

## PROJECT OVERVIEW

## DESIGN AND IMPLEMENT A WEB APPLICATION

This project is worth 30% of the final mark.

### Objectives

- To gain experience in the design of **complex** software, and in translating the design into implementation
- To gain experience in working in a large software development team
- To demonstrate software development skills using C# and .NET Core

## 1 INTRODUCTION

The class will be designing and implementing two **complex** Web applications. These applications are the “Automobile Workshop Jobs Ledger” and the “Travel Portal”. Students enrolled in Lab P2 will work on the “Travel Portal” application and students enrolled in Lab P1 will work on the “Automobile Workshop Jobs Ledger” application.

The requirements for each of these applications are provided in separate documents. Each application has been divided into a series of modules that provide some part of the functionality of the application. Each module will be normally worked on by a team of 5-6 students. The team will design and implement the module such that it will integrate well with the other modules to make the whole application work.

## 2 REQUIREMENTS

Each module is to be completed by a team of five or six students. Teams will be finalised by Wednesday, 15 January 2020 and announced on xSiTe once the team members allocations are confirmed. Selection of modules for each team has to be finalised by the Lab 1 (Week 2). Each team will work on only one module.

## 2.1 SOFTWARE DESIGN AND IMPLEMENTATION

The applications are to be developed using the ASP.NET Core Framework in C#. As such, the user interface to the application will use the Model-View-Controller architecture. Individual modules, however, may use different architectures as appropriate to the functionality of the module. Applying detailed design patterns as appropriate throughout your module (but do not use them where they are not appropriate!) should be considered.

Explain the rationale behind the design when documenting your design. In the explanation, cover why the team chose the architecture(s) and detailed design (including design patterns). Discuss the reasons why you have rejected other architectures or designs that might consider. Finally, explain any trade-offs the team made when selecting one design over another, and the software design quality attributes that you intend to achieve.

## 2.2 DOCUMENTATION

For consistency across the entire application, documentation should follow the template provided on xSiTe. Your documentation should be readily understood by software engineers who are familiar with general software engineering practice but may not be familiar with your particular project. In particular, the documentation for your module should be understandable to your peers, who will be implementing other modules that interact with it.

## 2.3 HOSTING AND INTEGRATION

Source code for the overall application can be hosted on a version control server such that each team can share the current version of its module with other teams, and each team will be able to build the whole application including its module(s) for testing purposes.

## 2.4 OTHER REQUIREMENTS

The Web site should have a clean and consistent look-and-feel throughout, using CSS stylesheets and conforming to principles of good (Web) design that you have learned in ICT1004 and ICT2102. However, since our focus in ICT2106 is on software design and not Web design, be aware that **only code on the server-side will be assessed**. Client-side code can be used to invoke features of the server (such as via AJAX) as appropriate to the application, but only the server-side code will be assessed.

### 3 SUBMISSION AND DELIVERABLES

The project component of ICT2106 is divided into an allocation phase followed by four iterations of the software. Table 1 summarises the deliverables and deadline for each iteration. Table 2 summarises the requirements for each deliverable.

Each module will be graded out of 100 marks, distributed amongst the iterations as shown in Table 1. Note that, even though the proposal is not graded, you may be asked to re-submit it if the proposal does not meet the minimum requirements outlined in Table 2.

TABLE 1 – PROJECT ITERATIONS AND DUE DATES (SUBJECT TO MINOR ALTERATIONS)

Iteration	Deliverable(s)	Marks	Deadline
0 – Allocation	<ul style="list-style-type: none"><li>• Submit preferences for the module to be developed</li></ul>	Not graded	Week 2
1 – Proposal	<ul style="list-style-type: none"><li>• Proposed project design</li><li>• Plan for completing Iteration 2*</li></ul>	Not graded	Tuesday, 28 January 2020
2 – Prototype I	<ul style="list-style-type: none"><li>• Features and boundary that your team is working on</li><li>• Initial design document**</li><li>• Prototype^<ul style="list-style-type: none"><li>• Complete Interfaces and Interface-command classes</li><li>• Structure of your components' classes</li></ul></li><li>• Plan for completing Iteration 3*</li><li>• Iteration 2 presentations and discussions</li></ul>	50	Week 6
3 - Prototype II	<ul style="list-style-type: none"><li>• The integrated design document for database and software**</li><li>• Changes made to the boundaries</li><li>• Plan for completing iteration 4*</li></ul>	Not graded	TBC
4 – Final Prototype	<ul style="list-style-type: none"><li>• Final design document**</li><li>• Final application^^</li><li>• Presentation/demonstration</li><li>• Completed project plan*</li></ul>	50	TBC
TOTAL		100	

