Cardiff School of Computer Science and Informatics

Coursework Assessment Pro-forma

Module Code: CM1202
Module Title: Developing Quality Software
Lecturer: Helen Phillips
Assessment Title: Implementation and Testing
Assessment Number: Coursework 3
Date Set: Tuesday 12th February 2019

Submission Date and Time:
Draft Test case - via email to phillipsHR@cardiff.ac.uk by 9:30am on Friday 8th March 2019.

Final Submission - Get one team member to submit the code, and completed test cases to Learning Central by 9.30 am on Thursday 4th April 2019. ALL team members should demonstrate their part of the prototype in your allocated session on Thursday 4th April 2019.

Return Date: Feedback on your Draft's will be provided via learning central (email) by Monday 18th March 2019. Feedback on your coursework will address the above criteria. Work will be returned along with written feedback by the end of week 12.

This assignment is worth 30 % of the total marks available for this module. The penalty for late or non-submission is an award of zero marks.

Your submission must include the official Coursework Submission Cover sheet, which can be found here: https://docs.cs.cf.ac.uk/downloads/coursework/Coversheet.pdf

Submission Instructions

- **Draft Test Case**, - Each team are invited to submit a draft test case for feedback. Consider carefully which test case to submit, remembering that feedback from this draft should be used to improve and enhance the other test cases. This test case should be submitted via email to phillipsHR@cardiff.ac.uk by 9:30am on Friday 8th March 2019.

- **Final Submission**
  Finish your implementation and upload all of your team's .py or .Java files to Learning Central by 9:30am on Thursday 4th April 2019
  Your team should also upload all of your test cases complete with their test results in a **single** document to Learning Central by 9:30am on Thursday 4th April 2019.
  Make sure your prototype can be accessed and will run successfully in the PC labs before the demonstration Thursday 4th April 2019.

- **Individual Submission** – You must submit a self-review plus peer reviews for all other members of your team, via Learning Central by 9:30am on Thursday 4th April 2019. These should be combined into a single pdf document. These will be used to monitor engagement within the team, and if necessary adjust marks. Please note these reviews will be shared with the rest of your team along with feedback on the team coursework.

- The Individual Submission **must** include your student ID on the first page of the document.

Description		Type	Name
Code from all team members	**Compulsory**	,pdf, .py or .java files	[filename].java or [filename].pdf or [filename].py
Set of Test Cases – (Containing one test case for each active team member)	**Compulsory**	One .docx or .pdf file	tests_team[team number].docx or tests_team[team number].pdf
Self-Review, and peer reviews for each team member	**Compulsory**	One .pdf	Student_ID&Team.pdf

Any deviation from the submission instructions above (including the number and types of files submitted) may result in a mark of zero for the assessment or question part.
If you have any problems with submission on Learning Central then email Helen Phillips (PhillipsHR@cardiff.ac.uk) before the deadline.

Assignment

Task

Your team needs to develop a ***working*** prototype system in Python or Java and a set of test cases to validate that your system has met these main requirements.

Main Requirements
It is essential that your team can demonstrate working functionality that meets the ***main requirements*** for the initial scenario.

All Teams

Task 1 - The prototype must:

- **Enable student to take a summative assessment.**
  (The basic test should be multiple choice, with student limited to a single attempt. Following the test students must be able to view feedback, and marks but the correct answers must only be available after the assessment deadline.

- **The lecturer can view student performance on summative assessments.** For example; summary of marks for all students, performance for a specific student.

- **Lecturer must be able to create an assessment**
  The lecturer can select whether the assessment will be formative or summative. Assign a test duration, plus start and end dates if appropriate.

- **Enable student to take a formative assessment**.
  Students must be provided with immediate feedback after each attempt and the correct answers following their final attempt. (more than one attempt)

Most teams will also be expected to provide extra functionality to extend the prototype. This will depend on team size:

- Teams of five or more: **The lecturer can view statistics about cohort's performance on formative assessments.** For example; percentage of time a question is answered correctly, the question most often answered incorrectly, and number of attempts at a test.

- Teams of six: **Lecturer must be able to modify the formative or summative assessments.**

Task 2 – Test Cases
You also need to develop a test case for EACH of the requirements stated above to validate that your prototype meets each of these main requirements. (One Test case per team member)

Implementation of Prototype

Teams need to make sure that EACH team member is assigned specific responsibility for developing code for a particular part of the prototype (see advice below). *However, if a team member does not complete their part of the implementation we do not expect other team members to take over their task.*

Each team member should also provide **three** examples in their implementation that demonstrates that they know how to produce quality code that enhances different quality criteria (e.g. usability, reliability, maintainability). This will be discussed in your individual reports. Note credit will be given for fully tested basic features over impressive but limited tested features.

