TMF2243/TME2413/TMI2113/TMN2223/TMS2833/TMT2673

Object-Oriented Software Engineering/ Object-Oriented Software Development Group Project (40%)

Due Date: 3rd Jan, 2025 (Friday 3 pm)

This group project is a continuation from Assignment. Please stick to the same group that you have in your assignment.

# Tasks

1. **Identify the classes** in the proposed system.
2. **Draw a class diagram** to show the structure of your classes and relationships that they may have with one another. Your class diagram should exhibit the object-oriented concepts of **polymorphism and inheritance**.
3. **List possible attributes and methods** for each of the classes identified above.
4. **Design the console-based user interface** to align with your class diagram and object-oriented design principles.
5. **Implement the classes** using an object-oriented programming language of your choice (Python/C++/Java/C# and only object-oriented programming language is allowed). Your implementation should reflect the structure of your classes (including the user interface design) accordingly and exhibit the object-oriented concepts of **polymorphism, inheritance, association and encapsulation**. Groups using Python will receive a bonus mark (see the marking scheme for details).
6. **Implement the user interfaces and methods** of the classes. It is not necessarily to implement the methods in detail. It is sufficient if each method can print an output on the screen so that we know which method is called.
7. **Prepare at least 5 test cases** (*other than the ‘login’ test case* sample as it is shown in the Test Case template already).
8. **Implement the driver (main program)** which will test all the classes and methods declared according to the **class diagram** and test according to the **test cases**.

# Group work

Remain to use the same Microsoft Teams/Online Group Discussion tools you set up for your Assignment. Again, you might be required to let the lecturer join the Team to observe team work among members. Recommendations:

* + Discuss on the class diagram, user interface, test cases and program.
  + All members do the class diagram.
  + Each member will design a user interface, implement a class, prepare a set of test cases and test another set of test case.

# Presentation

1. Your group will be required to present your project at the end of the semester.
2. A link will be provided on eLEAP to register for a presentation slot nearer to the time of submission.

# Report

1. Your report must **document all the tasks** performed above in a document file.
2. Clear state the name and matric number of the group member who contributed to the part of the report.
3. Your report must **include the test cases** that you had prepared, which should follow the template given in this project.
4. Your report must not exceed **8 pages (excluding cover page and table of contents)**.
5. Your report must include the **assignment cover sheet completed with signatures of all members**. Use the FCSIT standard cover with all information correctly filled.

# Submission

1. Due date for submission is on **Week 12 (the due date will be updated again later)**. Late penalty of *-5 marks per day* applies.

# All members must sign the Integrity Form.

1. Please submit **softcopy of your report, signed Integrity forms and soft copy of your coding**. Softcopy submission can be done through a link provided in eLEAP according to your group.

# Important Note

1. This is your **final assessment** for this course and all the rules for final examination are applied.
2. Follow all the instructions given in this project.
3. High quality projects and substantial differences between each group project are expected.
4. Highly similar work can be considered as plagiarism. **Plagiarism** will not be tolerated and similar work from multiple groups you will be awarded a **FAIL** for this course.
5. **No contribution** to the project will also be awarded a **FAIL** for this course.
6. Safeguard your work in reliable storage and make duplicate copies. Excuses such as lost files, no internet access, computer damage or stolen, storage damage and the likes will not be entertained.

**Report Outline**

1. Cover Page
2. Table of contents
3. Class Diagram (2~ 3 pages)

* Brief descriptions of key classes and their purposes within the system
* Class Diagram – a visual representation of the system’s class structure, showing relationships like inheritance, association, and aggregation
* Brief explanation of how core OO principles (e.g., inheritance, encapsulation, polymorphism) are implemented in the design

1. User Interface Design (OO rules) (2~3 pages)

* **Modularity and Cohesion - Brief explanation how each UI module focuses on a single function without overlapping tasks.**
* **Low Coupling - Brief description how each module works independently, with minimal dependencies on other modules.**
* **Consistency - Show examples of commands used in the UI, with syntax and terminology matching the class diagram and using a consistent format.**

***\*\*\*\* Include relevant examples or visuals, like mockups or screenshots, to support your explanations.***

1. Test Cases (2~ 3 pages)

* Table summarizing test cases, including inputs, actions, expected results, and actual outcomes (refer to the Test Case Template)

**TEST CASE TEMPLATE**

**Class Admin**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Module** | **Test Case ID** | **Test Case Name** | **Created by** | **Date Created** | **Tested by** | **Preconditions** | **Test Inputs** | **Actions** | **Expected Output** | **Actual Result** | **Pass/**  **Fail** | **Comments** |
| Login | TC-01 | Successful Login | Ali bin Ahmad (xxxxx) | xx/xx/2024 | Samantha James (yyyyy) | Admin account exists | Log in ID: `admin`, Password: `admin12#` | Enter log-in ID and password, then click 'Login'. | Admin main page displays successfully. | Admin page displays | Pass |  |
| Login | TC-02 | Failed Login - Incorrect Password | Ali bin Ahmad (xxxxx) | xx/xx/2024 | Samantha James (yyyyy) | Admin account exists | Log in ID: `admin`, Password: `wrongpass` | Enter log-in ID and incorrect password, then click 'Login'. | Error message stating 'Incorrect password.' | Error message displayed | Pass |  |
| User Management | TC-03 | Add User - Valid Input | Ali bin Ahmad (xxxxx) | xx/xx/2024 | Samantha James (yyyyy) | Logged in as Admin | User email: `user123@gmail.com`, Password: `123user$` | Click 'Add User,' enter email and password. | User `user123@gmail.com` successfully added; confirmation email sent. | Confirmation email sent | Pass |  |
| User Management | TC-04 | Modify User - Invalid Email Format | Ali bin Ahmad (xxxxx) | xx/xx/2024 | Samantha James (yyyyy) | Logged in as Admin, user exists in system | Existing User ID: `user123@gmail.com`, New Email: `invalid-email-format` | Access 'Modify User', enter User ID and invalid email, save changes. | Error message: 'Invalid email format.' | Email saved without error message | Fail | Needs email format validation |
| User Management | TC-05 | Delete User - Non-Existing User ID | Ali bin Ahmad (xxxxx) | xx/xx/2024 | Samantha James (yyyyy) | Logged in as Admin | User ID: `nonexistent@gmail.com` | Enter User ID, click 'Delete User.' | Error message: 'User not found.' | Error message displayed | Pass |  |