**Assessment Task 2 – Final project work 20%**

**ASSESSMENT TASK 2: DESCRIPTION**

The second part of the coding project assesses your ability to continue developing your software solution to the challenge. You are expected to: 1) adjust user stories as necessary, 2) implement GUI and event programming behaviours, 3) continue effective use of GitHub for your work, and 4) complete the coded prototype by implementing all user stories.

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 diagram improvements | Very well-formatted. Very easy to read and understand. All necessary changes to them have been made. | Exhibits aspects of exemplary (left) and satisfactory (right) | Well-formatted, easy to read and understand. Some changes have been made. | Exhibits aspects of satisfactory (left) and very limited (right) | Too many issues. |
| 2 GitHub | The majority of commits in the project repo are well motivated and represent significant software development steps. | 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 Prototype solution outcomes  **(Double marks)** | The prototype is an excellent solution to the challenge. The GUI logic and application logic are very well implemented, and very closely relate to the user stories and UML. | The prototype represents a good solution to the challenge. However, a few aspects of the implementation are unfulfilled, and/or a few user stories or some aspect of the UML diagram remains poorly implemented. | 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 carefully throughout the prototype code. | Java coding standards are mostly followed well except in some parts of the prototype code. | Too much of the code is not following Java coding standards. |
| 6 Coding practices | All coding practices covered in the subject content are utilised very effectively in the prototype code. | Coding practices are utilised well with a few exceptions. | Coding practices are not utilized well enough. |