National College of Ireland

BSc. (Hons) in Computing, Year 2, BSHCIFSC2
BSc. (Hons) in Computing, Year 2, BSHCIFSC2_B
BSc. (Hons) in Computing, Year 2, BSHCIFSC2_ColaisteDhulaigh
BSc. (Hons) in Computing, Year 2, BSHIFSC2_Rathmines
Higher Certificate in Computing, Year 2, HCCOMP2

Tuesday 5th March 2024
Saturday 20th April 2024 at 23:55
Dr A. E. Chis
Dr T. Jamal

Software Quality and Testing
Continuous Assessment (CA) Type:
Project

Weight: The assignment will be marked out of 100. This CA represents 50% of the overall marks for this module.

Instructions: You are required to submit all the project deliverables through Moodle via the dedicated submission pages.

SUBMISSION DETAILS: All the project deliverables must be submitted via Moodle before the deadline using the dedicated submission pages. **Late submissions will be penalised** unless applied for an extension through NCI360 and approved. The report deliverable should be formed from paragraphs and should NOT contain ordered and/or unordered (e.g., bullet points) lists.

Academic Integrity: IMPORTANT: It is your responsibility to avoid plagiarism. Please read the comprehensive guidelines on academic honesty and academic integrity, and how to avoid plagiarism made available by the NCI Library (https://libguides.ncirl.ie/referencingandavoidingplagiarism).

It is expected that the Submitted Assignment is FULLY THE WORK OF THE STUDENT except where quoted material is clearly indicated. The use of Artificial Intelligence (AI) text generators (e.g., ChatGPT, Quillbot, GitHub Copilot, etc.) to generate/rewrite/paraphrase text and/or to generate code could be considered a breach of academic integrity as you are not doing the work yourself and are using AI to carry out the work YOU should be doing. Paraphrasing is a very important part of an assignment as it proves a person understood the information enough to put it into their own words.

NOTE: YOU ARE NOT ALLOWED TO PUBLISH THIS ASSIGNMENT BRIEF OR A PART THEREOF ON ANY WEBSITES. YOU ARE NOT ALLOWED TO PUBLISH/SHARE YOUR SOLUTION WITH OTHERS. All work submitted should be YOUR own. Conferring with others is NOT permitted. This is NOT a collaborative assessment.

All submissions will be electronically screened for evidence of academic misconduct, e.g., plagiarism, collusion, AI use, misrepresentation, etc. Any submission showing evidence of such misconduct will be referred to the College's Academic Disciplinary Committee.

TURNITIN: All submissions will be electronically screened for evidence of academic misconduct (e.g., plagiarism, collusion, AI generated text and/or code, misrepresentation, etc.).

1. Project Description

For this assignment, you are required to practically test and document through a report the entire testing strategy you have carried out for testing a medium-scale software application that you have created.

2. Project Assessment

2.1. For students who take the Team Project module – this is a Cross Module Assessment

For the project assessment of the Software Quality and Testing module, students, who are taking the *Team Project* module this semester, will have to test the application that they develop as part of the *Team Project* module. Consequently, as the assessment for the *Team Project* module requires students to work in teams, this is a team project, and you work as part of the same team that you have formed for the *Team Project* module.

Note that all students receive individual marks in line with the College Policy on marking group work based on evidence of work carried out within the project. Each student has to show evidence of completing each of the requirements mentioned in the *Project Deliverables* section to receive marks for that particular requirement. Each student will have to complete a peer review form regarding the tasks contributed to the project by each team member. Note that the peer review forms are not marked, but they will be used to inform the marking when awarding the individual marks for the work carried out within the project.

There will be a final joint project presentation and demonstration for the *Team Project* module and the *Software Quality and Testing* module. At that time, students will present both their finished application as part of the assessment for the *Team Project* module, and the test strategy implemented to practically test their application together with running all the automated tests as part of the assessment for the *Software Quality and Testing* module.

2.2. For students who DO not take the Team Project module – this assessment does not involve Cross Module Assessment

For the project assessment of the Software Quality and Testing module, students, who are NOT taking the *Team Project* module this semester, will have to test an application that they develop. Consequently, this is an individual project.

There will be a final project presentation and demonstration for the *Software Quality and Testing* module. At that time, students will present both their finished application and the test strategy implemented to practically test their application together with running all the automated tests as part of the assessment for the *Software Quality and Testing* module.

