CSC306: Writing Mobile Apps:
Assignment Specification

The assignments for *CSC306: Writing Mobile Apps* require you to design and implement an Android application, following the specification in this document. The learning outcomes of the assignments build upon the module outcomes and additionally present you with the opportunity to enhance your skills in mobile application development and explore techniques that go beyond the material covered in lectures.

The purpose of Assignment 1 is to encourage you to consider some of the design principles required before embarking upon the implementation of a Mobile Application.

Assignment 2 will see you review your peers' designs and XML layouts and assess them.

The purpose of Assignment 3 is to develop an Android application building upon lecture examples and enhance your own Android (and Kotlin) development skills.

Component	**Task**	**Weighting**	**Deadline**
Assignment 1	Design	30%	7th November
Assignment 2	Review	10%	15th November
Assignment 3	Implementation and Demo	60%	9th January

Assignment Guidance

- Start early in your work, the earlier you start the more prepared you will be for submission. You don't want to be rushing things near the deadline.
- If you encounter a problem, try to solve this yourself first (e.g., Google has been shown to be useful). Lab demonstrators will want you to have tried to solve a problem before simply asking for help at the first hurdle.
- Use the Canvas coursework discussion page to ask questions about the assignment, you can help each other and discuss approaches or queries that you might have. It's a place where general queries will be answered, rather than several private emails.
- Make use of the lab time, you can get guidance there on your design and implementation, along with feedback on the approach you are taking.
- Pay close attention to the marking scheme. Use the rubric to mark your own work and see how you can improve using the next tier of the rubric.
- Enjoy yourself, have some fun creating a mobile app and thinking of features you would like to see if this was an app you were to use.

Academic Integrity

By submitting coursework, electronically and/or hardcopy, you state that you fully understand and are complying with the University?s policy on Academic Integrity and Academic Misconduct. The policy can be found at `https://myuni.swansea.ac.uk/academic-life/academic-misconduct`. The consequences of committing academic misconduct can be extremely serious and may have a profound effect on your results and/or further progression. The penalties range from a written reprimand to cancellation of all of your marks and withdrawal from the University.

1 App Specification

The assignments for CSC306 will require you to design and implement a news aggregator app on the Android platform. In the app users will be able to personalise the type of news that they are interested in and receive alerts when new stories become available. Your application should allow users to have an account they can enter key terms that they are interested in, with the app fetching news from multiple sources using a service such as `https://newsapi.org/` (*be sure to check daily usage limits*) to pull news data and be able to present key information to a user in an attractive way. Key information to include would be news publisher, article title, article image and summary text. The app should also be able to setup alerts that can be pushed to a user when stories they are interested in become available. Within the app, the user should configure which topics they would like to receive alerts for.

Assignment 1 - Design

In the first part of the assessment you will specify the application you will build, propose bonus features and present GUI screens that you have prepared. **You should complete this step before moving onto any implementation (i.e. Kotlin programming) tasks**.

1.1 Task

In the first part of the assignment, worth **30%** of the module assessment, you should produce a report of no more than **8 pages** that sets out key design decisions and example GUI elements for your proposed application.

In your report, you should flesh out the details of the specification given in Section 1, which is broad and leaves scope for several finer decisions to be made. By the end of your report, there should be a clear foundation to build upon for your later implementation task.

Additionally, you should consider what features should be added to enhance the appeal of the application. There is no specific guidance or criteria on what you should include here, but be creative in the capabilities of the Android platform when considering your features. You should also consider what will enhance the usability / appeal of the application. These features should be substantial, and not simple aesthetic additions.

The mark for bonus features will be calculated as the product of a base mark and a richness factor. For example, if your chosen features are very basic then this will attract a low richness factor, say 0.6. If you now scored a base mark of 30 (out of 35), then the mark would be (30 / 35) * 0.6 * 35 = 18. If your features are more complex then this will attract a richness factor of 1.0 resulting in an overall mark that would be (30 / 35) * 1.0 * 35 = 30.

Finally, you are required to produce examples of activity screens that you have prepared. At this stage, it is likely the case that you may need to populate layouts with *'dummy'* data, as your app should have no implementation at this stage.

