SCCT
ACADEMIC YEAR 2023/24

Assessment Brief

Submission and feedback dates

Submission deadline: **Before 14:00 on 11th January 2024**

Is eligible for 48 hour late submission window

Marks and Feedback due on: 12th February 2024

N.B. all times are 24-hour clock, current local time (at time of submission) in the UK

Submission details

Module title and code: UFCF7H-15-3 Mobile Applications

Assessment type: Coursework

Assessment title: Group project development of a mobile application

Assessment weighting: 75% of total module mark

Module learning outcomes assessed by this task:

1. Analyse and critically evaluate mobile platform technologies for the development of mobile applications
2. Interpret user expectations and apply these in the context of mobile applications
3. Design, develop, test, and document a working application for a mobile device.
4. Consider current and emerging trends in mobile device technology and have regard to commercial licensing frameworks for mobile development.

Table of Contents

Section 1: What am I required to do on this assessment?

You will design and then create your own mobile application using Android Studio and Kotlin through a group work.

Section 1.1 Task specification

When designing your app, you must consider the UX of the application whilst implementing the concepts of functional programming, control, and data structures, and developing your software design skills by organising and formatting your code for readability by implementing design patterns and industry coding standards.

Overall, you should work in a group to develop your **OWN** mobile app from design to implementation.

Stage 1: Forming a group (no marking)

This is a group assignment which should be finished by up to **three** students. We recommend that each group consists of three members, and a two-member group is also allowed if you cannot find a suitable third member. However, this assignment is assessed by its creativity, complexity, and functionality, and usually the group with more members may produce a better-quality mobile application.

A group with a single member is not allowed. You need to contact the module leader to discuss it if the special adjustment is needed.

Please email the module leader (kun.wei@uwe.ac.uk) about your group members by **30th Oct 2023**, and the group information is also published and updated in BB->assignment timely. Note that the module leader will group the ungrounded students randomly after the deadline.

Tips for the group assignment:

1. Schedule a meeting regularly
2. Split the workload equally
3. Create a project plan and track the project progress
4. Build trust and collaboration

Forming a group involves effective communication, clear role assignments, and collaboration. By following these steps, you can establish a productive and cohesive project team.

Stage 2: Designing your app (35%)

All great apps start with an innovative idea and smart design. Jumping straight into development without considering what you are creating, who you are creating it for, or how to go about building it are guaranteed routes to failure. Research is key as you need to provide evidence for the decisions you are making.

The design documentation of your app will be assessed based on the criteria detailed in the grading criteria section. It is expected there will be significant variety between student submissions as app ideas and research directions will vary between groups based on the apps you are building.

Students are expected to produce an up to 10 pages design document (although this is not a strict limit) that could be considered as "client facing". This represents the type of documentation you would produce for a client and would be completed before any actual development of an app would commence.

A basic template will be provided in Blackboard which you are required to edit to make your own assessed by the following criteria:

- **REQUIREMENTS (10%):** Why is your app concept significant? Who is the target audience? Explore existing apps on GooglePlay, evaluating good practice or identifying gaps in the market. Your concept can be based on an existing application but consider how you can improve upon it in your concept. Gather details about what requirements are needed to achieve your goals, such as sourcing APIs, external libraries, asset collections and more. Think technically about what you need to create your final app.
- **WIREFRAMES (10%):** Wireframes form the middle ground between low-fidelity sketches and first composite prototypes, but the processes can vary drastically. They allow you to visualise your ideas and consider the UX whilst giving you clear goals of what is required to build a functioning app. Wireframes should include consideration for the flow of your app (with scale, layout and supported devices considered).

- **COMPOSITES (15%):** When you know how your app will be laid out and flow you will want to give it a custom look and feel. Focused on design and using the wireframes as a guide, you can use any tools you like such as Sketch, Figma or any software that allows you to manipulate images that can be used as final assets. You are not required to use interactive prototyping at this stage. Concept designs, and style guides such as colour palettes, UX/UI design considerations and app icon generation are all important. Use of sourced images is fine for this assignment submission, but these final image assets should be useable in the final development of your application and a final display of your app 'in context' is required.

All documentation should be produced professionally as if you were to present it to a client. Each member of a group should submit the same documentation individually, and the same mark will be given to all members.

Stage 3: Building your app (60%)

Following the design of your idea you will now be required to make the app. It is expected that the original concepts from the design stage may evolve and potentially differ from the original design as you build and test your application, but it should always be clear why any deviations occur, and the final app must remain true to your original concept.

Work collaboratively with your group members or even peers and share good practice. Share resources you find during your research to advance your skills in app development. Your final application will be better as a result.

