# Cardiff School of Computer Science and Informatics

## Coursework Assessment Pro-forma

**Module Code**: CMT303

**Module Title**: Software Engineering

**Lecturer**: Helen Phillips

**Assessment Title**: Team Portfolio Part B

**Assessment Number**: 2

**Date Set**: Thursday 30th January 2020

**Submission Date and Time**: *Thursday 23rd April* 2020 at 9:30am.

**Return Date**: Week 2 Spring Exam Semester via Learning Central

This assignment is worth 30% of the total marks available for this module. If coursework is submitted late (and where there are no extenuating circumstances):

1 If the assessment is submitted no later than 24 hours after the deadline, the mark for the assessment will be capped at the minimum pass mark;

2 If the assessment is submitted more than 24 hours after the deadline, a mark of 0 will be given for the assessment.

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

Video – Walk-through demo of software developed

You have been provided with the application Kazam on your COMSC laptops on which to record your video. During contact sessions for this module you will be given opportunities to do trial recordings using the Kazam App please make use of these.

Prior to handing in make sure all documentation has been collected, including a coursework coversheet from each Team member. Please ensure the student number of each team member who **contributed** is in the footer. A nominated team member shouldsubmit your coursework, via Learning Central by 9:30am on 23rd April 2020 . (Use your team number as part of the name of the files). If you have any difficulties submitting via Learning Central you MUST e-mail the module leader **Helen Phillips** ([PhillipsHR@cardiff.ac.uk](mailto:PhillipsHR@cardiff.ac.uk)) at least half an hour before the deadline time.

|  |  |  |  |
| --- | --- | --- | --- |
| Description | | Type | Name |
| Team Portfolio Part B, Should including coursework coversheets, Test cases, with interface diagrams, Report on the implementation process details below. | **Compulsory** | One .pdf file | [team number]portfolio\_partB.pdf |
| Video – Walk-through demo of software developed using Kazam. | **Compulsory** | MP4 video of approx. 15 minutes | [teamnumber]demoVideo.mp4 |

Any deviation from the submission instructions above (including the number and types of files submitted) will result in a mark of zero for the assessment or question part.

Staff reserve the right to invite students to a meeting to discuss coursework submissions

## Assignment

## The team portfolio Part B will be made up of the work undertaken by the team during the module in the Spring Semester. You will be given specific pointers as to which section(s) to work on until the next contact session. There will be specified opportunities for teams to obtain formative feedback on draft versions of most elements of the portfolio, either from the module lecturer or via peer assessment.

**INTRODUCTION**

CMT303: Software Engineering is assessed through coursework that is developed as part of a software development project carried out in teams of five or six members. You will need to complete team tasks and individual tasks as part of the project. This assessment covers the Second, (Implementation) part of the project which will be carried out in the Spring Semester.

The module leader will usually arrange progress meetings with each team in weeks 6 and 9 of the Semester. These progress meetings will be an opportunity for the team to review how well they are working as a team. Team members will also have an opportunity to raise any questions or concerns they may have about the assessment tasks. Your team should also meet regularly outside the scheduled meetings without the module leader.

**Non-participation of Team Members**

Your team will normally share the marks for this portfolio component equally. If your team believes that someone is not contributing then you should email the module leader **Helen Phillips** (PhillipsHR@cardiff.ac.uk) as soon as possible so that I can investigate further. It is also important that anyone who is having difficulty contacting their team or has any other issues that are affecting their ability to work with the team informs me as soon as possible.

**Some important advice:**

Your team will need to split up the system into appropriate features / modules which can be allocated to team members to develop. Requirements that need more complicated functionality or complex interfaces can be split. However, each team member should take responsible for the development and testing of at least one feature. 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. 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. It is a good idea to 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. This way if there are problems, there is not too much code to check. Make sure all team members have copies of the current working system. Team members should frequently integrate any new working code. You should consider investigating version control software. (Git Lab is available on the machines in COMSC.) Don’t leave integration to the end hoping it will work.

**Team Portfolio Deliverables**

You Team portfolio Part B should include

* Test cases, including any necessary interface sketches
* Short report (max 1,000 words) on the implementation process including discussions of the
  + techniques and development methodologies used in planning and managing the project
  + strategies and methods used to ensure good quality software.

