

**SOEN 343:  
SOFTWARE ARCHITECTURE AND DESIGN**

**Phase II Instructions**

**Instructor: Dr. Rodrigo Morales Alvarado**

**Winter 2024**

# Contents

1 General Information	3
2 Introduction	3
3 Ground rules	4
4 Project description	4
5 Your Project (phase II)	4
5.1 System Architecture	4
5.2 Use Case Model	5
5.3 Design Patterns	6
5.4 Implementation of Use Cases	6
6 What to Submit	8
<b>7 Submission Notes</b>	<b>9</b>
8 Grading Scheme	10
9 Rubric	10
<b>References</b>	<b>10</b>
<b>Appendix A. User Profiles</b>	<b>10</b>
Smart Home Core (SHC) module	11
<b>Appendix B. Use Case template</b>	<b>12</b>

## 1 General Information

**Date posted:** February 15, 2024.

**Date due:** March 11, 2024.

**Weight:** 5% of the overall grade.

**Total:** 100 marks

## 2 Introduction

You must deliver an operational version demonstrating a subset of the capacity of your system. This is about demonstrating that the code build is effectively aimed at solving specific project problems or completely implementing specific system features. The code build must not be just a "portion of the final project", but rather be something useful with a purpose on its own, that can be demonstrated by its operational usage.

You are graded according to how effectively you can demonstrate that the features are implemented. If you cannot demonstrate the features through execution, you will have to prove that the features are implemented by explaining how your code implements the features.

To summarize, delivery 2 is comprised of the following elements:

- A written report in PDF and submitted through Moodle as a team
- Source code of the corresponding deliverable in the GitHub repository (from delivery 1). Remember to add a label in the Git repository named phase <#> where # indicates the number of the phase delivery. You must have provided the link and granted access to the TA and the Instructor

### 3 Ground rules

Same as Delivery 1.

### 4 Project description

A full project description can be found at the following link:

<https://docs.google.com/document/d/1jdzdEo8V0MBYU2sUqjqv5uwJ9OgYd-HyppraUGc0Gm0/edit?usp=sharing>

### 5 Your Project (phase II)

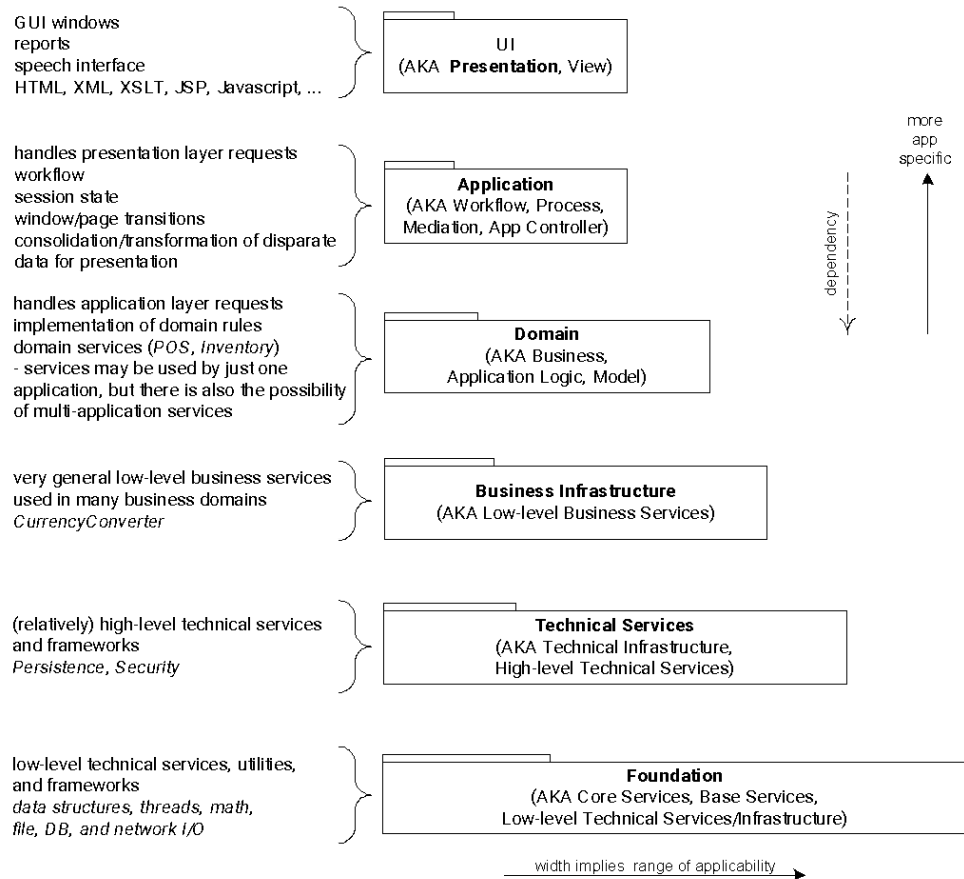
You will work on the following activities:

#### 5.1 System Architecture

The System Architecture (SA) defines the packages within which software classes are defined. In this part of the work, you should define **at least three layers** among the below possible ones. For each layer, you need to provide a description and details of its elements.

