COMP1004
Computing Practice
ASSESSMENT BRIEF

40 CREDIT MODULE

ASSESSMENT: 80% Coursework
20% Practice

MODULE LEADER: **Dr Shirley Atkinson / Dr Hai-Van Dang and Dr Alaa Alkhafaji**

CO-TEACHERS **Dr Liz Stuart**

MARKER PANEL: **Dr Marco Palomino**
Dr Vasilios Kelefouras

MODULE AIMS

- To familiarise students with the software development lifecycle
- To develop techniques to elicit requirements from abstract problems
- To expose students to both technical and non-technical skills required to complete a software engineering project

ASSESSED LEARNING OUTCOMES (ALO):

1. Articulate the phases of the software development lifecycle.
2. Define suitable functional and non-functional requirements that when implemented in software will solve a given problem.
3. Identify and describe operational considerations of software, including the physical systems upon which they reside, cyber security and networking.
4. Implement a system that fulfils a design specification to solve a given problem
5. Identify and discuss trade-offs in requirements analysis and the production of software versus operational considerations such as time and cost.

Overview

This document contains all the necessary information pertaining to the assessment of *COMP1004 Computing Practice* module. The module is assessed using two elements. A **coursework** element worth **80%** of the whole module and a **practice** element worth **20%** of the whole module. Both elements have two submission points for each, one at the end of Semester one and the last at the end of Semester two.

The sections that follow will detail the assessment tasks that are to be undertaken. The submission and expected feedback dates are presented in Table 1. All documents are to be submitted electronically via the module DLE pages before the stated deadlines and all presentations will be held in the School Computing labs (Smeaton 101, 107, 108, 109, 200 or 201)

	Submission Deadline	Feedback
1 Interim Report	**Tuesday 10 January 2023, 4:00 PM**	**8 February 2023**
2 Interim Video	**Tuesday, 10 January 2023, 4:00 PM**	**8 February 2023**
3 Final Portfolio	**Tuesday, 16 May 2023, 4:00 PM**	**14 June 2023**
4 Showcase demonstration	**Monday, 22 May 2023**	**23 June 2023**

Table 1: Assessment Deadlines

All assessments will be introduced during the first timetabled session to provide further clarity over what is expected and how you can access support and formative feedback prior to submission.

This project module is an all year, 40 credit, module that provides you with an opportunity to learn how to apply problem-solving techniques to understand how to create technological solutions to solve a problem. This gives you the chance to really understand the core principles of Computer Science, the software development lifecycle and how that applies to the degree you are studying.

Purpose

The purpose of this assignment is for you to gain the knowledge you need to be successful in your studies for your degree and later for being an employable, effective graduate in this overall discipline. The assignment is designed to focus you on the skills you need to learn and for you to start the process of acquiring those skills.

Task

You are to create a single page application (SPA) addressing a particular scenario.
You may either work in a group or individually. Whatever your choice, you will need to attend the group stand-up meetings as usual. Each student (working individually or in a group) must submit an individual report and present individually at the Project Closure Workshop.
The project must span the whole academic year, starting now at the end of September and completing in May. Your project must encompass the milestones given below.

Milestones

Milestone	**Description**	**Deliverable**
18th October	Finalise Cluster Formation.	Meeting minutes in group repo
20th October	Finalise Requirements Documentation (supported via COMP1000)	Product Backlog and initial UML diagrams in group repo
27th October	Sprint 1 Review with Liz Stuart	Review document in group repo
10th November	Sprint 2 Review with Liz Stuart	Review document in group repo
24th November	Sprint 3 Review with Liz Stuart	Review document in group repo
8th December	Sprint 4 Review with Liz Stuart	Review document in group repo

10th January	Interim Submission	PDF documentation and Video submitted via DLE
	Fortnightly sprint reviews to be arranged	Review documentation in group repo
5th April at 4:00 PM	Register your choice (Group or Individual) on the spreadsheet. See below for full details.	Register your choice at link
16th May	Final documentation submission	PDF documentation via DLE
22nd May	Project Closure Workshop	In person event

Project Information

The module is divided into two halves – aligning with the semesters. At the start of the module, you will be given a scenario to work to and for the rest of Semester 1 you will undertake planning activities to create the software required to meet that scenario.

