EIT
TE AHO A MĀUI
Te Pūkenga

# ASSIGNMENT ONE

*Semester 2, 2024*

# ITPR7.508 Business Application Programming

| | |
|---|---|
| **Weighting:** | 70% of final grade |
| **Marks:** | 200 |
| **Due Date:** | Refer to milestones with the marking schedule and the course outline:<br>+ Week 3: Part 1 - Proposal documentation – Sunday 11/8.<br>+ Week 8: Part 2 - Formal Documentation – Sunday 15/9<br>+ Week 10: Prototype 1.0 Release – Sunday 29/9<br>+ Week 12: Prototype 2.0 Release – Sunday 27/10<br>+ Week 14: Final Application (3.0) Release & Deploy – Sunday 10/11 |
| **Time:** | 11:30 PM |

*The School of Computing has a policy of no late assignments. However, an assignment handed in up to 24 hours late will be marked. A deduction of 20% of the total marks available will be made. Any assignments received more than 24 hours late will not be marked but can be used as evidence of completing terms.*

## INSTRUCTIONS:

- You will be working in a **team** and will hand in one set of documentation and program code.
- All work submitted must be original and entirely your own work, except where you use ideas, quotations, tables, diagrams, code or any other material from other writers. In such cases you must acknowledge the source using the APA referencing style.
- No part of the work submitted may be used as part of any assessed work for any other academic course.
- All work submitted must be of a professional standard. Assignments must be delivered on-line through the assignment submission link on or before the due date.

# Marking and team contribution

## Introduction

Assessing and assigning marks is a tricky and subjective exercise as there are many ways to write documentation and code that performs the same function/operation.

## Marking

We will adopt the following approach to evaluating your code, code quality and documentation:

- Code does not work, missing or poor documentation - expect < 40% of the mark
- Code works but may contain program breaking bugs, documentation exists but could be do with additional detail – expect up to 75% of the mark
- Code works as expected and contains no program breaking bugs and well written documentation – expect > 75% of the mark

## Team Member Contribution

To reduce the workload we are executing the programming activities as a team based assessment. Therefore it is important to assess the level of contribution of each team member to ensure fairness.

At the completion of the assessment, each team member is expected to submit a brief summary of their contribution and how they think their team member contributed/performed.
Use the following as a guide for the degree of contribution;

### *Criteria*

- **Preparation** (Was the team member prepared?)
- **Participation** (Did each team member do their fair share of the development?)
- **Reliability** (Was the team member reliable in delivering the work?)

### *Rating*

1 = consistently poor
2 = usually poor
…
5 = ok
…
9 = usually excellent
10 = consistently excellent

# The Enterprise Application

## Introduction

You will be working as part of a development team and you will work on one individual and two team-oriented assignments. A project management server will be required to facilitate the team activities. These activities include and are not limited to:

- Online meetings and meeting minutes.
- Task scheduling & issue tracking.
- Document and file sharing.
- News & email.
- Wiki.
- Shared code development using git.
- Programming in language of choice.

## Participation

All meetings and individual participation and effort will need to be documented in detail. All team activity needs to be conducted through the developer collaboration platform like GitHub (or EITOnline if any issues arise with the project management server). All team members will be required to participate in joint reports. These activities document the individual experiences and assist in determining the percentage effort contributed by each of the team members. The work done on the project management server will be considered when assigning the individual student marks.

## Deliverables

Documents must be in a professional electronic common format (e.g.: PDF, MS Office, OpenOffice, Apple Productivity Apps, etc.) and diagrams must be produced using either a case tool or some form of automated drawing tool such as https://www.diagrams.net/. A professional set of documents must be the result of these activities.

Programming code must be managed through a code versioning system (e.g.: Git). Each team member MUST actively participate in this code versioning system through their own account.

## Teams

Team composition will be determined at the start of the course and team members will be matched according to abilities and the tasks required by the case study. One member of the team may be assigned the role of Team Leader which will be considered as one of the required tasks. Each member's tasks and expected contribution needs to be clearly indicated in the *Part1: Proposal*.

