

# **COMP2000**

## **Software Engineering 2**

**20 CREDIT MODULE**

**ASSESSMENT: 100% Coursework**

**W1: 30% Set Exercises**

**W2: 70% Assignment**

**MODULE LEADER: Dr. Vivek Singh**

### **MODULE AIMS**

- To learn about topics that instil best practice into the students' software development activity.
- To explore a range of commonly used programming paradigms.
- To understand the benefit of using standardised design patterns.

### **ASSESSED LEARNING OUTCOMES (ALO):**

1. Apply HCI and parallelism to construct more efficient and usable software.
2. Illustrate the relative merits of various programming paradigms
3. Use appropriate programming paradigms to implement software that solves a given problem.
4. Compare standard design patterns and explain why they are an important aspect of software engineering

## Overview

This document contains all the necessary information pertaining to the assessment of *COMP2000 software engineering 2*. The module is assessed via **100% coursework**, across two elements:

- 1- 30% *Set Exercises*
- 2- 70% Assignment Submission.

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 assessments are to be submitted electronically via the respective DLE module pages before the stated deadlines.

	Submission Deadline	Feedback
Set Exercises (30%)	<b>04/11/2024</b>	<b>03/12/2024</b>
Assignment Submission (70%)	<b>07/01/2025</b>	<b>05/02/2025</b>

Table 1: Assessment Deadlines

All assessments will be introduced in class to provide further clarity over what is expected and how you can access support and formative feedback prior to submission. Whilst the assessment information is provided at the start of the module, it is not necessarily expected you will start this immediately – as you will often not have sufficient understanding of the topic. The module leader will provide guidance in this respect.

### Set Exercises

There are six set exercises for you to complete during the course of the module. The exercises should be submitted to the module DLE by the deadline shown in the table above. The answers for all six exercises should be submitted in one PDF file.

## Assessment 1: Set Exercises

### CW1 Overview

There are six set exercises outlined below that will allow you to demonstrate your knowledge and understanding around user graphical design, mobile applications, usability evaluation. Submit a single PDF file that contains the outputs from each of the exercises in the same order as they appear below. The outputs for each exercise are clearly described. You are to use the GitHub Classroom set up for this module – the link for signing up is here:

<https://classroom.github.com/a/tkdtCm72>

You are to use the scenario in the appendix and base your six exercises for both admin and employee sides.

Based on the scenario you are to undertake and document an analysis of the users' needs. To do this you must:

#### Set Exercise 1: 5 marks

- Analyse the scenario to Identify the context of use and the user(s) of the system.

#### Set Exercise 2: 15 marks

- Produce a design of the UIs for a client-side mobile application - a low fidelity prototype (i.e. paper-based prototype) and provide a storyboard. The design should follow the HCI principles that you learnt during the semester.

#### Set Exercise 3: 25 marks

- Carry out a user testing study (Formative usability evaluation) for the initial design using the paper-based prototype: Invite a minimum of two potential users. Set the user study based on knowledge you gained in the course using one of the user's testing studies you learnt in the course.

#### Set Exercise 4: 15 marks

- Re-design the UIs based on the outcome of the user testing in the previous exercise (users' feedback). Highlight the changes with a brief explanation. You might consider different screen sizes for different mobile devices.
- Produce a storyboard for the final design with appropriate narrative to describe each interface from the user perspective.

#### Set Exercise 5: 30 marks

- Implement the final UIs design of the employee side using Java and Android Studio (i.e. a high-fidelity prototype).
  - The UI components in the layouts should be of correct type for adding functionality at later stage.
  - The UI should contain all the pages/layouts for the final solution. You don't need to add any functional components in java at this stage.

**Set Exercise 6: 10 marks**

- The designed UI should compile without error and should be visible in the emulator correctly.
- All warning messages should also be handled properly.