- UI layer
- Application layer
- Domain layer
- Business infrastructure layer
- Technical services
- Foundation and domain objects

The Figure below contains details that you might consider when selecting and defining the layers in your app.



## 5.2 Use Case Model

### 5.2.1 Use Case Model

Write the use cases for the following features:

House layout

- Read and load a house-layout file

Simulation parameters

- Add/remove/edit [User Profiles](#)
- Set Date and time
- Log in using an existing user profile and set house location
- System shall allow [simulator's user](#) to grant/deny permissions for executing commands on the registered smart home modules
- System shall save in a file the profiles, with their corresponding permissions, to avoid reentering the information every time you open the simulator

## Context of the simulation

- System shall allow the [simulator's user](#) to change the time speed
- Start/stop simulation
- Modify date and time
- Move the logged user to different room
- Place people in specific rooms, or outside home
- Modify temperature outside home
- Block windows movement by putting an arbitrary object

## Smart home core functionality (SHC) module

- System shall allow [users /smart home modules](#) to open/close doors
- System shall allow [users /smart home modules](#) to open/close windows
- System shall allow [users /smart home modules](#) to turn on/off lights
- System shall allow [users](#) to set an [Auto mode](#), that turn on/off lights automatically when someone enters to a room

## Smart home dashboard

- System shall update the house view to graphically represent the current state of the simulation, i.e., when the simulation is activated (on-toggle button)

Use the template provided in [Appendix B](#).

5.2.2 To complement the use case model section, include the following diagrammatic notations:

- 1 Use-case diagram for the whole system
- 1 Sequence diagram for the Smart home core functionality (SHC) module
- 1 State-machine diagram for the Context of the simulation
- 1 Activity diagram for Context of the simulation

## 5.3 Design Patterns

Implement at least three different Object-Oriented design patterns ([see catalog](#)).

- You must write a paragraph justifying your design decision.
- You must include a UML class diagram to document your implementation of the design pattern.

## 5.4 Implementation of Use Cases

Functional Requirements (software prototype running)	
House layout	Read and load a house-layout file
Simulation parameters	<p>SHS tab with the following options:</p> <ul style="list-style-type: none"> <li>• Add/remove edit user profiles</li> <li>• Set Date and time</li> <li>• Log in using an existing user profile and set house location</li> <li>• In user profiles, you should list the existing modules (e.g, SHC) and their available commands and grant/deny permissions based on the user profile. You need to specify if the permission depends on the location of the user profile or not. For example, if the user is inside or outside the home.</li> </ul>
Context of the simulation	<p>The edit button in the simulation group box opens a form that allows users to perform the following actions:</p> <ul style="list-style-type: none"> <li>• Place house inhabitants in specific rooms, or outside the home</li> <li>• Block windows movement by putting an arbitrary object</li> </ul>
Smart Home dashboard	<ul style="list-style-type: none"> <li>• Smart home simulator user interface</li> <li>• Simulation group completely working including on/off, edit simulation buttons, and links to change current user, location, outside temperature, and date and time</li> </ul>
Smart Home Core functionality (SHC)	<p>SHC tab with the following options:</p> <ul style="list-style-type: none"> <li>• A frame with the rooms in the house, backyard, garage, and building entrance. This information is taken from the house layout file</li> <li>• Buttons to switch between lights, door and windows controls</li> <li>• One light for each room and other areas</li> </ul> <p>The doors that can be automatically locked are located in the building entrance, backyard, and garage (which is a room itself).</p>

	<p>There must exist lights in the building entrance and the backyard. And users or other smart modules can control them. For example the SHP module (implemented in the next delivery).</p> <p>In the lights frame, the user has an option to turn on/off the Auto Mode.</p>
Smart home dashboard	<ul style="list-style-type: none"> <li>• Smart home simulator user interface with a tab for SHC working.</li> <li>• The simulation window fully working, including on/off, edit simulation buttons, and links to change the current user, location, outside temperature, and date and time</li> <li>• House view. Update the <a href="#">house view</a> to represent the current state of the simulation, when the simulation is activated (<a href="#">on-toggle</a> button)</li> <li>• Show icons to represent the following actions in a room: turn on/off a light; open/close door or window; indicate the presence of one or more people inside a room, at the entrance, backyard, or garage or any of the main door access; when away mode is on; when a window path is blocked</li> </ul>

## 6 What to Submit

**6.1** The report is submitted through Moodle by the due date under the corresponding assignment link. It has to be completed by ALL members of the team and only one file in PDF will be accepted.