## Tasks

The case study requires several skills to produce a final application and professional documentation. The following short-lists some of the expected tasks that will be assigned to various team members based on their skills:

- Project management
- Design documentation (specifications and diagrams for wireframes, stories, ERD, DFD/UML, etc.)
- Database design & usage
- Interface design & construction
- Programming
- Testing

Tasks can be shared or split up.

# Marking schedule 200 Marks

*Requirements Analysis and Specification for a Software Application*

You will meet and communicate with your client who will discuss the product they want to have built. This will be the starting point for this project and you will need to document the User Requirements Specification (URS). The following needs to be completed based on the URS.

## *Part 1: Proposal – due end of week 3*

Process the URS given by the client and produce the following:

- A detailed quote for the project based on the URS. You will need to quote for a product delivery at the end of Week 14.
- The quote will include:
  - business issues and value-add to the client.
  - program specifications.

[10 marks]

- Summary of program specifications & requirements
- A report outlining the different tasks and roles of the team members, and how these relate to the actual final product. Construct a list of deliverables per team member.

[10 marks]

## *Part 2: Formal Documentation – due end of week 8*

Produce the following documentation:

- Reflection/Analysis of the business issues and justifications of decisions that you have agreed to.

  [15 marks]
- Reflection/Analysis of the program specifications. Include any changes or modifications that may be required for the final software application.

  [15 marks]
- A detailed ERD that best reflects the relationships, tables, and business rules documented in the URS.

  [10 marks]
- Design documents (diagrams as DFD or UML) for the user interface, the client-server setup, and the individual processes within the software application.

  [15 marks]
- A detailed description of a test framework and plan that can cover all required features. Include code examples of such tests – unit testing.

  [15 marks]

## *Part 3: Presentation – during week 14*

Give an oral presentation of your project including the following topics:

- General overview of the client's business.
- Discuss how your proposed product fits the business and provides a value-add.
- Discuss highlights that have resulted from your work and can be of interest to the client.
- A clear understanding of the constraints/boundaries of this project.

[10 marks]

### *Part 4: Final Software Application - due end of week 14*

- Design a software application that fulfils the requirements as outlined in the given URS.
- Use a team-oriented project management tool (e.g. Asana, Bitrix, Trello, ...) to manage all tasks, resources, meetings, and documentation.
- Use source/revision/version control software (e.g. BitBucket, GitHub, ...) to manage the project's
- source code.
- The software application must be a complete product offering.
- The software application must be written in the programming languages that are determined in this course.
- These programming languages can interface/connect to an industry recognised database server (Firebird, MS SQL server, MySQL, PostgreSQL, Oracle, Sybase, etc.).

Marks are distributed as follows:

- All features in the program specification have been professionally implemented and documented.
  - ○ Program code does not work, missing or poor documentation - expect < 40% of the mark
  - ○ Program code works but may contain program breaking bugs, documentation exists but could be do with additional detail – expect up to 75% of the mark
  - ○ Program code works as expected and contains no program breaking bugs and well written documentation – expect > 75% of the mark

**[30 marks]**

- Most features / the significant features have been tested applying the test framework.

**[20 marks]**

- The product has been constructed as a team. Team communications, task distributions, issue resolutions, team meetings etc. are readily available and of a professional standard.

**[20 marks]**

- The final product is handed over to the client (incl. final meeting / presentation)
  The presentation includes:

| | |
|---|---|
| ○ a brief introduction to your project | [5 marks] |
| ○ an overview of the development methodology | [5 marks] |
| ○ reflection on team work and its effect on the final product | [5 marks] |
| ○ significant features and consequences for the client | [5 marks] |
| ○ a product demonstration | [5 marks] |
| ○ a transfer of the complete product resources and sign-off of UATs | [5 marks] |

**[30 marks]**

Since you are working as a development team, you will be required to conduct regular team meetings. The minutes and action plans produced at the meetings must be documented and published within the online project management tool.