

CS-UH

2012

Software Engineering

Project Deliverable 4 (Construction & Testing)

Purpose:

You are now at the stage of implementing your designed system. In this deliverable, you will pick three out of the five use cases you selected in deliverable 3. You will implement and test these three selected use cases. The goal is to have a fully working prototype for these selected functionalities, while testing them.

Submission Instructions:

* Submit a PDF of this report on Brightspace
* Submit your prerecorded video for section 3 as a separate file in the submission on BrightSpace. Alternatively, you can upload your video on as an unlisted YouTube video and include the URL in section 3. In either case, please specify how you submitted your video in Section 3.

This deliverable is graded out of 100 marks

Please note that this deliverable includes a bonus component worth 10 marks

Section 1: Feature Description (Total 3 marks)

Include the three selected use cases that you are implementing and testing.

# Section 2: Technical Details and Assumptions (Total 5 marks)

2.1 Technology stack (3 marks): Briefly describe the technology stack (Tech stack) used to develop your project, including the testing tools you employed.

2.2 Assumptions or Exclusions (2 marks): List any scenarios from your selected use cases that are not included in the implementation or testing. Or any assumptions you have made while implementing/testing.

# Section 3: Functional Prototype (40 marks)

Include a short (e.g., 1-3 min) video of a demo of the three selected use cases you implemented and tested. The starting point of the video is compiling and running your system, and then demo how a user would experience the selected use cases (e.g., if this is a hospital admin app, how would a doctor check a patient record and prescribe a medication to them using your app). You are free to narrate what is happening using your voice or text descriptions on the screen.

# Section 4: System Test Cases (50 marks)

List your created system test cases for all your project (at least 25 test cases). Some of these test cases will belong to the three selected use cases (at least 10 test cases); while others belong to the future test cases that you will implement for Deliverable 5. If the test case belongs to one of the already implemented use cases for this delivery, then you have also to show the actual results of executing the test case (e.g., screenshot).

Make sure to select meaningful test cases that demonstrate essential functionality from your system (i.e., not all are related to log in/log out or user registration).

Follow the following test case template, which also allows test to requirements traceability:

* Test Case Title: Make sure to have a meaningful and clear title
* Use Case Tested by the Test Case: Make sure to also indicate the selected scenario in the use case, e.g., main success scenario, exceptional scenario 2a etc.
* Requirement: reference from the related requirement(s) from your SRS.
* Technique Used: Indicate the technique(s) you used to create the test case and to select the specific input values used in the test case. (e.g., Boundary Value Analysis, Equivalence Partitioning, Pair-Wise testing etc.)
* Pre-condition: Indicate any assumptions about the state of the system, previous actions taken by the user, or the start point of the test case.
* Steps to Execute: Include the EXACT steps the tester executing this test case should follow. If a step has expected inputs (e.g., entering some value in a text box or selecting a specific date), you must indicate the exact value the tester must input. These steps should be exactly reproducible without any chance for ambiguity that may lead to different interpretation.
* Expected Results: Describe exactly what the executer should see (e.g. should they see a specific error message, should they see their cart updated with the number of items they added etc.).
* Include a screenshot of the result after you executed the test case, if the test case belongs to one of the three use cases you already implemented for this deliverable.

# Section 5: Reflection on Software Quality (2 marks)

Provide suggestions to enhance the software's quality based on the testing results. For example, do you have failing tests that need improvement? Was writing tests difficult due to high coupling? Did you have to fix many bugs along the way due to poor development practices such as lack of proper code review? In this section, you can provide any real insights you have obtained so far, reflecting on what we have learned in class and your first-hand experience testing the system. You will not be graded negatively if you have faced issues along the way; the point is to reflect and learn.

# \*\*SECTION 6: (BONUS): Unit Testing (10 points) \*\*

Include screenshots of the results of running your unit-test suite. The screenshots need to show: (1) how many tests are in your test suite, (2) how many tests pass (ideally 100%), and (3) what is the statement coverage of these tests. At this stage, you should have ideally 70% statement coverage for the selected functionality that you implemented.