INSE – Introduction to Software Engineering

Academic Year 2016/2017

Coursework Specification

Unit co-ordinator: Dr Claudia Iacob

Email: iacob@port.ac.uk

Room: Buckingham Building, 1.40

1. Coursework Overview

The coursework for this unit is designed as a group project and it aims to develop a range of career skills, including levels of team-working, planning skills, negotiation skills, initiative skills, leadership skills, project-scheduling skills, and report-writing skills. The coursework will give you initial experience of the systematic construction of a significant piece of software, demonstrating the theory imparted in the unit lectures.

As part of the coursework, you are required to design and develop a medium-size 3-tier software system within an application domain of your choice. You will be working in teams of up to 6 students. As a team, you are free to choose the software process model, programming language(s), tools, and methods to be used along the way. However, you must ensure the following submissions:

Deadline	Submission	Description	Mark	Feedback
14-Oct-2016	Project proposal and project plan	The project proposal identifies the application domain of the system to be developed and describes the motivation and main purpose of the system. The plan is a draft project plan, including roles, milestones, deadlines, risks, estimates, etc.	5%	21-Oct-2016 Formative feedback during tutorial.
21-Nov-2016	System Requirements Specification Document	The SRS document describes the system requirements elicitation process conducted and identifies the requirements for the system to be developed.	20%	16-Dec-2016
16-Jan-2016	Design Documentation	The modelling and mock-up document should include context, interaction, structural, and behavioural models of your system together with mock-ups of the system's UI.	20%	10-Feb-2017
17-Feb-2017	Prototype demo	The initial system prototype submission will consist of a short (3-5 minute) video depicting the current state of the system.	10%	24-Feb-2017 Formative feedback during tutorial.
20-Mar-2017	Testing documentation	The testing documentation includes details on the design, execution, and result of the testing process run on the system developed.	20%	21-Apr-2017
28-Apr-2017	Final report	Your final submission should include: - demo of your final product - 3-page retrospective account of the project: what worked, what didn't, how were the risks met, were the estimates correct, etc. - evidence of system documentation (eg. JavaDoc format) - source code made available on GitHub	25%	26-May-2017

2. Project proposal and plan

The project proposal identifies the application domain of the system to be developed and describes the motivation and main purpose of the system. It should not exceed 3 pages and it should include:

- Introduction providing details on the application domain, motivation, objectives and main constraints of the project. Describe the problem and the proposed solution.
- Project organization describing the way in which the team is organized, the people involved, and their roles in the team. The role in the team would identify your responsibilities in the context of the project.
- Risk analysis providing a list of possible project risks, their likelihood and consequences, and strategies put in place to address them.
- Resource requirements identifying the software and hardware resources needed throughout the project.
- Work breakdown providing details on the breakdown of the project into activities and the milestones and deliverables associated with each activity. Milestones are key stages in the project where progress can be assessed.
- Project schedule illustrates dependencies between the activities, the estimated time required to reach each milestone, and the allocation of team members to activities.
- Monitoring mechanisms describes the strategies put in place to make sure the deadlines are met and the project is on time.

Some acceptable ideas for an application include:

- Service booking application for particular business, eg. restaurant, travel, accommodation.
- Library system
- Calendar application
- Meeting Scheduler

Simple applications that do not include a UI, logic and storage tier are not acceptable. If in doubt, discuss your idea with the tutor during the tutorials.

All project proposals must be approved by your tutor before any work on the project can begin. This will be organized during the tutorials. Expect your project proposal and plan to go through several iterations based on the feedback received during the tutorials. All projects must be ready to start by week 5.

3. System Requirements Specification Document

The SRS document describes the system requirements elicitation process conducted and identifies the requirements for the system to be developed. It should not exceed 5 pages and it should include:

- Preface provides details on the version of the document and its revision history.
- Introduction describes the need for the system, a brief description of the system's functions and how it will work, similar systems and overlapping functionality.
- Glossary defines the technical terms used in the document.
- Method provides details on the method used to elicit the requirements and how you applied this method.

- User requirements definition describes the services provided for the user. At this stage, this description is high level, and provided using natural language and diagrams that are understandable to a non-specialized audience.
- System requirements specification describes the functional and non-functional requirements in details.

Examples of SRS documents available at:

- http://www.lostclouds.com/2Communicate/project/SRS.pdf
- http://www.artemis-emmon.eu/deliverables/FP7-JU-EMMON-2010-DL-WP7-003-D7.1-software-requirements-specification-document.pdf

You should view the SRS document as a contract between yourselves and the client of the system you are developing. Please make sure you address all your users' concerns when writing the SRS document.