1.2 Marking

The overall mark will be calculated as the product of a base mark and a richness factor as indicated above. The base mark will be awarded based on the following rubric:

1. **Design choices - 20%** - The specification you have been given is brief, and there is plenty of scope for more clarification on how the app will work. Some questions (this list is not comprehensive, you should add to these questions) you might want to consider include:

   - How will a user enter their preferred news topics?
   - How will users be alerted to new stories being posted that they are interested in?
   - What information can be hidden from news stories to provide a cleaner user interface, and how would you do this?

   Clear justification for you design choices based on external evidence will attract higher marks in the marking rubric.

2. **Bonus features** - **40%** - You are expected to design a minimum of 3 bonus features to your app. Your submission should clearly identify what these are. Marks will be allocated here based on the added value provided by the bonus features which you offer. Are they just some additional decorations or do they actually make the application more interesting to use, or more rewarding?

3. **Activity GUIs- 40%** - You are required to submit XML layouts of **3** of your in-progress activities (i.e. 3 different screens that exist in your proposed app) that **must be created within Android Studio**. In your report you should reference these layouts and clarify aspects of your design expressed in your design decisions. Think here of adhering to Design Guidelines discussed in lectures. You should strive to create a look that suits the purpose of the app, and have a consistent look-and-feel throughout. For this part of the assignment, you may have to use dummy data

(as there may be no implementation at this stage) which is suitable for this section. **The focus here on your design ideas, rather than a functioning app at this stage.** The XML layouts you submit will be used as part of the Coursework 2 code review.

1.3 Submission

You should submit a **single PDF document** containing no more than 8 pages and a **single zip folder** containing your XML layouts to the CSC306 Canvas module page before **11:00am, 7th November 2022**.

Any content beyond 8 pages will not be marked as it does not follow the submission instructions.

2 Assignment 2 - Code Review

An important part of working in a software engineering project is code reviews (which now feature in many software methodologies). In this process designers and developers look at each other's code to spot bugs and ensure standards are being maintained. As part of Assignment 1 you must submit 3 XML layouts and for Assignment 2 you will be assigned to review and mark 5 other student's work (within one week). The quality of the reviews you provide will be marked using the marking rubric given at the end of this document.

3 Assignment 3 - Implementation

Assignment 3 should follow directly on from the work you complete in Assignment 1 and the feedback you have received from the marker, and your peers in Assignment 2. The implementation section should largely follow the decisions you made in your design document, unless you received feedback advising a change of direction. You should target an emulator device that uses **API level 28**, this is what the marker will be using when assessing your submission, potentially alongside an actual Android device running the same API level.

You will likely be using an external API / resource component to retrieve information from a news source. In addition to this, **your application development should contain at least one other external API / resource**.

Your submission will be formed of an app APK file as well as your source code, structured as indicated in 4.3 below. This is necessary since the marker will check that your app compiles and runs as indicated in your demonstration video and to inspect the quality of your coding.

3.1 Task

Your task for Assignment 3 is to fully implement the application you proposed in Assignment 1, including 2 of your bonus features (it is possible to change your features based on feedback from Assignment 1 / your own increased knowledge and experience in Android development). The app you submit should conform to the outline specification given in Section 1 and follow the fundamental elements of the application.

3.1.1 Video Demonstration

You are required to create a video to demonstrate the features of your mobile app. If your app were to be published on the Play Store, it would usually have a trailer showing key features to encourage people to download the app. This should be seen as a good approach for how you structure your video. Use this video as an opportunity to demonstrate all of the features described in this document that you have implemented, along with the bonus features you have added.

Your video should clearly highlight the bonus features that you have implemented.

For your video, you could consider a structure of spending the **first 2 minutes demonstrating the core functionality of the app, with the final minute demonstrating your bonus features**. You should **clearly indicate** when you are demonstrating your bonus features (with text on screen for example). **Your video submission will be used by the marker as evidence that your app meets the specifications as well as the design that your proposed**.

Several links to (free) screen capturing software can be found on pages such as ScreenRec or by Googling for a solution that meets your needs.

Your file should be named based on your student number, e.g.:

*STUDENT_NUMBER*_applicationDemo.mpg.