The team should ensure their application has a consistent look and feel. Don't forget to get one member of your team to upload your .py or .java files on Learning Central by 9:30am on Thursday 4th April 2019.

Developing Test Cases

Your team needs to create a set of test cases that a user can follow to validate that your prototype meets the Main Requirements. EACH member of the team should be assigned responsibility for developing a test case for ONE of these main requirements. You need to ensure each member covers a different requirement so that you will be able to validate all of the requirements stated for the size of your team.

Each requirement stated in the Task section will need a separate test case with a clear procedure that can be followed by a user to carry out the essential steps for the ***basic flow*** and a clear indication of the outputs that your prototype should give in response to the user's actions (alternative flows do not need to be tested). Each test case should include appropriate test data.

We will need to see your interfaces when checking and marking the test cases. The interfaces will help determine if you have sufficient information in your test cases to cover the essential steps of the basic flow for each requirement. You can either provide a drawing of the design of your interface (neat hand-drawn sketches are acceptable) or a screen shot if the interface is complete. Please ensure that all information presented in the interface is readable.

Each test case should be presented **using the test case template** which will be available on Learning Central. Once your prototype is complete you should use your test cases to validate your prototype. You need to provide test results to clearly show what steps in the basic flow passed or failed. It is good practice to get someone other than the author of the test case to carry out the testing and fill in the test results.

Don't forget to get one of your team members to upload your completed test cases with the test results and the interfaces on Learning Central by 9:30am on Thursday 4th April 2019.

Demonstration of Prototype

In your demonstration your team will need to clearly show that the functionality of your prototype meets all the main requirements appropriate to the size of your team.

Each team member will need to demonstrate ***their own code*** by running through the range of functionality and highlighting any extra features or interesting functionality that they have successfully implemented.

If team members have worked together on a requirement with more complex functionality then they can demonstrate their code together but they must clearly state what each person has done.

Weightings

Task	**/100**
1. Prototype	80
2. Test cases	20

Some important advice:

Your team will need to split up the system into appropriate modules which can be allocated to team members to develop. Requirements that need more complicated functionality or complex interfaces can be split into several modules so these can be developed by two or more team members simultaneously. However, each team member is responsible for developing and testing the code for their own modules.

Make sure your team manages the dependencies. If core functionality is required which is needed by other parts of the prototype ensure this is developed first. If several modules are needed to deliver functionality ensure each team member is clear what they are developing and how this relates to the work of others.

You need to demonstrate ***working software***. We recommend that you develop your code in small chunks and test this frequently, so you don't have too much code to check through if it stops working. Consider using version control, for example Git to manage your working code as you go along. You can always revert to the latest backup if you really make a mess of the code you are working on. Team members should frequently integrate any new working code into the system. This way if there are problems there is not too much code to check and you can make sure you have correctly managed your dependencies. Make sure all team members have copies of the current working system. Don't leave the integration to the end and hope it will work.

Don't be too ambitious. You do not have to include all the functionality described in the use case descriptions that your team provided for your design presentation. Focus on main requirements as stated in this assignment brief. For real projects clients will expect to see functionality working before they will agree that a requirement is met. You therefore need to ensure all essential functionality is working before trying to exceed the requirements. Note credit will be given for fully tested basic features over impressive but limited tested features.

Learning Outcomes Assessed

- Understand the importance of basic Software Engineering concepts, principles and practices
- Show an understanding of how to plan and manage a project through the effective use of a variety of tools and techniques
- Develop a set of test cases to demonstrate how the system can be validated
- Create a prototype system to demonstrate how the main requirements can be implemented
- Professionally record and document the results of Software Engineering development work

Criteria for assessment - Credit will be awarded against the following criteria.

Prototype:

- Comprehensiveness of prototype in covering the basic steps for all of the main requirements
- Fully tested functionality in meeting the main requirements
- Appreciation of Quality criteria
- Consistency of the look and feel of your prototype

Test Cases:

- Clarity of Instructions including test data
- Relevance of steps in meeting the requirements

Feedback and suggestion for future learning

Feedback on your Draft’s will be provided via learning central (email) by Monday 18th March.

Feedback on your coursework will address the above criteria. Work will be returned along with written feedback by the end of week 12.

Feedback from this assignment will be useful for the second part of this module and for the Second year group project.