**Deliverable**

Submit a single PDF file to the DLE submission point for this module

- The PDF document reporting your work:
  - The users of the system.
  - The initial design of the UIs and the storyboard (i.e. screenshots and/ or sketches).
  - Full details of the process of usability evaluation (user testing) (e.g. how the study was conducted, the testing plan, demographics of the participants, the consent form, the outcome of the study) with evidence of the study setting (i.e. pics or link to the video).
  - The final design based on the user's feedback highlighting the differences with the initial design.
  - The storyboard of the final design with the appropriate narratives.
  - The link to the Github repo that contains your source code.

## Assessment 2: Assignment Submission

You are required to produce an Android application – a client-side application for the admin side only to work with The RESTful API. The focus for your application is to enable the admin to be able to upload their employees' details to the website. A reminder of the API detail is provided in the appendix below with the user stories adjusted to focus on the admin requirements.

A link to the API is available in the **API document** on the DLE page under Assessments. The file provides details of the root path and different available endpoints for that API. Please refer to the appendix for further details on the coursework tasks.

Your implementation must be in the Java programming language using Android Studio. You must create a worker thread to handle the connection to the API.

Use design patterns that you learnt throughout the course. You could use the design you produced in the previous assessment. Carry out a summative usability evaluation with at least two users. The app should push notifications based on user preferences.

The Java project will provide a graphical user interface (GUI) for usage. The GUI should provide the functionality listed in the Scenario. You **MUST** use Git version control with GitHub in your development process. You will be assessed on appropriate use of version control. You are to use the GitHub Classroom set up for this module – the link for signing up is here: <https://classroom.github.com/a/1oOjCPDs>

### Assessment Criteria:

Individual Coursework comprising one assignment with a GIT repository of incremental development leading to a final build or release. Repository includes itemized deliverables for final module submission. Typically, this will include your Android project and usability evaluation documentation in PDF, with a current build or release executable. All code to be version controlled and commented, all 3rd party assets and resources to be formally credited in the README file on the repository.

### Deliverables

Submit the following in a zip file to the DLE submission point for this module

- A PDF document reporting your work:
  - The design of the UIs (sketches and/or screenshots)
  - Usability evaluation of the project (full details)
  - A link to the GitHub repository for your project
- A video rundown of your application and highlighting the main requirements.

## Deliverables in detail

### GIT Repository

An online repository of your project using the GIT service. You are Graded on the following criteria:

- Repository readme file includes all additional resources (art, sound FX etc.) fully credited
- No previous versions of the project are present in the repository in a .zip or other compressed format
- Commits to the repository are appropriately commented
- Commits are in a consistent timely manner, at least once every week

### A PDF document reporting your work

A PDF to document the full details of the design, implementation and evaluation studies that were carried out including:

- Interface design (high-fidelity prototyping)
- Details of the implementation phase highlighting the choose of the design patterns, worker thread and push notifications.
- Information of user studies, participants who took part in the study, and how they were invited, the consent form.
- How the studies were set and more details on the method/technique that was adopted, evidence of the study setting (i.e. pics or link to a vide).
- Details of how the results could help enhancing the app (i.e. recommendations).

### Video rundown of your application

- Demonstrate the functionality of your project in a short video
- The video should be between 3-5 minutes long
- Please include:
  - Explain any design pattern implementations
  - Go through the code and explain your implementation
  - Demonstrate the program running with data.
- You may be invited to a zoom video call for further discussion at a later date

#### Video settings

<b>File-Type:</b>	MP4
<b>Resolution:</b>	720p or 1080p
<b>Framerate:</b>	30
<b>Video Bitrate:</b>	16 MBS
<b>Audio Bitrate:</b>	Mono – 128 kpbs, Stereo – 384 kpbs
<b>Compression:</b>	H.264

You must present the work carried out in a report submitted in PDF format. No other format is acceptable. Your report must be approximately 2000 words, please use screenshots and links to code files to illustrate functionality where appropriate and high-fidelity prototype to illustrate the design of the interfaces.