TABLE 2 – REQUIREMENTS FOR DELIVERABLES (SUBJECT TO MINOR ALTERATIONS)

Deliverable	Requirements
Proposed Project Design (Iteration 1)	<ul style="list-style-type: none"> <li>• Identify the features to be provided by your team's module.</li> <li>• Overview components diagram of the project and highlight the component(s) your team is working on in the overview components diagram.</li> <li>• Discuss the requirements gathered from the project for your team's module. Please also include the requirements negotiated with the other teams for your team's component(s).</li> <li>• Discuss the generic architecture of your team's component(s) and the architectures for the individual components if they are independent of the generic architecture of your team.</li> <li>• The class diagram(s) including each new class(es) and subclass(es) to be implemented.</li> <li>• No more than 600 words</li> </ul>
*Project Plan	<ul style="list-style-type: none"> <li>• A list of the tasks assigned to each team member since the previous iteration. Include the current state of the task.</li> <li>• Using 2-3 paragraphs, each team member should reflect on the state of the project, his/her contributions, and the cause of any delays to tasks intended to be completed</li> <li>• A list of the tasks assigned to each team member for the next iteration</li> </ul>
**Design Document	<ul style="list-style-type: none"> <li>• Architecture diagram of the overview project and the individual component(s) in your team's module.</li> <li>• A UML class diagram including all of the classes and components in the module</li> <li>• An explanation of the rationale of the design decisions made, including which architecture and design patterns have been used</li> <li>• Note: Iteration 2 will contain a draft design to be implemented while Iteration 3 will contain the final design as implemented</li> <li>• No more than 2000 words</li> </ul>
^Prototype	<ul style="list-style-type: none"> <li>• The module with at least one feature fully implemented and working</li> <li>• Source code</li> </ul>
^Final Application	<ul style="list-style-type: none"> <li>• The module with features implemented and working</li> <li>• Source code</li> </ul>

All submissions should be made through the Assessment Dropbox on xSiTe. At the completion of Iteration 4, your team will be required to present your work and discuss your component with the teaching staff. The time and location for demonstrations will be announced in the lead up to each submission deadline.

Late submissions will receive a penalty of 15% per day, up to four days after the due date. Late submission of proposals will be penalised by a reduction of the marks awarded. No submissions will be accepted more than four days after the due date.

## 4 ASSESSMENT CRITERIA

At the end of each iteration, each module will be assessed for

- completeness and consistency – does the submission meet all of the requirements listed in Table 2, and are the documentation, design and implementation complete and consistent?
- Design quality – does the design conform to the guidelines for good design given in the course materials (see below)?
- Correctness – does the software function as intended?
- Presentation – is the documentation readily understandable and formatted in a consistent manner?

Note that if your project proposal is not submitted on time in iteration 1 (Project Proposal). Your iteration 2 (Project Prototype I) submission will not be marked, and zero marks will be awarded to your team's iteration 2 – Project Prototype I assessment component.

The assessors will pay particular attention to the following. Make sure that your design:

- covers its **design purpose** and **primary functionality** of the project
- considers the “**Design Qualities**” and “**Run-time Qualities**” (see Week 1 – Lecture, Quality attributes)
  - Quality assurance
  - Design decisions and challenges faced
  - Software engineering approaches applied
  - Documentation to illustrate your design concepts
  - Ability to translate your design concepts to actual implementation
- makes appropriate use of design patterns (Part 2 of ICT2106 onwards)
- conforms to the S.O.L.I.D. design principles (see Week 8 - Lecture) (Part 2 of ICT2106 onwards)

## 5 NOTES ON PLAGIARISM

The University's policy on copying does not allow you to copy software as well as your assessment solutions from another person. While this project requires you to work with software created by other students, presenting another person's work as your own is unacceptable, and it is the responsibility of all students that their assessment solutions are their own work. Where such plagiarism is detected, the assessments involved will receive **ZERO** mark.

If your application incorporates third-party libraries (apart from .NET Core itself), these should be identified in your design document according to the licence under which you are using those libraries.