**Assessment Task 1 – Initial project work 10%**

**ASSESSMENT TASK 1: DESCRIPTION**

You are given a programming challenge that consists of frontend GUI Java programming and backend application logic programming using appropriate Java APIs and your own Java class code. The first part of the coding project assesses your ability to interpret project requirements and start developing a software solution to the challenge. You are expected to: 1) construct appropriate user stories based on the challenge requirements, 2) create a UML class diagram that reveals the overall structure of your work, 3) establish a GitHub repository for your work, and 4) start developing a coded prototype that implements the important user stories central to the incremental development of your software solution and is based on the UML class diagram.

The quality of your code is assessed in terms of: 1) general readability, 2) application of Java coding standards, and 3) your ability to apply good coding practices as discussed during the subject.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task** | **Exemplary 4**  **HD**  **85 - 100%** | **Good 3**  **D**  **75 - 84%** | **Satisfactory 2**  **C**  **65 - 74%** | **Limited 1**  **P**  **50 - 64%** | **Very Limited 0**  **F**  **0 - 49%** |
| 1 User stories and UML class diagram | Very well-formatted. Very easy to read and understand. A meaningful collection of user stories exists including those that are central to solving the challenge. A well-structured UML class diagram that details the main classes in the program code. | Exhibits aspects of exemplary (left) and satisfactory (right) | Well-formatted, easy to read and understand. A collection of user stories exists but some lack meaning or utility. The UML class diagram is missing important details. | Exhibits aspects of satisfactory (left) and very limited (right) | Too many readability issues. |
| 2 Use of GitHub | The majority of commits so far in the project repo are well motivated and represent significant steps during software development. | The repo contains some commits that aren’t well motivated, or don’t represent significant steps during software development. | GitHub is not used well enough. |
| 3 Progress made on prototype solution  **(Double marks)** | Outstanding progress is made. Important user stories are very well implemented. Excellent progress is made on the remaining user stories. | Reasonable progress is made. Most important user stories are implemented. No progress is made on the remaining user stories. | Not enough progress is made. |
| 4 Code testing | Very useful tests exist that carefully check the stability of prototype code implemented so far. | Some prototype code has corresponding test code. This test code is somewhat effective at checking the stability of prototype code. | Not enough test code exists. |
| 5 Code readability | Java coding standards are followed very carefully throughout the prototype code. | Java coding standards are mostly followed except in some parts of the prototype code. | Too much of the code is not following Java coding standards. |
| 6 Coding practices | The coding practices covered in the first part of the subject are utilised very effectively. | Coding practices are utilised, however there are some exceptions. | Coding practices are not utilized well enough. |