You need to identify the skills you need to create the software bearing in mind your role will be tailored for your degree type – eg: games students should create a game element, security students should create software to address a security element (passwords for example).

Having identified your skills, you must make a plan, consider how the software development lifecycle affects how you carry out your activities and work together to create the blueprint of your software.

Semester 2 sees you implement the software and evaluate your plan. The technologies in use will be basic web technologies HTML, JavaScript (including jQuery, JavaScript Canvas), JS libraries (including Phaser), AJAX and CSS (including Bootstrap) combined with C#, the language you are learning in your other modules. You must not use any other frameworks or languages other than those specified in the materials. For example, node.js, Jetty API, Angular and React are not allowed, neither is Unity or any game engine.

You must use the GitHub classroom link provided here (https://classroom.github.com/a/hHe5KfX8) for your source code and documentation. You are expected to commit on a regular basis – multiple times per sprint.

Throughout this all-year module you will be expected to work through the materials on the DLE, to participate in unsupervised pair-programming activities in the labs, to engage with taught sessions and check-in with a tutor throughout the academic year whilst you are developing your own software project.

Submissions

This section outlines what you are expected to submit.

A portfolio is a collection of items that together will demonstrate you have met the learning outcomes of the module. Please refer to the template structure provided on the DLE.

Interim Report

This section provides you with the information you need for this **individual** submission. This will account for 30% of your Coursework element.

Reflect upon your activities to date and write a reflective account providing evidence for the following sections:

- Project Plan

    - How did you plan what you were going to do?
    - Link to the project backlog, sprint plans, test plan.
- Self-Learning Plan
    - How are you going to learn what you need to learn.
- Requirements
    - Reflect on how you analysed the requirements aspect for your part of the project.
    - Link to the UML documentation that you have created and the product vision.
- Architecture
    - Reflect on the architecture and how that relates to your part of the project.
    - Link to the UML diagrams that represent your aspect of the project.
- Prototype implementation to date
    - Provide screenshots of what YOU have implemented so far.

To justify your reflections, you should provide links to the items you were assigned on the product backlog, the sprint plans you were part of, the tests you carried out etc. You are providing evidence for YOUR activities.

Interim video

This section provides you with the information you need for this **individual** submission. This will account for 30% of your Practice element.

Record you and others in your cluster are to record a short video of you having a structured discussion regarding your reflections of your project to date how this affects what you wish to learn. Each of you must take it in turn to show what you have been working on. You should account for 2 minutes each. Upload your part for your submission.

You must upload your video to YouTube, have it unlisted (NOT private) and copy the URL into the text box for the submission. DO NOT upload the video.

Interim Assessment Criteria

All submitted items are evaluated based on the following categories

Learning Outcome mapping	Weighting	Evidence
LO1	30%	The following evidence is examined for understanding how to plan appropriately and the impact of the SDLC upon those plans. • Project Plan (Backlog, Sprint plans, Test plan) • Video
LO2	30%	The following evidence is examined for understanding how to extract requirements from a scenario. • Project Vision, Requirements outline • Video
LO3	30%	The following evidence is examined for understanding the operational considerations for the scenario. • Architectural outline, UML documents • Video
LO1, LO2, LO3	5%	Evidence is sought for reflections on how to move personal skill set forward appropriately.

		• Self-learning plan
LO4	5%	Evidence is sought to see what prototyping has been done to date. • Implementation screenshots • Video

Threshold Criteria (these are indicative only)

To achieve a 3rd class (40%+), you must demonstrate an understanding of the key steps to planning, analysing and creating a piece of software.

To achieve a 2.2 (50%+), you must demonstrate an understanding of the key steps to planning, analysing and creating a piece of software and show how this is applied. Some evidence of referencing.

To achieve a 2.1 (60%+) you must demonstrate a good understanding of the key steps to planning, analysing and creating a piece of software and have a good knowledge of applying appropriate coding strategies to your work. Evidence of referencing throughout the background.

To achieve a 1st (70%+), you must demonstrate an excellent understanding of the procedure of planning, analysing and creating a piece of software and have a solid knowledge of applying appropriate coding strategies. It is expected that there would be a good use of references to support the background.