The movie files are saved in MPEG or MP4 format. You may use as many screen capture files as necessary to capture the features of your application. **Only submit 1 video that show all features**. Your animated screen captures are placed in a folder called demo that resides in root directory of your .zip folder.

Your video should not exceed 3 minutes in length, the marker will not view any content that is beyond the 3 minute mark of a video.

3.1.2 Self-assessment and Reflection

You are required to answer several short questions to show that you understand fundamental questions about the material delivered as part of this course. This will require you to identify where in your code you have evidenced good practice and that you understand the material taught on the module.

3.2 Final Marking

1. **Functionality** - You will be marked here on how complete of an app you submit that meets the specification detailed above, in that a user will be able to use the application fully in its basic form - **35%**
2. **Bonus Features** - For this category, you will be marked on how complete and rich the implementation of your extra features proposed in Part 1 are. You can add to or extend the features you proposed in Part 1 here - **15%**
3. **Turnitin Questions** - A series of questions that will assess your application and understanding of content delivered on the module - **50%**

3.3 Final Submission

You should submit a **single .zip file** which conforms the following structure:

- all of your project .kt and .xml in a directory named **ProjectSource**
- A project .apk file in directory named **ProjectAPK**
- A (maximum) 3 minute long video demonstrating your application in the root directory of your submission
- A readme file with username and password credentials for your app, and a brief explanation of the changes in the design and content of your app.

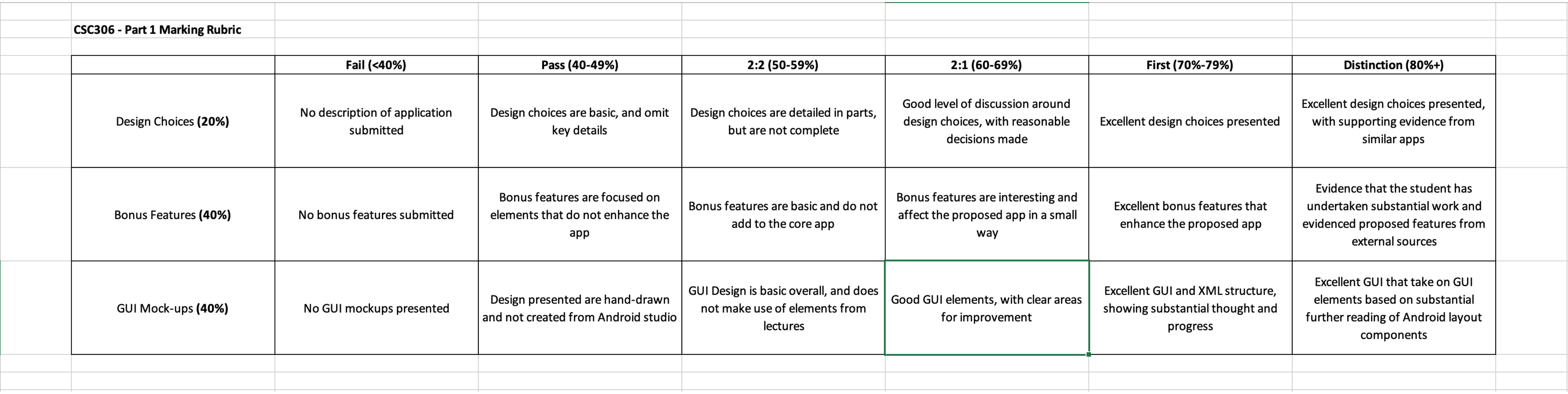
to Canvas before **11:00am 9th January 2021**. Failing to submit all of the files will lose you marks as the marker won't be able to test all features of your app. Your .zip folder should be named:

*STUDENT_NUMBER_*.zip.

This is an **individual** assignment and must be completed by you alone. If you copy files or methods from another source these must be referenced in comments in the usual way.

