

Faculty of Science and Engineering

Referred Coursework – 2023/24 Academic Year

PLEASE NOTE: If you have been referred in the **COURSEWORK** element of this module and are required to be reassessed by **COURSEWORK**, please complete this referred work.

Module Code: COMP2000HK

Module Title: Software Engineering 2

Module Leader: Dr Alaa Alkhafaji

School: Engineering, Computing and Mathematics

DEADLINE FOR SUBMISSION: 22nd August 2024 at 3PM BST/22:00 HKT

SUBMISSION INSTRUCTIONS FOR CANDIDATES

Referred coursework must be submitted electronically using the online submission facility in the DLE by the deadline above.

When you submit your assessment you are stating that it is your own work. Please ensure you are familiar with the relevant referencing guidelines and our procedures by familiarising yourself with the **Assessment Offences and Research Misconduct Procedure**. This procedure can be found in the [Academic Regulations](#).

If you have any queries on submission or in relation to the referred work, please contact the Module Leader in the first instance, if they are unavailable please contact the Faculty Office on 01752 584584 immediately so any problems can be rectified.

PLEASE NOTE that we cannot accept work submitted via email.

Instructions to Candidates

This is an individual piece of work.

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 referral assessment of *COMP2000HK software engineering 2*. The module is assessed via **100% coursework**.

The sections that follow will detail the assessment tasks that are to be undertaken. All assessments are to be submitted electronically via the respective DLE module pages before the stated deadlines.

Task:

You are required to produce an Android mobile application that meets the requirement specification in appendix A of this assignment (design and implementation). Your implementation MUST be in Java programming language using Android Studio. The Java project will provide a graphical user interface (GUI) for usage. The GUI should provide the functionality listed in appendix A. The API for interacting with the database is provided at this link:

(<https://web.socem.plymouth.ac.uk/COMP2000/ReferralApi/api/Bookings>).

- You must use a worker thread to handle the API connection.
- You are required to use appropriate design patterns that you learnt throughout the module.
- This application should follow the HCI guidance in designing the UIs (as defined in lectures).
- Analyse the scenario in appendix A to identify the context of use and the users.
- Produce a UIs design for a client-side mobile application (see Appendix A for more details).
- Carry out a usability heuristic evaluation (discuss your design in terms of the 10 usability heuristics).
- Summarise the outcome from the evaluation study. (**No participants are required to be involved in this stage and it's out of the scope of this assessment**)
- You MUST use Git version control with GitHub in your development process.
- You are to use the GitHub Classroom set up for this module – the link for signing up is here (<https://classroom.github.com/a/v1LbE5Az>). You will be assessed on appropriate use of version control.

Assessment Criteria:

Individual Coursework comprising one assignment with a GIT repository of incremental development leading to a final build or release. Repository includes itemised deliverables for final module submission. Typically, this will include your Android project and the 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 a **single PDF document** reporting your work to the DLE:

- The design of the UIs (sketches/ screenshots).
- System architecture (diagrams with suitable narratives).
- Usability Heuristic evaluation of the project (i.e. discuss the design of your project in terms of the 10 usability heuristics).
- A link to the GitHub repository for your project. **It's the responsibility of the students to ensure that their repository is accessible.**
- Screenshots of the working system (i.e. the implementation)
- A link to the YouTube video rundown of your application. **It's the responsibility of the students to ensure that their video is accessible.**

Deliverables in detail

GIT Repository

An online repository of your project using the GIT service. You will be assessed on the following criteria:

- 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 document contains details of the design, implementation and evaluation studies that were carried out including:

- Interface design (sketches/ screenshots). The design should follow the HCI principles that you learnt during the module. You might consider different screens' size for different mobile devices.
- System architecture with diagrams and suitable narratives.

- Details of the implementation phase highlighting the design choices including design patterns, worker thread and push notifications. Go through the code and explain your implementation briefly.
- The testing and evaluation of your project. Discuss the design of your project in terms of the 10 usability heuristics. Highlight the strength and weakness in this regard. Test your implementation and functionality of your system using a testing method of your choice. Summarise the outcome of this stage and clearly indicate area of future improvement.

Video rundown of your application

- Demonstrate the functionality of your project running with actual data and user interaction in a short video.
- The video should be between THREE minutes long in the normal pace. **NOTE: the marker will stop the video after THREE minutes.**
- No need for flashy intros please!
- You may be invited to an interview (either online or in-person) for further discussion at a later date.

Video settings

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

You must present the work carried out in a report submitted in PDF format; **no any other format is accepted.** Your report must be approximately 2000 words, please use screenshots and links to code to illustrate functionality where appropriate.

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 and the video rundown.
2. Background. Explain here what information your application provides, who are the potential users and in which context will be used.
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 and the system architecture (i.e. sketches/ screenshots, storyboard and diagrams) with suitable narratives.
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. Testing and evaluation. Discuss your UIs design in terms of usability heuristics and highlight the strength and weaknesses in your design. Show how you tested your implementation and functionality clearly indicating areas for further work and improvement.
7. Summary. Summarise here your work briefly and highlight the main points in each section.

The Rubric below on page 7 will be used to assess and provide feedback on the submissions.

Level descriptors. <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%
Application (40%) 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 or no indication of defining a plan	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.

Design and Testing (40%) 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.	Little or no indication of applying testing.	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.

Approach (20%) 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 or no indication of defining a plan	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

Table 2: Feedback Template for Assessment 1

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/22/22876/Extenuating_Circumstances_Policy_and_Procedures.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 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: https://www-citethemrightonline-com.plymouth.idm.oclc.org/](https://www-citethemrightonline-com.plymouth.idm.oclc.org/); this 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-assignments-references/>

Appendix A

Product vision:

A tennis club would like to develop a mobile app to enable their members to handle courts' booking. The club has ten courts in total, four artificial grass, two hard courts and four grass. The grass courts are only available over summer; June, July and August. Members should login to their account in order to book a court, which should not be more than 48 hours in advance. The app push notification to notify members of any changes in the booking such as if the court booking was cancelled for any reason e.g. heavy rain. Members could turn notification on/ off when needed. Members can book one court at a time.

Functional requirements

- As a member I wish to create an account.
- As a member I wish to view courts and their type and availability.
- As a member I wish to book a court.
- As a member I wish to manage my bookings (edit/ cancel).