Cardiff School of Computer Science and Informatics

Coursework Assessment Pro-forma

Module Code: CM1202
Module Title: Developing Quality Software
Lecturer: Helen Phillips
Assessment Title: Requirements
Assessment Number: Coursework 1
Date Set: Monday 5th November 2018
Submission Date and Time: Wednesday 3rd December 2018 at 9:30am. Teams have the opportunity to hand in a draft version on Monday Week 8, 19th November via email.

Return Date: Feedback to Teams in Week 12 – Week starting Monday 7th January 2019. Feedback on the draft will be in tutorial sessions.

This assignment is worth 15 % 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 copy**
  Email to PhillipsHR@cardiff.ac.uk by 9:30am on Monday 19th November

- **Final copy** –
  - **Team Submission** - A nominated TEAM member should submit your coursework, electronically, **as a single pdf document**, via Learning Central by 9:30am on Wednesday 3rd December 2018. If you have any difficulties submitting via Learning Central you MUST e-mail the module leader **Helen Phillips** (PhillipsHR@cardiff.ac.uk) before the deadline.

  - **Individual Submission** – You must submit a self-review plus peer reviews for all other members of your team, **via Learning Central - Team File Exchange** by 9:30am on Wednesday 3rd December 2018. 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.
  - The Individual Submission, via Learning Central - Team File Exchange (**must** include your student ID on the first page of the document.

	Description		Type	Name
Team Submission	Requirements document including coursework coversheet	**Compulsory**	One .pdf file	[team number]requirements.pdf
Individual Submission **Team File**	Self-Review, and peer reviews for each team member	**Compulsory**	One PDF	Student_ID&Team.pdf

Exchange				

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.

Assignment

On the bottom of each page of the team portfolio type the name of the members of the team that have contributed to that particular section (this will be used to monitor individual student's engagement with the team).

INSTRUCTIONS:
CM1202 Developing Quality Software is assessed through coursework that is developed as part of developing a piece of software 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 first part of the project which will be carried out in the Autumn Semester.

Non-participation of Team Members

If your team believes that someone is not contributing then you should email, me, the module leader **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 inform me as soon as possible. The peer and self-reviews will be used by the module leader to monitor student engagement.

Deliverables This coursework should consist of the following components:

- A List of Functional User Requirements
- A use case diagram of the system
- An outline use case description or User story/Epic for each use case.
- A self-review and peer reviews for each of the other members of your team.

More detail on these deliverables is given below. In general, do not worry about how to implement the features that you select, however, try to be realistic in your goals.

1. List of Functional User Requirements

Develop a list of **functional requirements** with brief descriptions indicating the features you would like to provide in your application (see scenario). Requirements should be listed using the **MoSCoW notation** and be written so

that they can be **validated** / **acceptance tested.** [these brief descriptions will most likely develop into your use cases]

2. Use Case Diagram

Develop a Use Case diagram to capture the major functional requirements of the ` Educational Software Test System' from the client/users' point of view. Include in the use case diagram any possible relationships between the use cases (namely*, include, and extend*).

3. Use Case Description or User Story

For each use case identified, write a use case description or user story. **(Each team member should write ONE each).** Don't forget to include the associated non-functional requirements (attributes, constraints or characteristics). These should be written as testable acceptance criteria.

Weightings

	/100
1. List of Functional user requirements with brief descriptions (MoSCoW notation and write so they can be validated / acceptance tested)	25
2. Use Case Diagram	25
3 Use Case Descriptions or User Story with non-functional requirements and acceptance criteria	50

Learning Outcomes Assessed

- Understand the importance of basic Software Engineering concepts, principles and practices.
- Gain an appreciation of how main stages in the software development lifecycle contribute to the development of a high-quality software system by performing key technical tasks from each stage of the project
- Design a system using appropriate UML models to demonstrate how the main requirements can be delivered

