ITEH7201 Software Engineering Analysis and Design

Assignment 2

Overview

For this assignment, you will implement the System Design you did in your first assignment. Therefore, same team will continue work on the same System case scenario they designed in the first assignment. Each student works on the subsystem they worked in the assignment 1. We will follow **paired programming concept in that each student's code should be reviewed by another student.**

Timelines and Expectations

Percentage Value of Task: 35%

Due: Saturday 25 May 2024, 11.55pm

Minimum time expectation: 30 hours

Learning Outcomes Assessed

The following course learning outcomes are assessed by completing this assessment:

- Understand the significance of detailed project planning and control, good communication and documentation and the use of appropriate tools in order to provide a quality product
- Understand the distinction between software engineering and programming, and thus the distinction between a software configuration and a program
- Understand the methods and techniques involved in designing, implementing and maintaining an information system, in particular using an object-oriented approach
- Demonstrate skills in designing and implementing an information system

Learning Outcomes Assessed

The following course learning outcomes are assessed by completing this assessment:

- Understand the significance of detailed project planning and control, good communication and documentation and the use of appropriate tools in order to provide a quality product.
- Understand the distinction between software engineering and programming, and thus the distinction between a software configuration and a program.
- Understand the methods and techniques involved in designing, implementing and maintaining an information system, in particular using an object-oriented approach.
- Understand how unit tests are used during software development to assist in agile programming techniques such as refactoring.

- Work together in small teams to complete a fully documented, detailed design and implementation of a small business information system.
- Demonstrate skills in designing and implementing an information system.
- Demonstrate skills in designing Unit tests.

Assessment Details

Tasks:

Please note that you need to use AI for some of the tasks. Please check the marking guide before starting the work.

1. **Interface and inheritance**: Your application must show use of both interface and parent/child inheritance implementation. You can use any number of interfaces and inheritances as require.

2. **Implementation:** Each student is required to implement the subsystem based on the analysis and design carried out in the first assignment. This includes translating the use case diagrams, use case descriptions, class diagrams and sequence diagrams into actual code using Java programming language and suitable development environment. **Your document *must include the class diagram (which must have been updated from assignment 1, as you would have used many classes with new relationships)*** and make this document more like a technical document rather than filler.

3. **Design patterns:** Students must incorporate state and singleton design patterns where appropriate to enhance the structure and efficiency of their code. Must explain the role of design patterns in your system, clearly identifying the functionality where design patterns have had an impact and how this has occurred.

4. **Writing Test Cases:** Students must develop comprehensive test cases to validate (run the test cases on the code) the functionality of their implemented system. Test cases should cover various scenarios, including normal operation, edge cases, and error handling. Each test case should be documented clearly, including the expected outcomes.

   Write at least 15-unit tests (5 per student), but your tests must use each of assert statements (8) outlined in your course and annotations @before and @after at least once.
   (Assert statements such as assertEqual, assertNull, assertNotSame) you are developing in this assignment. You can write the test cases in any Java framework, but Junit is preferred.

5. **Code Review:** Students will participate in a code review process where they review and provide feedback on the code written by their team member. Each student will conduct a thorough review of one team member's code. **Make sure, each member's code gets reviewed and each member reviews one team member's**

code. The focus of the code review will be on readability, efficiency, adherence to coding standards, error handling, and overall quality.

6. You must use **Github** for repository management. Your repository must show at least three commits (at least one from each student with significant changes in code. If you have only two members in team, you must have at least two commits.)

Assignment Requirements In Pairs:

Do not work with your partner or any other person to complete your individual report. These must be unique and your own work.

Please note that assignments will NOT be marked and zero marks will be allocated if the individual statements of personal and partner contributions are not submitted.

Submission

Each student must submit a single zip file which contains all assignment files in the Assignment 2 submission box provided in Moodle. Submission files include a clear photograph or a scanned image of your hand-drawn map, code for each deliverable/milestone, presentation file, and an individual report containing your student number, name, your partner's student number and name, diagrams, user stories and contribution statements. A link to your demonstration video – anonymous link on YouTube - should be in your report.

Marking Criteria/Rubric ()

Student IDs		
Student Names		
Pre-Requisites for Marking □ Statement of personal contribution and partner's contribution		
Task	**Max Marks**	**Obtained Marks**
Group Tasks-30 marks (all team members work together)		

1.Functionality (Code delivers the expected outcome when executing the program)	5	
2. i) Implementation of Singleton pattern (at least in one place in the whole system – group work) – 5 marks ii) Explanation of where and why this pattern is used (5 marks)	10	
3. i) Implementation of Sate pattern (at least in one place in the whole system– group work) ii) Explanation of where and why this pattern is used (5 marks)	10	
4. Make a video showing entire functionality – focus is on showcasing the functionality not on explanation	5	
Individual Work (50 marks)		
1.User stories for the subsystem you are working on.	5	
2.For your subsystem, generate code using AI (each student implements their own subsystem) i) What questions you asked (2 mark) ii) What errors you found and how you rectified the errors (4 marks) iii) Final code for the subsystem (4)	10	
3. Interface and inheritance are used. Each student implements this in their subsystem. i) what questions you asked to have interface and inheritance in your subsystem (2 mark) ii) What errors you found and how you rectified the code (4 marks) iii)Explanation of how and why you used this interface and inheritance (4 marks)	10	
4. Use of Github and multiple commits. Each student commits independently. Commit unreviewed code first (5 marks) and then revised code after review (5 marks). Keep the review comments in the revised code. Write the summary of review in the report as well.	10	
5.Unit tests (Each student writes at least 5-unit tests for their subsystem and validates). Use AI to get the unit tests. i) what questions you asked (2) ii) what errors you found and how you rectified them (4)	10	

iv) Final test plans (4)		
6. Code review (Each student reviews code of other student and list out the issues found). Don’t forget to mention which student’s subsystem you reviewed.	5	
Total	80	
Final Mark out of 80	/35	

Feedback

Assessment marks will be made available in fdlMarks, feedback to individual students will be provided via Moodle or as direct feedback during your tutorial class.

Academic Misconduct:

To submit your assessment task, you must indicate that you have read and understood, and comply with, the Federation University Australia Academic Integrity and Student Plagiarism policies and procedures

http://policy.federation.edu.au/learning_and_teaching/compliance/academic_integrity/ch02.php

You must also agree that your work has not been outsourced and is entirely your own except where work quoted is duly acknowledged. Additionally, you must agree that your work has not been submitted for assessment in any other course or program.

Federation University General Guide to Referencing:

The University has published a style guide to help students correctly reference and cite information they use in assignments. A copy of the University’s citation guides can be found on the university’s

web site. It is imperative that students cite all sources of information. The General Guide to Referencing can be purchased from the University bookshop or accessed online at:

https://federation.edu.au/library/guides/referencing

Submission

Submit the report and zip file of the code in the dropbox of Moodle for assignment 2. Share the Github repository with your lecturer and tutor. One student submits the group report and the individual tasks, other students submit only individual tasks.