**SCHOOL OF COMPUTER SCIENCE & INFORMATICS**

**COURSEWORK ASSESSMENT PROFORMA**

**MODULE:** CM1202 Developing Quality Software

**DATE SET:** Monday 7th March 2016

**SUBMISSION DATES:** Get one team member tosubmit the code, completed test cases and an updated Gantt Chart to Learning Central by 9.00am on Friday 15th April. ALL team members should demonstrate their part of the prototype in your Friday Lab Class on Friday 15th April 2016.

**SUBMISSION ARRANGEMENTS:**

1. Bring your draft test cases and interface designs to the lecture at 2.10pm on Tuesday 15th March.
2. Finish your implementation and upload all of your team’s .py files to Learning Central by 9:00am on Friday 15th April.
3. Your team should also upload all of your test cases complete with their test results in a **single** document to Learning Central by 9:00am on Friday 15th April.
4. Upload an updated version of your Gantt Chart to Learning Central by 9:00am on Friday 15th April.
5. Make sure your prototype can be accessed and will run successfully in the PC labs before the demonstration in your Friday Lab Class on 15th April.
6. You will be required to sign an official coursework submission cover sheet with the other team members who have contributed during the demonstration.

One member of the team should upload the following files to Learning Central by 9:00am on Friday 15th April:

|  |  |  |  |
| --- | --- | --- | --- |
| Description | | Type | Name |
| Code from all team members | **Compulsory** | ,pdf files or .py files | [filename].py or [filename].pdf |
| Test Cases – one for each team member | **Compulsory** | One .docx or .pdf file | tests\_team[team number].docx or  tests\_team[team number].pdf |
| Gantt Chart | **Compulsory** | One .docx or .pdf file | plan\_team[team number].docx or  plan\_team[team number].pdf |

If you have any problems with submission on Learning Central then email Helen Phillips ([PhillipsHR@cardiff.ac.uk](mailto:PhillipsHR@cardiff.ac.uk)) at least half an hour before the deadline.

**TITLE:** Implementation and Test

This coursework is worth 30% of the total marks available for this module. The penalty for late or non-submission is an award of zero marks. You are reminded of the need to comply with Cardiff University’s Student Guide to Academic Integrity.

**LEARNING OUTCOMES ADDRESSED:**

* 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

**Task**

Your team needs to develop a ***working*** prototype system in Python 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 functionality that meets the ***main requirements*** for the initial scenario. Therefore, your team should demonstrate that the prototype can provide a suitable basic flow for the following (alternative flows do not need to be demonstrated):

All Teams

1. **Provide a short lesson on a topic from COMSC’s autumn semester year 1 modules.**
2. **Provide a short test for each of the chosen topics, which displays the result of the test to the student and stores the result once the test is completed.** The test should have appropriate instructions and give feedback to the student.
3. **Provide the facility for storing and retrieving students’ results.** The Lecturer should be able to retrieve the results of a student at a later point.
4. **Provide ONE interesting extension to your functionality** as part of your main requirements. *This should be clearly stated in your demonstration.*

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

1. Teams of five: **Provide a second short lesson and test on a different topic from COMSC’s autumn semester year 1.**
2. Teams of six: **Provide a second interesting extensions to your functionality** as part of your main requirements. *These should be clearly stated in your demonstration.*

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.

**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.

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 files on Learning Central by 9:00am on Friday 15th April.

**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.

There will be an opportunity in the lecture at 2.10pm on Tuesday 15th March to get your test cases checked and get feedback on how they can be improved. 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 a 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. Make sure to include a screen shot of the interface that has been tested with your test case.

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 9:00am on Friday 15th April.

**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.

**Gantt Chart**

The updated Gantt Chart will not be assessed but will be used to monitor the workload of the groups. We expect you to have added detail to the implementation and test sections of the Chart. This should be uploaded on Learning Central by 9:00am on Friday 15th April.

**Weightings**

|  |  |
| --- | --- |
| **Task** |  |
| 1. Prototype | 80% |
| 1. 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. Frequently backup 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.

**Assessment Criteria**

Your coursework will assessed on the following:

Prototype:

* Comprehensiveness of prototype in covering the basic steps for all of the main requirements
* Higher marks can be achieved for impressive functionality in meeting the main requirements
* Consistency of the look and feel of your prototype

Test Cases:

* Clarity of Instructions
* Relevance of steps in meeting the requirements
* Validation of the test cases

**Feedback and suggestions for improvement**

Feedback on your coursework will address the above criteria. Initial verbal feedback on test cases will be given to the teams in week 8. Work will be returned along with written feedback by week 11.