**6.2** The source code and unit tests are committed to the GitHub repository of the project.

**6.3** Submission template

- 1 Cover page including the name of the team, student names, student IDs, and Lab section
- 2 Table of contents
- 3 System Architecture
- 4 Use Case Model (with cross-referenced diagrams with the use cases)
- 5 Design pattern implementations
- 6 Implementation of Use Cases



## 7 Submission Notes

- Include the names and student IDs of all members of the team in the submission.
- Only one copy of the assignment is to be submitted.
- You must make sure that you upload the project to the correct link on Moodle. No email submissions are accepted. Projects uploaded to the wrong system, or the wrong folder, or submitted via email will be discarded and no resubmission will be allowed. Make sure you can access Moodle before the submission deadline.
- **The deadline will not be extended.**
- Software repository. Well-populated history with dozens of commits, distributed evenly among team members, as well as evenly distributed over the time allocated to Phase 2. A tagged version should have been created for Phase 2. Students will grant access to TAs, and the instructor. Instructor Github user is **moar82**
- The architectural design should be reflected in the implementation of well-separated modules and/or folders.
- One unit test class for each production class, clearly identified with the class name
- Coding Standards. Consistent and proper use of code indenting, naming conventions, and comments; no “commented out” code; the coding standard used is indicated in the Wiki of the GitHub repository
- The README file of the Github repository contains detailed information on how to compile, and deploy the simulator. Platform requirements are specified.
- Naming convention for report: Create a PDF file using the following naming convention. The file should be called D#\_teamName, where # is the number of the project phase, and teamName is the team's name as registered in Moodle. For example, for the first project phase, when team's name is Alpha, they would submit a PDF file named D1\_Alpha.pdf. Submit your project electronically on Moodle based on the instruction given by your instructor as indicated above:  
<https://moodle.concordia.ca>

## 8 Grading Scheme

System Architecture	0~10 marks
Use Case Model	0~30 marks
Design patterns implemented	0~20 marks
Implementation of Use Cases	0~30 marks
Documentation compliance with the template	0~10 marks
<b>Total</b>	<b>0~100 marks</b>

## 9 Rubric

The following [link](#) will provide access to a detailed rubric.

## References

1. How to Design a Java Framework? - A simple example  
<https://www.programcreek.com/2011/09/how-to-design-a-java-framework/>
2. McLaughlin, Brett, Gary Pollice, and David West. Head First Object-Oriented Analysis and Design: A Brain Friendly Guide to OOA&D. " O'Reilly Media, Inc.", 2007.  
<https://resources.oreilly.com/examples/9780596008673/tree/master>
3. Home I/O Simulation of a smart house and surrounding environment  
<https://realgames.co/home-io/>
4. OpenSHS: Open Smart Home Simulator  
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5469526/>

## Appendix A. User Profiles

## Smart Home Core (SHC) module

User	Permissions
Parents	All permissions granted to open/close windows, unlock doors, open/close garage and turn on/off lights
Children	Limited permissions to turn on/off lights and open/close windows on the room that they are located. If they are not at home, all permissions are revoked
Guests	Limited permissions to turn on/off lights and open/close windows in the room where they are located. If they are not at home, all permissions are revoked
Strangers	Non-identified users have no permissions no matter where they are located

## Appendix B. Use Case template

<b>ID:</b>	[Unique ID of this use case]
<b>Title:</b>	[Enter the goal of the use case - preferably as a short, active verb phrase]
<b>Description:</b>	[Describe the goal and context of this use case. This is usually an expanded version of what you entered in the "Title" field.]
<b>Primary Actor:</b>	[A person or a software/hardware system that interacts with your system to achieve the goal of this use case.]
<b>Preconditions:</b>	[Describe the state the system is in before the first event in this use case.]
<b>Postconditions:</b>	[Describe the state the system is in after all the events in this use case have taken place.]
<b>Inputs:</b>	[known inputs for the use case]
<b>Outputs:</b>	[Known outputs of the use case]
<b>Main Success Scenario:</b>	[Describe the flow of events from preconditions to postconditions, when nothing goes wrong.]