

SWE30003 Software Architectures and Design

Assignment 3 – Object-oriented Design (Part II)

11.59 PM, Friday, Week 12 of Semester

Electronic submission (via Canvas)

Detailed Design and Implementation:

For Assignment 2, you were given the task of coming up with an initial object-oriented design of an online system. Based on this initial design, in this assignment (Assignment 3), you are required to (1) carry out a detailed design and implementation for the system and then (2) reflect on the initial design based on your experience with the detailed design and implementation.

Here are a few *ground rules* for this assignment:

(1) Detailed Design

- Refine and extend the initial high-level object-oriented design from Assignment 2 to arrive at a detailed design suitable for direct object-oriented implementation. This may include but is not limited to, refining/adjusting the existing design, adding UI design and adding database design (all in an object-oriented manner).
- You are also allowed to change the design further if, during the implementation process, you find that the initial or extended/detailed design is incomplete, incorrect, or inadequate for the problem at hand.
- All changes to the original design (from Assignment 2) during the detailed design stage and the implementation stage *must be justified and documented accordingly*. Please note that changes, in general, do not only affect the static structure of your design (that is, the class diagram or similar) but may also affect class responsibilities, the bootstrap process, and/or interaction patterns/scenarios at runtime.
- You must provide a *reflection* on the quality of your initial design from Assignment 2. This may include, but is not limited to, the following:
 - Which aspects of the problem did your initial design address adequately?
 - Which aspects were missing from your initial design?
 - What errors were introduced in the initial design?
 - How much interpretation of the initial design was required (i.e., how *ambiguous* was the initial design)?
 - How did you change the initial design to address the omissions, errors and ambiguities (if applicable)?
 - **Lessons learnt:** Given the experience gained from your detailed design and implementation, how would you tackle a high-level initial design problem differently next time?

(2) Implementation

- The implementation can be completed in an object-oriented language *of your choice*. The source code of your implementation must be included in your submission.

- No coding standard is prescribed for use in your implementation, but make sure that your code follows a 'standard' required for professionally developed software (and provide a reference) – part of the marks will be allocated to the use of a suitable coding standard!
- As a simplification, the implementation **does not** need to support payment options, as we cannot have a banking system to validate transactions. The implementation must allow the users to carry out their activities (except for the actual payments, for which some simple message will be sufficient).
- To simplify the implementation task, you may choose to implement part of the system. However, the implementation needs to cover **at least four (4) areas of operation** (see the case study description for Assignment 1) **fully**; in this regard, you need to consider the dependency between these implemented areas so that the chosen areas are fully functional, even when involving dependency on other areas.
- A simple user interface (graphical or textual) will be sufficient. It may help to label each of the input requirements clearly and provide appropriate validation of inputs. For example, the name field cannot be blank, and the correct data type should be used.
- If preferred, files may also be used for persistent data storage instead of using databases.
- You may develop the system as a web-based application or a separate mobile application.

(3) Execution and operation

- Use several scenarios to demonstrate ***all the implemented system capabilities***. Provide brief descriptions of how the user(s) can enter various information required for the different scenarios and steps, including various options (if applicable).
- To demonstrate that your implementation *works correctly*, for each scenario, provide screenshots of your application that, among others, illustrate (i) an 'empty' UI (graphical or textual) at the beginning of the scenario, (ii) takes correct input, (iii) validation of incorrect input, (iv) change or deletion of input when the customer/user had a change of mind, and (v) successful completion of the scenario. (For the payment step, you only need to give a message indicating that it has been processed, and there is no need to carry out actual money-related processing). *Note that instead of using screen shorts, you may record a video that shows all the required scenarios and steps with narration in a structured way.*
- You must explicitly state which *platform* (operating system, IDE, etc.) you used for the development and testing of your implementation and provide a brief description of how your program is to be deployed and run.
- *You must provide evidence* that your implementation compiles, runs, and correctly follows the requirements. Only in exceptional circumstances will a marker deploy and run your code. It is your responsibility to provide, in the submission, ***sufficient evidence*** that the code compiles correctly, the system accepts only correct inputs, and the relevant user requests can be processed as expected.

