System is has valid parameters set up |
| Triggering event | User identifies himself and powers on the simulation |
| Main flow | 1. User powers on the simulation 2. User modify date and time 3. Move the logged user to different room 4. Place people in specific rooms, or outside home 5. Modify temperature outside home 6. Block windows movement by putting an arbitrary object 7. System receives input and updates accordingly |
| extensions | * User is denied authorisation * System malfunctions and does not output inputs accordingly * User inputs invalid settings * User is able to access system even though he does not have authorisation(authorisation failure) |
| postconditions | user turns on simulation to simulate scenarios and modify parameters, the system receives input and reacts accordingly |

Cross-referenced Diagrams

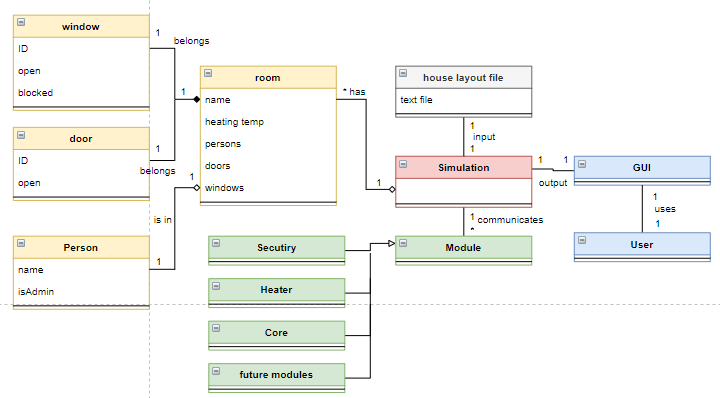


Fig2. Domain model class diagram

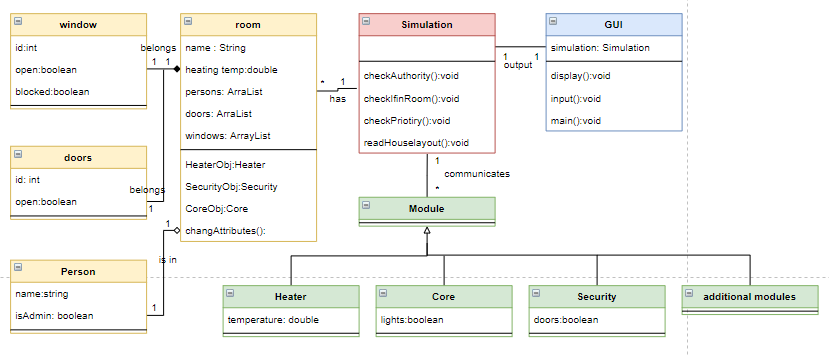


Fig3. Class diagram

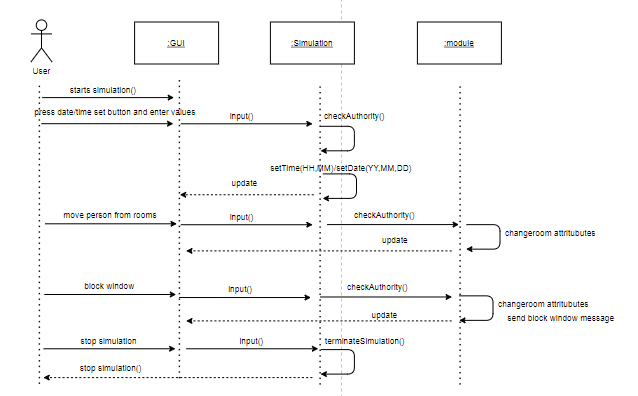


Fig4. Sequence diagram