The report must contain the following sections:

1. Introduction (approximately 2 paragraphs). Introduce the document and signpost the reader to what they will find in it. Provide links to your GitHub repository, your hosted application and the original dataset.
2. Background. Explain here the scenario, what information your application provides, who are the potential users.
3. Legal, Social, Ethical and Professional (LSEP). Highlight issues might arise in terms of privacy, integrity, security and discuss how you addressed these issues.
4. Design. Present here the design of the app (storyboard and the high- fidelity prototype).
5. Implementation. Illustrate with screenshots and hyperlinks to your source code in your GitHub repo how you implemented your design. Provide a suitable narrative to help your reader understand your screenshots and diagrams.
6. Evaluation. Present the full details of user studies you carried out for evaluating the app with details of participants and methods used with justification why you used any particular method/ technique. Show how you tested your implementation clearly indicating areas for further work and improvement.
7. Summary. Summarize your work briefly and highlight the main points in each section.
8. References. State here all references you used.

The Rubric below will be used to assess and provide feedback on the submissions.

Table 1: Marking scheme for Assessment 2: Assignment Submission

<b>Level descriptors.</b> <i>Note that these definitions are indicative of expected standards at each level, and may not be precise descriptors of the project submitted.</i>							
Category and marks weighting:	<30%	30-39%	40-49%	50-59%	60-69%	70-79%	80-100%
<b>Application (40%)</b> Implementation of interface design is appropriate. Implementation of code demonstrates good quality coding approaches. Application is shown running in the video. Implementation of code uses appropriate design patterns. Clear indication of innovative and proactive thought going beyond materials provided. GitHub commits are appropriate and not left to just before deadline.	Little or no evidence of coding skills in project implementation.	Poor skills in implementing code - incorrect &/or very confused.	Some skill in implementing the software, but with errors &/or confusion. Application provides more functionality than log in and registration.	Skill in most areas of software implementation - some issues/errors. Implementation of moderate complexity with suitable functionality demonstrated.	Competent implementation of software with minor issues/errors. Application is of suitable complexity, has appropriate architecture which is not monolithic but demonstrates interactions between levels and/or layers of software.	Highly skilled implementation of software (far beyond the level of taught modules). Application has good complexity and shows good quality software engineering.	Expert level of skill in all relevant areas clearly evident throughout. Software is of commercial quality and could be implemented in real world situation with very little modification.
	Little or no indication of user requirements having been implemented	User Requirements shown as implemented inconsequential	User Requirements shown as implemented but illustration not clear and lacks clarity	Requirements matched to implementation ok, some errors and omissions	Requirements matched to implementation are appropriate. Meaningful choice of requirements implemented.	Requirements matched to implementation are clearly indicative of minimum viable product	Requirements matched to implementation are of professional standard.
	Little to no indication of any attempt to go beyond the coursework teaching/brief.	Inadequate and poorly defined plan	Plan for innovation vague and/or largely unjustified	Relevant features considered. Accuracy, evidence &/or clarity could improve	Logical consideration given to innovative features, predominantly evidence-based and clearly articulated	Appropriate, well presented and well justified innovations	Clear, concise and fully justified innovation plan. Shows original thinking and proactive development



<b>Design and Testing (40%)</b> Appropriate use of UI components to provide a suitable HCI experience Interface design illustrates clear application of robust usability testing and HCI principles. Application architecture demonstrates suitable use of parallelism Tests have been designed to evaluate application appropriately	Project is devoid of appropriate testing plan	Poor skills in applying testing, incorrect &/or very confused	Some relevant testing applied. V&V superficial, sparse &/or often flawed	Appropriate testing in place but with some omissions, issues &/or errors	Competent testing plan in place. Appropriate Validation and Verification approach in place.	Highly competent testing regime in place both in plan and implementation. Shows a deep understanding of testing above and beyond taught modules.	Expert testing plans and implementations in place, could be appropriate for commercial application with very little modification.
	Little or no indication of requirements	Requirements present but vague and poorly defined. Diagrams not provided.	Requirements present but need more thought and development. Few diagrams present but lack of understanding demonstrated.	Requirements presented ok with appropriate diagrams. Some questionable logic demonstrated. Some errors and omissions	Requirements clear and sensibly defined. Requirements appropriate for complexity of project. Coverage of application is appropriate.	Requirements have good depth of coverage for application. Diagrams are clearly derived from requirements.	Requirements excellent with little further to add. Diagrams match with requirements and could be used in professional setting.
	Images/graphs/figs sparse, illegible &/or irrelevant.	Images/graphs/figs do not convey required information.	Most images/graphs/figs convey req'd info but may lack clarity &/or contain errors.	Mainly appropriate images/graphs/figs - aesthetics &/or labelling could improve.	Most images/graphs/figs of high standard; occasional minor errors/issues.	Images/graphs/figs of high standard, clearly conveying all required information.	Creative images/graphs/figs; peer reviewed journal standard.