4. Design Documentation

The modelling and mock-up document should include context, interaction, structural, and behavioural models of your system together with mock-ups of the system's GUI. It should not exceed 5 pages, and it should include at least:

- Use case modelling of at least 5 main usage scenarios of the system
- Sequence diagram modelling for the use cases described
- System architecture: representation, architectural patterns used
- Mock-ups of the system's UI

The design documentation is usually a working document that may suffer changes along the way depending on the development model employed. However, at this stage in the process, you will be expected to provide complete context, interaction, structural, and behavioural models of your system and drafts of your UI, aka mock-ups.

5. Prototype Video Demo

A first demo of your system should be provided in the form of a short video (3-5 minutes). Videos should:

- provide an overview of the prototype's capabilities
- walk through (some of) the prototype's capabilities
- where appropriate, provide clarifying voice-over and/or annotation highlights

All videos should be made available online. Feedback on the prototype will be provided during the tutorial sessions. Below, some examples of video demos:

- https://www.youtube.com/watch?v=EIyAflgHVpU
- https://www.youtube.com/watch?v=npDqMVP2e9Q
- http://www.cc.gatech.edu/~orso/software/testevol/testevol.html
- https://www.youtube.com/watch?v=icQVS6w0jTE
- https://www.youtube.com/watch?v=_HihwRNeK3I

Formative feedback will be provided on the video demos during the tutorials. The main aim of your video is to demonstrate the capabilities of your system to a large audience. Think of this as an opportunity to convince potential users to make use of your system and, possibly, help you improve it. Your system will evolve after you've made the demo available, so this video does not necessarily need to completely reflect the final version on your system.

6. Testing Documentation

The testing documentation includes the design, execution, and result of the testing process run on the system developed. It should not exceed 10 pages and it should include:

- Details on the testing strategies used during unit, component, and system testing.
- Description of the testing process followed, including information on the testing values, environment used, testing tools, issues encountered, etc.
- Description of the results of the testing process, together with the changes and iterations of the system made as a result of the testing process.
- Description of the usability evaluation process run on the system.

You are free to define your own template for the testing documentation as long as you include all of the above.

7. Final Report

Your final submission should include:

- a video demo (short video version) of your final product. This should not exceed 5 minutes and it should be made available online. The video should:
    - walk through all the capabilities of the system
    - where appropriate, provide clarifying voice-over and/or annotation highlights

- a 3-page retrospective account of the project: what worked, what didn't, how were the risks met, were the estimates correct, etc. The retrospective account should provide examples of lessons learned and guidelines for mitigating problems in a future team project.

- System documentation (eg. JavaDoc). You are required to document your code through comments. Evidence of the use of guidelines for code documentation will be looked for. Ideally, you will provide a JavaDoc format documentation. However, in case of using a programming language that does not support automatic generation of code documentation, the source code will be used as evidence.

- Source code. All source code should be made available on GitHub (publicly or as private repositories), together with a text file providing instructions for installing and running your system. Your tutor should be given access to the repository.

8. Marking Scheme

All work will be marked along the following criteria using academic judgement.

Submission	No	Task	Mark
Project Plan	1	Describe project organization	5%
	2	Produce risk analysis	
	3	Identify resource requirements for the project	
	4	Breakdown work into tasks	
	5	Schedule project	
SRS Document	6	Describe the need for the system and the context of its development	20%
	7	Define technical terms used in the document	
	8	Describe the process (method and its application) used for eliciting requirements	
	9	Provide an overview of the high-level user requirements for the system	
	10	Document the functional (user and system) requirements of the system	
	11	Document 3 types of non-functional requirements of the system	
Design Document	12	Provide use case modelling for at least 5 scenarios	20%
	13	Provide sequence diagram modelling for at least 5 scenarios	
	14	Provide system architecture: representation and architectural patterns used (please provide justification for decisions)	
	15	Mock-ups of the system's UI	
Demo	16	Provide overview of the prototype's capabilities	10%
	17	Walk through the prototype's capabilities	
	18	Provide voice-over/annotation highlights	
Testing Document	19	Provide details on the design of each of the testing phases: unit, component, and system testing	20%
	20	Provide details on the testing process followed: set up, testing values, issues encountered, tools used	
	21	Describe the results of the testing process	
	22	Provide details on the process followed for evaluating the usability of the system	
Final Report	23	Make source code available on GitHub together with running instructions document	25%
	24	Provide a demo of your final product: walkthrough of all capabilities of the system, voice-over/annotations	
	25	Provide evidence of system documentation	
	26	Provide a retrospective account of the project	

All submissions are mandatory. Failing to submit one deliverable will lead to failing the coursework.