3. Project Deliverables

You are required to submit all the project deliverables through Moodle via the dedicated submission pages.
You are required to document the entire testing strategy you have adopted to practically test your application through the following deliverables:

1. A **project report** which should include:
    - Headline – title of the report, module name, student name and student number OR in the case of cross module assessment the names and student numbers of the team members
    - Section 1: Introduction – a brief description of the software application to be tested and the main objectives of the project
    - Section 2: A test plan
    - Section 3: Test cases designed using black-box testing techniques (use at least two techniques covered within our module)
    - Section 4: Test cases designed using white-box testing techniques
    - Section 5: Automated testing (e.g., unit testing)
        i. Describe the automated tests (e.g., unit tests) you implemented and the functionalities tested.
        ii. Include in the report code snippets for all the unit tests you have implemented.
        iii. Execute the unit tests, and document the results of your tests, namely provide in your report screenshots that includes a summary of the execution of the unit tests (i.e., for each unit test, the name of the test method run and whether the test has passed or failed).
    - Section 6: Conclusions – what did you learn and find out?
    - Appendix: The project report should include as an appendix the requirements specification (i.e., functional and non-functional requirements) of the application

    Note: in the case of the students who take the cross-module assessment (i.e., students who works in teams) it has to be clearly stated in the report under each of the Section 3, Section 4 and Section 5 who has done what.
    This is a submission requirement
    Submit the report document via the submission page *Project Submission (Report part/tab)* available on the Software Quality and Testing Moodle page. **This is a submission requirement.**

2. The **source code** for all automated tests
    Submit the complete project that contains the source code of all the unit tests you have implemented as a .zip file via the submission page *Project Submission (Source Code part/tab)* available on the Software Quality and Testing Moodle page. **This is a submission requirement.**

3. Project **presentation** and **demonstration**
    Submit the Project presentation slide deck via the submission page *Project Submission (Presentation Slide Deck part/tab)* available on the Software Quality and Testing Moodle page. **This is a submission requirement.**

4. Assessment Criteria

The Project will be assessed based on the assessment criteria shown in *Table 1* and marking rubric shown in *Table 2.*

Table 1 Assessment Criteria

Test Plan	20%
Test cases designed using black-box testing techniques	20%
Test cases designed using white-box testing techniques	20%
Automated Testing (e.g., unit tests)	35%
Project Presentation & Demonstration	5%

Table 2 Marking Rubric

Grade Criterion	H1 (> 70%)	60-69%	50-59%	40-49%	Fail (< 40%)
Test Plan: 20%	Excellent/very good test plan in which all the required elements are included and are thoroughly documented.	Good test plan in which all the required elements are included.	Good test plan in which most of the required elements are included.	Weak test plan that provides a limited understanding of the testing to be carried out.	Very limited and poor or inexistent test plan.
Test cases designed using black-box testing techniques: 20%	Excellent/very good application of suitable black-box testing techniques in designing substantive and relevant test cases. Excellent/very good documentation of the test cases.	Good application of black-box testing techniques in designing relevant test cases. Good documentation of the test cases.	Adequate application of black-box testing techniques in designing relevant test cases. Adequate documentation of the test cases.	Weak/limited application of at least one black-box testing techniques in designing test cases. Weak documentation of the test cases.	Poor application of black-box testing techniques in designing test cases. Very weak/poor documentation of the test cases.
Test cases designed using white-box testing techniques: 20%	Excellent/very good application of suitable white-box testing techniques in designing substantive and relevant test cases. Excellent/very good documentation of the test cases.	Good application of white-box testing techniques in designing relevant test cases. Good documentation of the test cases.	Adequate application of white-box testing techniques in designing relevant test cases. Adequate documentation of the test cases.	Weak/limited application of white-box testing techniques in designing test cases. Weak documentation of the test cases.	Poor application of white-box testing techniques in designing test cases. Very weak/ poor documentation of the test cases.
Automated Testing (e.g., unit tests): 35%	Excellent/very good development of substantive and relevant unit tests. Excellent/very good documentation of the unit tests and their execution.	Good development of relevant unit tests. Good documentation of the unit tests and their execution.	Adequate development of relevant unit tests. Adequate documentation of the unit tests and their execution.	Weak/limited development of unit tests. Weak documentation of the unit tests and their execution.	Poor development of unit tests. Very weak/ poor documentation of the unit tests and their execution.
Project Presentation & Demonstration: 5%	Excellent well directed presentation and demonstration with impeccable handling of questions.	Clear presentation and demonstration with good handling of questions.	Neat oral presentation and demonstration and acceptable handling of questions.	Poor oral presentation and demonstration and weak handling of questions.	Unacceptable oral presentation and demonstration and poor handling of questions.