Note that any clarification questions should be posted to the Assignment 3 discussion forum on Canvas, and emailed questions will not be answered unless it is personal.

Marking Guidelines:

This assignment will be assessed using the following guidelines (110 points):

- Detailed object-oriented design with detailed description (including justification) of any refinements, extensions, changes and non-changes made to the original design from Assignment 2 (*30 points*)
- Discussion of the quality of the original design from Assignment 2 (*20 points*)
- Lessons learnt from the detailed design and implementation experience (*10 points*)
- Completed implementation (max *50 points*)
 - Source code and executable code, including suitable coding standard (20 points)
 - Evidence of compilation (5 points) and correct execution (25 points)

Please note that if (1) *no implementation, an obviously incomplete implementation, or a significantly incorrect implementation* is provided in your submission, (2) the original design from Assignment 2 (i.e., the whole Assignment 2 submission) is not included, or (3) the changes to the design are not adequately justified and documented, **zero marks** will be given to the design, discussion and reflection parts of the assignment and the implementation will be judged based on its merits.

A detailed marking sheet will be provided on Canvas.

Submission Details:

The submission is in electronic form only. An ***electronic copy*** of your solution must be submitted through Canvas ***by the due date and time*** in Week 12 (see the front page of this specification). It must include all the items specified above (including the marking guidelines/marking sheet). The submission should have the following components:

1. The source code and the executable code of your solution must be packaged in a single ZIP or RAR file (or equivalent), and
2. The remaining parts (see the marking guidelines above, i.e., excluding the source code and executable code) are in a single PDF document, including your Assignment 2 submission as an appendix. In preparing these parts of your submission, you need to **organise/structure your submission according to the marking guidelines/sheet and answer the relevant aspects as stipulated in the marking guidelines/sheet specifically (and in the assignment specification, in general)**. For example, for the second part (detailed object-oriented design), you need to include the final class diagram capturing your final detailed OO design and provide the change details and justifications for the class level, the responsibility level and the dynamic aspects (bootstrap and scenarios), with corresponding headings. (A video can be used as evidence of compilation and correct execution.)
3. The appropriately signed and completed "**Assignment and Project Cover Sheet**" declaration form must also be submitted.
4. "**A contribution document**", signed by *all* group members, that
 - a. lists the amount of time spent by each member on each significant part of the assignment,
 - b. describes briefly the contributions made by each group member and

- c. provides evidence showing that the assignment is done through *true* group collaboration, e.g., discussions and reviews of all major parts of the assignment.

Note: For this assignment, students are to work in groups of three or four, i.e., the same groups as in Assignment 2. Permission by the Tutor is required to *change groups well before* the submission deadline. Extensions to the submission deadline can only be granted for genuine reasons, and the Unit of Study convener must be contacted at least 48 hours prior to the submission deadline or at the earliest possible time.

The Unit of Study convener and/or the tutor reserve the right to call in any student/group to explain further and demonstrate their submission if there are doubts about the authorship of the presented solution and/or the completeness thereof.

Swinburne University of Technology Sarawak Campus

School of Information and Communication Technology

ASSIGNMENT AND PROJECT COVER SHEET

Subject Code: SWE30003

Unit Title: Software Architectures and Design

Assignment number and title: 3, Implementation Project Group No.: _____

To be completed as this is an individual assignment

I declare that this is an individual assignment and that no part of this submission has been copied from any other student's work or from any other source except where due acknowledgement is made explicitly in the text, nor has any part been written for us by another person.

Student ID

Name

Signature

Marker's comments:

Total Mark: _____

Extension certification:

This assignment has been given an extension and is now due on _____

Signature of Convener: _____