<b>Approach (20%)</b> Illustration provided for how HCI and parallelism applied Coding approach outlined How application meets needs for scenario demonstrated. Reflection at an appropriate depth and understanding demonstrated of design patterns and their application.	Writing incomprehensible.	Inappropriate written work	Poor literacy	Mainly appropriate style of writing and presentation - could improve.	Clear style of writing and presentation.	Lucid style of writing. Clear, unambiguous presentation.	Literacy presentation: peer reviewed journal standard.
	Little to no indication of any attempt to go beyond the coursework teaching/brief.	Inadequate and poorly defined plan	Plan for innovation vague and/or largely unjustified	Relevant features considered. Accuracy, evidence &/or clarity could improve	Logical consideration given to innovative features, predominantly evidence-based and clearly articulated	Appropriate, well presented and well justified innovations	Clear, concise and fully justified innovation plan. Shows original thinking and proactive development

## 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/rails/active\\_storage/blobs/proxy/eyJfcmFpbHMiOnsibWVzc2FnZSI6IkJBaHBBNWRiZVE9PSIsImV4cCI6bnVsbCwicHVyIjoiYmxvYl9pZCJ9fQ==--11bdf9753735b78518e41b272aabad67b9f71f32/Extenuating\\_Circumstances\\_Policy\\_and\\_Procedures\\_2024-25%20clean%20copy.pdf](https://www.plymouth.ac.uk/rails/active_storage/blobs/proxy/eyJfcmFpbHMiOnsibWVzc2FnZSI6IkJBaHBBNWRiZVE9PSIsImV4cCI6bnVsbCwicHVyIjoiYmxvYl9pZCJ9fQ==--11bdf9753735b78518e41b272aabad67b9f71f32/Extenuating_Circumstances_Policy_and_Procedures_2024-25%20clean%20copy.pdf)

### 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 person's 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/hc/en-us/categories/21850416398221-Student-hub>

### 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:

<https://aldinhe.ac.uk/product/learnhigher-resources/references-and-bibliographies-booklet/>

Checking your assignments' references:

<https://aldinhe.ac.uk/product/learnhigher-resources/checking-your-assignments-references/>

# Appendix A

## Product Vision

A company wants to develop a mobile application to handle their employees' data through a mobile phone. The app should have two sides admin and employee. The admin to manage employees' data and the employee side to enable employees view and edit their details. The app is supposed to push notifications to inform users (admins and employees) of specific information based on their preferences (e.g. when details get updated). The **admin gets notified when an employee requests a holiday. Employees get notified when their request approved/ declined.** Both users need to login to their account in order to perform the functionalities. Users should be able to turn notifications off. Users can manage preferences in their account. The design of the interfaces should consider different screen's size for different mobile devices.

### The admin side:

For administrators that need to enter their employees' details, the RESTful API web service providing simple and robust access to the data store via the HTTP infrastructure. **Admins also have the capacity to approve/ decline employees' holiday request if they wish to.**

### Functional Requirements

- As an administrator I wish to add employees' details
- As an administrator I wish to check the employees' details have been uploaded.
- As an administrator I wish to edit employees' details.
- As an administrator I wish to delete an employee's details.
- The app should also apply an automatic increment by 5% on the employee salary when they complete one year.

### The employee side:

#### Functional Requirements

- As an employee I wish to view my details.
- As an employee I wish to edit my details.
- As an employee I wish to book/ manage my annual leave/ holiday (the allowance is 30 days per year).