The development of your application will be assessed **collectively** and **individually**. We recommend that the application design consists of relatively independent views (e.g., each view is a subsystem) so that each member can work on a view independently. Or you can also split the functionalities into multiple work packages.

Each group should produce **a mini report** here to clearly claim each member's contribution, which must be agreed and signed by all members. For example, Student A is in charge of the login system, and the loginActivity.mk file is finished by A. In addition, the mini report should include the environment configuration for the app such as the SDK version and the virtual device for emulation.

Note that this mini report must be included in the submission, and the marker will use it to mark each member's code individually.

The mark for this stage consists of **the group mark (20% of the 60%)** and **the individual mark (40% of the 60%).** Therefore, each group member's mark is likely to be different, and it is partially affected by the group performance, but is dominantly decided by the individual performance.

The group mark can be claimed only if you have contributed to the coding of the app. For example, if you haven't done any coding, you might get zero mark for Stage 3. But if you've done a minimal contribution to the coding, you might get a low mark for the individual assessment but may get a decent mark from the group assessment if others have done a good job.

A successful group work can not only increase the group performance, but also promote individual's accomplishment.

The detailed marking criteria can be found in **Section 3**, and in brief the assessment focus on the following facets.

- **CODE STUCTURE**: Your code should be human readable. Variables and functions should have suitable names and code should be logical in its structure. Clarity at the point of use is the goal and is more important than brevity. Although Kotlin can be compact, this is not the goal for this assessment. Your code can be compact, but it is expected you will adhere to suitable frameworks where appropriate. The Functional Programming concepts should be followed and data structures, where applicable, should be suitable for the type of app being developed.
- **FLOW AND NAVIGATION**: Every application that consists of multiple views needs some form of navigation to enable users to move between different screens and to display information whilst reacting to events and user inputs. Here we will assess the flow and user experience of your application, as well as how responsive and clear it is to use.
- **FUNCTIONALITY**: How does the application function while in use? Are there missing features or elements not included which are considered standardised in modern applications? Referring to the original designs, how much of the planned functionality is present? Is the application built for one device or orientation? Are constraints and layout for multiple screen sizes catered for?
- **COMPLEXTITY**: Has effort been applied to push your own skills in mobile app development or is it a basic, easy to produce app? Does your code include only simple structures and communications between different views, or does it explore the full potential beyond readily available tutorials? This is your chance to try out some exciting technologies!

IMPORTANT: Use of online tutorials and external resources for your app development is expected, but you should always comment sourced code samples where appropriate. Not doing so can affect the grade you can achieve, but also it could be considered plagiarism.

Stage 4: Demonstrating your app (5%)

The group is required to produce a video (less than 8 mins) to demonstrate your app's functionalities. Every group member **MUST** contribute to the video to demo and explain your own code, and otherwise you may lose the mark for this stage. Ideally, a high-quality demo video should start from the overall implementation of the app about what the app can do, and then each member's code should be demonstrated with an explanation in sequence.

The video can be made with any tools you like, and you can also check BB for the recommended tools. Please pay attention to the video size because a large video file may take a long time to be uploaded in BB or even be blocked. If the video size is too big, you can store the video in OneDrive, and insert the link in the mini report.

Section 1.2 Deliverables

One folder in zip format (only) must be uploaded via the relevant link on the module's space on Blackboard. The zipped file should be names as 'MACoursework_GroupID_StudentNumber.zip'.

The link will be available two weeks before the due date and will be communicated to students via an email announcement.

The submission must contain:

- A word/PDF file (app design) for Stage 2.
- A mini report (word/PDF) for Stage 3
- An Android Studio project (including all the code and related resources) for Stage 3
- A video file (or multiple video files) for Stage 4 or a link for the video is included in the mini report if the video size is big.

The code must be finished in Android Studio with Kotlin. The app developed in iOS/Apple or other cross-platform tools is NOT accepted and will NOT be marked.

You should do the submission individually even if the submitted work might be same for all group members. For example, if one group member does not finish the coding unfortunately, this should not affect other group members' submission.

Section 2: Where should I start?

You should start to define your app idea via the group discussion and the research on GooglePlay and App Store. You need to consider that what problem does it solve, who is your target audience, what features will it have, and ensure that your idea is well-defined and addresses a specific need.

Our teaching materials for the app design will provide the methodologies and the templates step by step. With the teaching progress, you should consider how this material can be used on your own app.

From the implementation, the first four weeks' learning should provide you enough programming skills to a simple Android app. However, for an app with advanced features such as camera, Google map and so on, you should explore it together with the group members.