---

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

List of Functional Requirements with priority

List of functional requirements not done

Appropriate:	Excellent	Good	Pass	Fail
Explanation	Excellent	Good	Pass	Fail

Priority: Excellent Good Pass Fail

User - focused: Excellent Good Pass Fail

Rating	Appropriate	Explanation	Priority	User focused
Excellent	All-important functionality have been identified	Functionality identified have been clearly and concisely explained and expressed well so that validation is obvious	Appropriate and proficient use of MoSCoW to identify the priority of the requirement	Highly user focused demonstrating a comprehensive awareness of user/client needs
Good	Most of the important functionality have been identified	Features identified have been explained well and expressed with reference to validation	A good attempt at using MoSCoW to identify the priority of the requirements	Most elements are written from a user perspective. Demonstrating an understanding of user/client needs
Pass	Some of the important functionality have been identified	Adequate explaination of the features identified	Some attempt made to identify the priority of the requirements, but MoSCoW not used	Some attempt to be user focused but too much technical detail in places. Use cases should concentrate on ‘what’ not ‘how’
Fail	Very little functionality have been identified	A poor explanation of the features identified	No attempt to identify the priority of the requirements.	Too much technical detail for this stage of development. Concentrate on what the user/client wants

Use Case Diagram

HINTS

- Actors identified (clear name, unique role in the system, each actor involved with at least one use case)
- Use cases identified (unique intuitive name, justified behaviour that maps to functional requirements)

Top Level Use Case Diagram not done

Relevance:	Excellent	Good	Pass	Fail
Modelling conventions	Excellent	Good	Pass	Fail
Completeness:	Excellent	Good	Pass	Fail
User - focused:	Excellent	Good	Pass	Fail

Rating	Relevance	Modelling Conventions	Completeness	User focused
Excellent	The model is highly relevant to the provided client scenario	The model clearly follows UML modelling conventions and has no errors in modelling syntax.	The model clearly meets the requirements of the assignment and no important information is missing.	Highly user focused demonstrating a comprehensive awareness of user/client needs
Good	The model is relevant to the provided client scenario	The model is a good attempt at following UML modelling conventions and contains a few errors in modelling syntax.	The model is a good attempt at meeting the requirements of the assignment and little of the important information is missing.	Most elements are written from a user perspective. Demonstrating an understanding of user/client needs
Pass	The model has attempted to be relevant to the provided client scenario	The model is a reasonable attempt at following UML modelling conventions and contains some errors in modelling syntax.	The model is a reasonable attempt at meeting the requirements of the assignment and some of the important information is missing.	Some attempt to be user focused but too much technical detail in places. Use cases should concentrate on ‘what’ not ‘how’
Fail	The model has little relevance to the provide client scenario	The model does not follow UML modelling conventions and has many errors in modelling syntax.	The model meets few of the requirements of the assignment and much of the important information is missing.	Too much technical detail for this stage of development. Concentrate on what the user/client wants

Use Cases descriptions or User Stories

Relevance:	Excellent	Good	Pass	Fail
Consistency	Excellent	Good	Pass	Fail
attributes / characteristics:	Excellent	Good	Pass	Fail
User - focused:	Excellent	Good	Pass	Fail
Provided descriptions / user stories:	All	Most	Some	Few/None

Rating	Relevance	Consistency	attributes / characteristics	User focused
Excellent	Highly relevant and expressed well so that validation is obvious	Clearly follows appropriate conventions and has no errors in syntax.	Relevant associated attributes, constraints or characteristics have been identified and are clearly testable.	Highly user focused demonstrating a comprehensive awareness of user/client needs
Good	relevant and expressed so that validation is achievable	Good attempt at following conventions but contains a few errors.	Some relevant associated attributes, constraints or characteristics have been identified and are testable.	Most elements are written from a user perspective. Demonstrating an understanding of user/client needs
Pass	Relevant but challenging to validation	Reasonable attempt at following conventions. Contains some errors.	Some relevant associated attributes, constraints or characteristics have been identified and difficult to test / check	Some attempt to be user focused but too much technical detail in places. Use cases should concentrate on ‘what’ not ‘how’
Fail	little relevance to the provide client scenario	Does not follow conventions with many errors and omissions.	Few if any relevant associated attributes, constraints or characteristics have been identified. Not testable	Too much technical detail for this stage of development. Concentrate on what the user/client wants

---

Feedback and suggestion for future learning

Feedback on your coursework will address the above criteria. Feedback and marks will be returned in Week 12, via Learning Central as an attachment, there will be opportunities for additional feedback in team meetings.

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