Task 1 - Test Cases

Hand in Test cases, with interface sketches, for 6 of the main requirements (One by each team member). You can refer to the Feedback sheet attached for the main requirements). Each test case should be presented using the test case template supplied in the lecture notes:

* Create a set of test cases that a user can follow to validate that your prototype meets all the **main** requirements for the initial scenario. Each requirement 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 output**s 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 which retains consistency across your set of tests.
* Interface sketches: You will also need to provide Interface drawings that the user will use to help them carry out the tests. These will be used to help determine if you have provided sufficient information in your test cases to cover the essential steps for each requirement.

30%

Task 2 - Developing the prototype

In the portfolio

* Explain and justify the techniques and development methodologies your team used to plan and manage the implement of the prototype. This section should include charts, or boards appropriate to the methodology used.
* Discuss the strategies and methods used to aid the team’s production of good quality software. Use examples from your implementation to demonstrate that the team knows how to produce quality code that enhances different quality criteria (e.g. usability, reliability, maintainability).

35%

Video – Walk through demonstration of software –

* Your team needs to develop a working prototype system. It is essential that your team can demonstrate functionality that meets the main requirements for the initial scenario.
* Using Kazam make a 15 minute video that demonstrates that runs through the working functionality of the software and highlighting any extra features or interesting functionality that they have successfully implemented.
* Prior to doing the final video recording, teams should develop a script and refer to the ‘Video – Walk through demonstration of software’ criteria in the feedback sheet below.
* Teams will be given the opportunity to test Kazam and do a practice video recording during the contact sessions.

35%

|  |  |
| --- | --- |
| **Portfolio Element** | **Contribution 100%** |
| Test cases | 30% |
| In the portfolio (project management, development methodologies and team strategy for enhancing Quality Criteria) | 35% |
| At the Demonstration (Prototype) | 35% |

## Learning Outcomes Assessed

* Utilise their knowledge and understanding of the different phases in the software lifecycle.
* Demonstrate an understanding of issues involved in creating and managing a project plan, including a consideration of activities relevant to the management of risk and quality.
* Demonstrate comprehension of the characteristics of software quality, and the impact of quality requirements on project planning.

# Criteria for assessment

An outline of the criteria of assessment of each section of the portfolio is given below.

1. Test Cases

Each test case will be assessed against the following criteria:

Clarity of Test Data:

|  |  |
| --- | --- |
| **Excellent** | All steps in the test case clearly provide the information required for a user to carry out a procedure using appropriate input test data with a clear indication of the outputs that the user should expect to see in response to their actions. |
| **Good** | Most steps in the test case provide appropriate information for a user to carry out a procedure using appropriate input test data with an indication of the outputs that the user should expect to see in response to their actions. |
| **Adequate** | The test case gives the user a reasonable indication of how to carry out the test and gives some indication of the expected outputs. |
| **Poor** | Gives the user little or no indication of how to carry out test or what outputs to expect. |

Clarity of step in basic flow

|  |  |
| --- | --- |
| **Excellent** | The test case covers all essential steps in the basic flow that are relevant for validating whether the prototype has met the main requirement. |
| **Good** | The test case covers most essential steps in the basic flow that are relevant for validating whether the prototype has met the main requirement. |
| **Adequate** | The test case provides some reasonable steps for validating whether the prototype has met the main requirement |
| **Poor** | Steps provided (if any) give little confidence that the main requirement can be validated. |

|  |  |  |
| --- | --- | --- |
|  | Clarity of Test Data | Clarity of steps in basic flow |
| Test Case 1 |  |  |
| Test Case 2 |  |  |
| Test Case 3 |  |  |
| Test Case 4 |  |  |
| Test Case 5 |  |  |
| Test Case 6 |  |  |

Consistency of the Set of Test Cases

Test Case Descriptions:

Excellent: All test cases used the same appropriate test case template

Good: Most test cases used a similar test case template.

Adequate: At least half of the test cases were reasonably similar.

Poor: Most of the test cases were inconsistent

/30

Developing the prototype – In the Portfolio

* Explanation and justification of the techniques and development methodologies used to plan and manage the implement of the prototype.