Section 3: What do I need to do to pass? (Marking Criteria)

To pass the coursework, you must get 40% or above. Please check the following tables of the marking criteria for Stage 2, 3 and 4.

STAGE 2: GRADING CRITERIA for DESIGNING YOUR APP (35%)

Below are relevant qualitative descriptors for mark boundaries in the design stage. These are purposely broad in their descriptions, as the body of work requires you to explore content outside of the lecture and tutorial materials.

Fail 0 > 39%	Little or no evidence of research and requirements. Wireframes are little more than the initial ideas and are non-workable, only detailing elements of an app idea but not explaining how it will function or flow. Where external information is included, there is no evaluative consideration. Documentation requires significant work before it could be used, and the end user is never or rarely considered in terms of design interaction. No inclusion of any specific requirements is provided, and any composites are not related to wireframes or the final application, or they are just screenshots of the final build.
40 > 49%	Limited understanding of available apps in the marketplace is demonstrated. Wireframes show a basic layout of the concept, perhaps missing several notable features and/or is structured poorly. Documentation requires significant effort to make sense of some of the concepts presented. Requirements are included but are little more than a list of external tools or standard interface features. The design seems unrealistic or does not consider all aspects identified in the requirements. The wireframes and/or composites are clearly reversed engineered, rather than considered from the start. The presentation of the document is readable, but information is ordered poorly.
50 > 59%	There is a developing understanding of how the app marketplace functions. Design concepts can be understood from what has been produced and graphics have been used to show the completed idea using logical wireframes that explain well the flow of the app. Composites are clear and represent what the final app concept could look like, and effort has been made towards demonstrating what user experience will be, although this may not be entirely clear to the reader. With a little work this will make a great app. Presentation of the documentation is clear and suitable for a client, but information can be confusing at times or seem tagged on in places.
60 > 69%	Documentation is professionally produced covering all aspects of the app design and technical requirements. Whilst not always suitable, the documentation has demonstrated you can plan to produce the app concept fully. The presentation is smart and really helps the reader understand the concept from how it will work, to how it will be produced and what is needed in the development of the application. There are still production questions, but generally everything is understood from a consumer/client perspective and a developer's requirements. Significant effort has been made to show a near final app in your design. You should make this app!!
70% +	Amazing! Let's build this app! All documentation contains surprising results and demonstrates professional standards across all stages of the design phase. Your concept is grounded in solid research and there is little more a professional company or client would require moving into development of the application.

STAGE 3: GRADING CRITERIA for BUILDING YOUR APP (60%)

Fail **0 > 39%**	Code is unreadable or clearly the result of cutting and pasting sourced samples into your project. Platform guidelines seem to have been mostly ignored. The app's layout constraints have been disregarded and there is little code beyond what can be easily sourced in online tutorials. The app's flow is jilted with the user able to get stuck with no way to navigate views or there is little to no navigation or interface elements are nonresponsive. The ideas presented in the design stage are not present or do not work as intended and the overall app is little more than a collection of ideas poorly implemented.
40 > 49%	The code is of a poor standard, ignoring modern techniques and app development frameworks. The app layout functions on a selected device but there are issues with other devices and/or orientations. The final applications source code does not push the expectations of modern app development, or it sticks to readily available online resources. Flow between views is present but only to allows the user to move between screens with little consideration of how information is presented back to a user. Ideas from the design stage have made their way into the final implementation, although not fully or the final application is little more than a collection of ideas put together with no clear direction.
50 > 59%	App complexity is becoming sophisticated, and effort has been made to explore what is possible with the tools available. The code is sensibly structured and there is evidence of good software design and use of an appropriate framework with sensible use of naming conventions. If external APIs are used, or local data is included, it is formatted logically with view to scaling the application. Views scale generally well, and work between most or all mobile devices in the expected orientation, although issues still exist if pushed. Most of the ideas from the design stage are present or, effort has been made to include them, but perhaps they are limited in terms of functionality. There are apps of this standard available on the various marketplaces, but they have...mixed reviews.
60 > 69%	This app feels complete. The user experience is smooth and at no point does the user feel unsure how to progress or navigate the interface. The complexity is high and works in parallel with the flow of the app, such as external API requests not breaking other functionality. Animations are used to add flourish to the design and added with contemplation. Overall, the app is professionally produced matching what was originally documented in the design stage or at times expanding upon it. Some concerns linger around code structure and/or the user can still identify some features that are lacking, or that they would like to see included, but this is a great app!
70% +	Your app submission is excellent, and you should consider marketing your app and upload it to the app store! This is a professionally produced product that either fills a gap in the marketplace or does what existing apps do...but better. All assessment criteria are met to a high standard.