3 Final Report*NEW

This report is an individual deliverable that accounts for **70%** of your Coursework mark.

You must choose either (i) to continue working in your assigned cluster or (ii) to work individually for the remainder of the module. You are required to register your choice before Wednesday 5th April at 4:00 PM. To register your choice add the word “Cluster” or “Individual”, adjacent to your name in this spreadsheet (link).

Note that if you continue to work in your assigned Cluster:

- You are required to submit all deliverables individually.
- On the first page of your final report, you must first clearly state your Cluster number, and the names of everyone in your Cluster.
- Follow this with the link to your GitHub repository. Each member of your Cluster will provide the same link to the Cluster’s GitHub repository. Therefore, it is essential that you clearly highlight which are your individual contributions to the overall project. This will enable your contributions to the project to be marked individually.

Alternatively, if you choose to work individually:

- You are required to submit all deliverables individually.
- On the first page of your final report, you must first clearly state your name.
- Follow this with the link to your own individual GitHub repository.

Your final submission should reflect how you have implemented your project plan, highlighting any changes or developments that have occurred. As before, you should link to the evidence in the GitHub repository to justify your reflections.

The structure of the final report is as follows:

i. Front page (this will not be counted into the page count).
ii. GitHub link
iii. Product vision
iv. Project plan
    a. Sprint plans (with dates and main highlights). Sprints should cover your work on Requirements, Analysis (including UML), Design, Implementation & Testing. Sprint plans would need to show how the user stories were implemented **with links to the appropriate commits**
    b. Project Backlog (what is still to be developed)
v. Evaluation
    a. Product review (evaluating what was produced)
    b. Project review and reflection (evaluating your process)

Throughout the report, you are required to comment on how the project has changed since the interim submission. As well as specifying how the project has changed, you are required to reflect and report on the impact of these changes.

Format your report as follows: Use Arial as the font, with a minimum size of 11pts, all margins should be at least 2cm. The final report should be no longer than 10 pages (the markers will stop assessing at the end of the tenth page). Further instructions are described in Final Assessment Criteria section.

4 Participation in Project Closure Workshop

Participation in this workshop will account for **70%** of your Practice mark.

As per the date provided above. The workshop will take place over the whole day. Each student will be allocated to a session, during which they will present. All students allocated to a session must stay for the whole session. Students unable to attend on the day due to illness must claim ECs for non-submission.

You will have 6 minutes and must present the following in this order:

1. The application running
2. Your final project plan and sprints
3. One of the more complex set of UML diagrams
4. High level summary of issues and challenges faced
5. High level summary of software testing, verification and validation

The mark will be based on the presentation and demonstration.

Final Assessment Criteria

The final report will be evaluated using the following categories

Learning Outcome mapping	Weighting	Evidence for the final report
LO4	60%	The following evidence is examined for implementation skill: • GitHub source code (You need to provide your GitHub classroom link) • Evidence of implementation: Illustrate with screenshots of the web application, and links to your source code in your GitHub repo to demonstrate how you implemented each feature. Provide suitable narrative to help your reader understand your screenshots and diagrams. • The functionality and scale of your product should reflect the number of people who have contributed to it
LO1, LO2, LO3	20%	The following evidence is examined for understanding planning cycles, requirements and architectures: • Aspects of the report relating to planning, requirements and system architecture. • Description and reflection on the process the project has followed, with a rationale behind any changes that have been made.
LO5	20%	Evaluation: • Software testing, verification and validation: Test plan showing how the web application and user stories have been tested and the results. Discussion of the results. • Project review and reflection: Detailed evaluation of the achievement of the project aims and objectives. A reflective evaluation of the project methods, approaches, technologies, summary of issues and challenges etc. • Recommendations for further work.

The Project Closure Workshop presentation is evaluated under the following categories (You must attend the Project closure workshop day)

Learning Outcome mapping	Weighting	Evidence in the workshop
LO4	60%	The following evidence is examined for implementation skill: • The application running and demonstration of the implemented features • The understanding of your implementation
LO1, LO2, LO3	20%	The following evidence is examined for understanding planning cycles, requirements and architectures: • The presentation of your final project plan and sprints, one of the more complex set of UML diagrams
LO5	20%	Evaluation: • The presentation of high level summary of software testing, verification and validation

		• The presentation of high level summary of issues and challenges faced

Threshold Criteria (these are indicative only)

To achieve a 3rd class (40%+), you must demonstrate an understanding of the key steps to planning, analysing and creating a piece of software.

