# Case Studies in Software Design

**Subtask Three:** Implementation & Evaluation

**Due Date:** Friday 11th March 2016

**Marks Available:** 50 (50% of the overall coursework assessment)

For this final subtask, you are required to implement designs for the alternative case study, test your implementation, and to evaluate the user interface to the system (UI). As with subtask two, you will be presenting your work via a walkthrough shortly after the submission deadline. You will also be expected to demonstrate the testing you have performed, and provide written documentation about the testing and the evaluation that you have carried out.

This subtask will require a coordinated approach and as such, we expect you to plan your time appropriately using an appropriate project management model. While we will not request to see these, you should reflect upon this and time management when it comes to the critical evaluation component.

## Implementation

You will be provided with the documents from another team that have designed a system for the alternative case study to the one that you worked on. You are required to implement the system following these given designs. You should only deviate from the designs if you can present a very good, critical reason for making the change. You will be asked to explain and justify any changes you make from the given designs.

While the case studies include mobile capabilities, you may choose to develop your applications for a desktop using a suitably restricted window size to simulate a mobile screen if you find this more convenient. We encourage the use of Java and if you choose a desktop demonstration, and we recommend that you use the NetBeans IDE, although you are free to use an alternative IDE as long as it is available on a standard campus PC (e.g. Eclipse).

The designs provided will cover the major functionality of the system, so while you may be required to implement functionality not fully covered, you do not need to create additional design documents to support the code for these new areas of functionality. Additionally, there is no need to write complex backends to manage the data, so you should consider using serialisation to store the state of the OO design for the purposes of this assignment, it is perfectly acceptable to assume the data and program will coexist on the device. Similarly, it is not necessary to build a fully distributed version of the system – simulating the behaviours of remote elements of the system will attract full marks. We are looking for quality of software engineering (good coding practice, testing, UI evaluation) over quantity. An overview of the marks for implementation is given below, but please also check with the module team to clarify the scope of your implementation.

While the primary focus of this subtask is the end-product and the evaluation, we will be looking for quality within your engineering practice such as code commenting, use of JavaDoc, careful testing and version control.

## Testing

In addition to implementing the design and producing a working system for the case study, you need to formally test it, and during the walkthrough you will be asked to demonstrate and discuss the testing that you have done. Both unit testing and acceptance testing should be undertaken. You are not required to formally test the entire system, although you should be informally testing components as you build them. As a team, you need to produce JUnit test suites that cover four classes that range in complexity across your system.

For acceptance testing, you need to produce four Test Specifications per team. These should appear on a single top level Test Plan and each Test Specification should have an accompanying Test Descriptor.

## Evaluation and Critical Reflection

Along with the unit testing, you are also to conduct an evaluation of your UI design. For this you will be expected to review user interface evaluation approaches and select one specific approach that is appropriate to the system that you have developed. The choice of approach should be based upon both theoretical ground and also the practical constraints of evaluating your UI. You should report on how the evaluation was conducted, and the results obtained. The evaluations should be conducted with real users who are not members of your assignment team (up to 4 + 2 sides of A4, +2 being pictures and illustrations).

## Critical Reflection

Finally, you are required to produce a critical reflection report, focusing on the ways in which your group’s implementation differs from the original designs (including appropriate diagrams and code fragments to illustrate changes were appropriate) and explain the reasons for the change. You should also reflect upon the time management approach you used to plan the work required for this subtask (up to 2 sides of A4 remember, quality over quantity is being assessed).

## Deliverables

You will present your implementations during a walkthrough where all team members must attend; individuals that do not attend their group walkthrough will be given a mark of zero for this subtask. You must also upload all required code and documents (proof of testing, UI evaluation report and critical reflection) to the module Blackboard site by the deadline. The walkthrough will last approximately 50 minutes. This will consist of 25 minutes for you to present your work and 25 minutes for discussions and feedback.

You should present your implementations to us as if we were the clients and demonstrate its robustness by showing how you tested the system and the results.

## Marking Process and Submission

The majority of the marking and feedback will be undertaken during the walkthrough you provide as a group shortly after the submission deadline. All group members should participate equally during the walkthroughs, perhaps choosing to talk about the specific aspects that you worked on. The walkthroughs will be presented to one or two of the module delivery team. Your team will be able to choose a timeslot to present your work via a scheduling link on the module Blackboard site.

In brief, marks will be awarded under the following headings:

|  |  |
| --- | --- |
| **Implementation**   * Class structure as per class diagram * Functionality from sequence designs * User interfaces from designs * Functionality and scope | 16 marks |
| **Quality of engineering practice**   * Appropriate JavaDoc generated * Use of Version control * Well commented code * Use of design patterns and other techniques to enhance maintainability | 7 marks |
| **Formal Testing**   * Four unit test suites across a varying complexity of classes * Acceptance test plan (four test specifications and test descriptors) | 7 marks |
| **UI Evaluation Report** (max 4 sides of A4 + appendices)   * Approach to evaluation used and justification * Plan for evaluation * Ethical considerations * Report on evaluation conducted * Results and re-design suggestions | 15 marks |
| **Critical reflection report** (max 2 sides of A4) | 5 marks |

*Reports on UI Evaluation and Critical reflections may be reviewed and marked outside of the*

*walkthroughs.*

The same mark will be awarded to each group member so it is up to you how you divide the work. If there are issues with the level of contribution within the team, these need to be raised in advance of the submission deadline with the module delivery team.

All the implementation and documentation created for this subtask must be uploaded as a single ZIP or 7z file to the module Blackboard site via the subtask three assignment upload link. You must include all project files that will allow the Java application to be opened and executed on a campus PC. This deadline is shortly before the walkthroughs but must contain all materials you wish to present.

The deadline for electronic submission is **Friday 11th March 2016.**