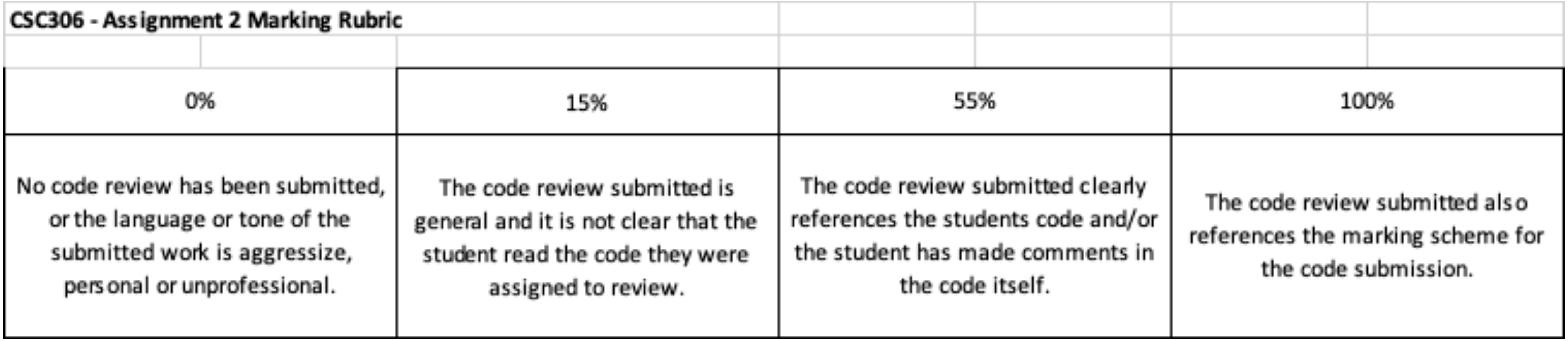
4 Marking

Marking for all assignments will follow the attached marking rubrics. Use these rubrics to mark your own work as you approach completion, reflecting on the marking criteria to help you make the best final version of your app as possible.



CSC306 - Part 1 Marking Rubric

	Fail (<40%)	Pass (40-49%)	2:2 (50-59%)	2:1 (60-69%)	First (70%-79%)	Distinction (80%+)
Design Choices **(20%)**	No description of application submitted	Design choices are basic, and omit key details	Design choices are detailed in parts, but are not complete	Good level of discussion around design choices, with reasonable decisions made	Excellent design choices presented	Excellent design choices presented, with supporting evidence from similar apps
Bonus Features **(40%)**	No bonus features submitted	Bonus features are focused on elements that do not enhance the app	Bonus features are basic and do not add to the core app	Bonus features are interesting and affect the proposed app in a small way	Excellent bonus features that enhance the proposed app	Evidence that the student has undertaken substantial work and evidenced proposed features from external sources
GUI Mock-ups **(40%)**	No GUI mockups presented	Design presented are hand-drawn and not created from Android studio	GUI Design is basic overall, and does not make use of elements from lectures	Good GUI elements, with clear areas for improvement	Excellent GUI and XML structure, showing substantial thought and progress	Excellent GUI that take on GUI elements based on substantial further reading of Android layout components



CSC306 - Assignment 2 Marking Rubric

0%	15%	55%	100%
No code review has been submitted, or the language or tone of the submitted work is aggressize, personal or unprofessional.	The code review submitted is general and it is not clear that the student read the code they were assigned to review.	The code review submitted clearly references the students code and/or the student has made comments in the code itself.	The code review submitted also references the marking scheme for the code submission.

CSC306 - Assignment 3 Marking Rubric

	Fail (<40%)	Pass (40-49%)	2:2 (50-59%)	2:1 (60-69%)	First (70%-79%)	Distinction (80%+)
Functionality **(35%)**	Work of very poor quality. Doesn't run, or runs but doesn't have any functionality.	The app is limited in functionality and has clear errors	The app has limited functionality and does not meet all requirements	The app is of good quality, but some key elements are missing	All features described in Coursework Outline have been implemented	Excellent quality of functionality
Bonus Features **(15%)**	No bonus features implemented	Bonus features are of a low quality and do not enhance the app beyond the base functionality	Bonus features are of reasonable quality, but there are errors in use	Bonus features are of good quality, though there is room for improvements in quality	Excellent quality bonus features	Excellent quality of Bonus features with a high number of features added
Module Knowledge **(30%)**	No answers to questions	Questions answered to a poor standard	Simple answers given that demonstrate a base level of understanding of module content	Good answers to questions, though more technical detail could have been given	Excellent answers to questions	Answers demonstrate a professional approach to development and knowledge from the student that exceeds taught content