To achieve a 2.2 (50%+), you must demonstrate an understanding of the key steps to planning, analysing and creating a piece of software and show how this is applied. Some evidence of referencing.

To achieve a 2.1 (60%+) you must demonstrate a good understanding of the key steps to planning, analysing and creating a piece of software and have a good knowledge of applying appropriate coding strategies to your work. Evidence of referencing throughout the background.

To achieve a 1st (70%+), you must demonstrate an excellent understanding of the procedure of planning, analysing and creating a piece of software and have a solid knowledge of applying appropriate coding strategies. It is expected that there would be a good use of references to support the background.

General Guidance

Extenuating Circumstances

There may be a time during this module where you experience a serious situation which has a significant impact on your ability to complete the assessments. The definition of these can be found in the University Policy on Extenuating Circumstances here: https://www.plymouth.ac.uk/uploads/production/document/path/15/15317/Extenuating_Circumstances_Policy_and_Procedures.pdf

A time extension can only be claimed for the interim submissions and the coursework submission, it cannot be claimed for the Showcase presentation. If unforeseen circumstances have affected your ability to deliver on the day of the showcase presentation you must claim for non-submission. This will lead to you being offered the opportunity to present at the Referral Showcase day which will be a day held on campus in mid-August.

Referral

Students who fail the module or have Extenuating Circumstances (ECs) for non-submission may be offered the chance to carry out a referral over the summer. The referral is suitable only for those who have carried out some work during the academic year, it is not suitable for those who have not been able to work on the project during the academic year. Those students should ask to repeat.

This module has two elements, and it is possible to be referred in one or both elements. Students unable to complete their practice element, (for example, due to ECs were unable to attend the Marketplace Demo and the final Showcase) should claim their ECs for non-submission and providing their referred credits do not exceed the current limit (60 credits) will be invited to present at a Showcase day in the August referral period. This will take place during the University referred examinations period. The information of the exact date can be found here https://www.plymouth.ac.uk/student-life/your-studies/essential-information/regulations/important-dates

Plagiarism

All of your work must be of your own words. You must use references for your sources, however you acquire them. Where you wish to use quotations, these must be a very minor part of your overall work.

To copy another person's work is viewed as plagiarism and is not allowed. Any issues of plagiarism and any form of academic dishonesty are treated very seriously. All your work must be your own and other sources must be identified as being theirs, not yours. The copying of another persons' work could result in a penalty being invoked.

Further information on plagiarism policy can be found here:

Plagiarism: https://www.plymouth.ac.uk/student-life/your-studies/essential-information/regulations/plagiarism

Examination Offences: https://www.plymouth.ac.uk/student-life/your-studies/essential-information/exams/exam-rules-and-regulations/examination-offences

Turnitin (http://www.turnitinuk.com/) is an Internet-based 'originality checking tool' which allows documents to be compared with content on the Internet, in journals and in an archive of previously submitted works. It can help to detect unintentional or deliberate plagiarism.

It is a formative tool that makes it easy for students to review their citations and referencing as an aid to learning good academic practice. Turnitin produces an 'originality report' to help guide you. To learn more about Turnitin go to:
https://guides.turnitin.com/01_Manuals_and_Guides/Student/Student_User_Manual

Referencing

The University of Plymouth Library has produced an online support referencing guide which is available here: http://plymouth.libguides.com/referencing.

Another recommended referencing resource is Cite Them Right Online; this is an online resource which provides you with specific guidance about how to reference lots of different types of materials.

The Learn Higher Network has also provided a number of documents to support students with referencing:
References and Bibliographies Booklet:
http://www.learnhigher.ac.uk/writing-for-university/referencing/references-and-bibliographies-booklet/
Checking your assignments' references:
http://www.learnhigher.ac.uk/writing-for-university/academic-writing/checking-your-assigments-references/