STAGE 4: GRADING CRITERIA for DEMONSTRATION (5%)

Fail **0 > 39%**	Video demo is none or minimal, or the video demonstrates up to 40% of the functionalities of the app.
40 > 49%	Video demonstrates up to 60% of the functionalities of the app.
50 > 59%	Video demonstrates up to 80% of the functionalities of the app.
60 > 69%	Video demonstrates up to all the functionalities of the app.
70% +	Clarification is clear and concise, and main features are emphasised. Or even very professional.

Please read the marking criteria for the pass boundary carefully and discuss it with the lab tutors or the module leader if you have any questions.

Section 4: How do I achieve high marks in this assessment?

Achieving high marks in an assessment requires a combination of effective study strategies, time management, and a deep understanding of the subject matter. Here are some tips to help you excel in your assessment:

1. **Understand the Assessment Criteria**: carefully review the assessment guidelines and criteria provided by your instructor or syllabus. Understand what is expected in terms of content, format, and grading criteria.
2. **Create a Study Plan**: develop a structured study plan that outlines what topics you need to cover, how much time to allocate to each, and when to start studying. Prioritize difficult or unfamiliar topics.
3. **Take Effective Notes**: when attending lectures or studying from textbooks, take organized and concise notes.
4. **Practice Active Learning**: engage actively with the material by asking questions, making connections, and summarizing key concepts in your own words. This helps deepen your understanding.
5. **Seek Clarification**: if you have questions or encounter challenging topics, don't hesitate to seek clarification from your instructor, classmates, or online resources. Understanding difficult concepts is crucial.
6. **Manage Your Time**: allocate sufficient time for studying and avoid last-minute cramming. Regular, spaced study sessions are more effective than long, stressful cramming sessions.
7. **Study Groups**: discussing and teaching concepts to others can enhance your understanding.

Section 5: How does the learning and teaching relate to the assessment?

The lecture slides, lab exercises and related tutorials from Week 1 to Week 8 are sufficient for finishing the coursework well. Our learning and teaching will direct you to become an Android app beginner from a novice step by step. Therefore, engaging the lectures and lab sessions is important for achieving decent marks. For example, Week 5, 6 and 7 will teach how to design your app, which is for Stage 2.

Section 6: What additional resources may help me complete this assessment?

There are a number of resources in BB to help you finish the assessment. But, a self-directed study and research and the close group work are also crucial. Please take advantage of the lab sessions and you can get prompt and in-person feedback from the lab tutor if you present some of your work.

You can also make the appointment with the module leader on Monday (11:00 – 13:00) to discuss any issues about your coursework.

The frequently asked questions will be published on BB if the module leader thinks it's necessary to announce to all.

For the general study skill, you can refer to UWE library page https://www.uwe.ac.uk/study/study-support/study-skills

Section 7: What do I do if I am concerned about completing this assessment?

UWE Bristol offer a range of Assessment Support Options that you can explore through this link, and both Academic Support and Wellbeing Support are available.

For further information, please see the Academic Survival Guide.

Section 8: Marks and Feedback

Your assessment will be marked according to the marking criteria represented in Section 3. The formal feedback will be returned to the students with the mark within four weeks after the submission. Please check the exact date from the cover page.

You can also contact the markers or the module leader about the clarification of the feedback and the mark when they are published.

There are more general advice before the coursework submission.

1. In line with UWE Bristol's Assessment Content Limit Policy (formerly the Word Count Policy), word count includes all text, including (but not limited to): the main body of text (including headings), all citations (both in and out of brackets), text boxes, tables and graphs, figures and diagrams, quotes, lists.
2. UWE Bristol's UWE's Assessment Offences Policy requires that you submit work that is entirely your own and reflects your own learning, so it is important to:
    - Ensure you reference all sources used, using the UWE Harvard/OSCOLA system and the guidance available on UWE's Study Skills referencing pages.
    - Avoid copying and pasting any work into this assessment, including your own previous assessments, work from other students or internet sources
    - Develop your own style, arguments and wording, so avoid copying sources and changing individual words but keeping, essentially, the same sentences and/or structures from other sources
    - Never give your work to others who may copy it

- If an individual assessment, develop your own work and preparation, and do not allow anyone to make amends on your work (including proof-readers, who may highlight issues but not edit the work) and

When submitting your work, you will be required to confirm that the work is your own, and text-matching software and other methods are routinely used to check submissions against other submissions to the university and internet sources. Details of what constitutes plagiarism and how to avoid it can be found on UWE's Study Skills pages about avoiding plagiarism.