No explanation included No project planning or management evident

|  |  |
| --- | --- |
| **Excellent** | * The portfolio contains a clear and consistent explanation of the techniques and methodologies used by the team when implementing the prototype. * Explanation, diagrams and charts demonstrate a well managed project, highly appropriate to the selected development methodology. * Project broken down into clear and appropriate activities, with fair and appropriate consideration of workload. |
| **Good** | * The portfolio contains a mostly clear explanation of the techniques and methodologies used by the team when implementing the prototype. * Explanation, diagrams and charts demonstrate planning and project management, appropriate to the development methodology. * Project broken down into clear activities, with mostly fair and appropriate consideration of workload. |
| **Adequate** | * The portfolio contains an explanation of the techniques and methodologies used by the team, however this explanation is difficult to follow and could have been structured better. * Explanation, diagrams and charts demonstrate some attempt to plan and manage the project however relevance to the development methodology needs more consideration. * An attempt to break the project down into workable activities, with some consideration of workload. |
| **Poor** | * The portfolio contains only a very brief explanation of the techniques and methodologies used by the team. * Brief explanation, diagrams and charts demonstrate limited planning and project management, Limited relevance to the development methodology used. * How the project is break down into activities is unclear, with limited consideration of workload. |

# Quality Criteria - Explanation of the team strategies used to enhance the quality of the prototype

No explanation included

|  |  |
| --- | --- |
| **Excellent** | * The portfolio contains a **clear and consist explanation** of the strategy used to enhance the quality of the prototype. The report demonstrats that the team were **proactive** in considering the different software quality criteria. * Provided **insightful** examples of features that demonstrate quality code and **clearly** explained how quality is enhanced by these features for **four** different quality criteria |
| **Good** | * The portfolio contains a mostly clear explanation of the strategy used to enhance the qulaity of the prototype. The report demonstrates that the team considered some of the different software quality criteria. * Discussed some interesting and relevant examples of features that demonstrate quality code and explained how quality is enhanced by these features for at least three different quality criteria |
| **Adequate** | * The portfolio contains an adequate explanation of the teams considenation of a few of the different software quality criteria. * A few examples of features discussed were relevant to providing quality code for at least two different quality criteria |
| **Poor** | * The portfolio contains only a very brief explanation of where software quality criteria was considered. * Failed to provide many relevant examples of features that could address the different quality criteria |

/30

**Video – Walk-through demo of software developed**

|  |  |
| --- | --- |
| **Excellent** | The implemented system exceeds this requirement. It meets the criteria for a good demonstration and has provided interesting and relevant extra features or impressive functionality for this requirement. |
| **Good** | The implemented system clearly satisfies this requirement by covering all essential steps for a suitable basic flow in a logical order and providing a reasonable interface for the task performed |
| **Adequate** | The implemented system shows reasonable progress towards satisfying this requirement. However, a bit more work is needed in one or more of the following: to include some missing steps in the basic flow, correct the logic of the flow, correct some errors in the task or improve the interface. |
| **Poor** | The implemented system does not satisfy this requirement and will require a significant amount of work to address one or more of the following: to include missing steps in the basic flow, correct the logic of the flow, correct errors in the task or improve the interface. |

The package should

* Enable visitors to take part in a quiz. (The quiz should be multiple choice)

Excellent: Good: Adequate: Poor:

* Enable visitors to take part in a second quiz, containing questions of a different format/style to the first. (these can be picture questions)

Excellent: Good: Adequate: Poor:

* Following the entry of an answer, the visitor must receive immediate feedback on their answer.

Excellent: Good: Adequate: Poor:

* Each visitor can select to restart the quiz and if the visitor completed the entire quiz they should at the end be presented with a summary of their performance.

Excellent: Good: Adequate: Poor:

* The COMSC engagement team need the ability to add new questions.

Excellent: Good: Adequate: Poor:

* The COMSC engagement team need the ability to amend or delete current questions.

Excellent: Good: Adequate: Poor:

* The COMSC engagement team are able to view statistics about visitors overall performance on the quiz. (For example; percentage of time a question is answered correctly, the question most often answered incorrectly, and percentage of people who gave up at that question.)

Excellent: Good: Adequate: Poor:

Plus extra features

These are examples

* + - Questions are displayed in a random order

Excellent: Good: Adequate: Poor:

* + - The Engagement team can select a particular topic for the quiz

Excellent: Good: Adequate: Poor:

* + - The Engagement team can enter information about the schools attending an event

Excellent: Good: Adequate: Poor:

* The Engagement team can download statistics for a particular event or school.

Excellent: Good: Adequate: Poor:

Consistency of the System:

Excellent: All interfaces had a consistent look and feel.

Good: Most of the interfaces had a similar look and feel. .

Adequate: At least half of of the interfaces were reasonably similar.

Poor: Most of the interfaces were inconsistent

# Feedback and suggestion for future learning

Feedback on your coursework will address the above criteria. Feedback and marks will be returned by **21st May 2020** via Learning Central as an attachment, there will be opportunities for additional feedback on request.

Feedback from this assignment will be useful for your Dissertation.