|  |  |  |
| --- | --- | --- |
| COMP1786 (2024/25) | **Mobile Application Design and Development** | **Contribution: 80% of the course** |
| **Course Leader: Dr Tuan Nguyen** | **Practical Coursework 1** | **Deadline Date:** |

|  |
| --- |
| **Plagiarism is presenting somebody else's work as your own. It includes: copying information directly from the Web or books without referencing the material; submitting joint coursework as an individual effort; copying another student's coursework; stealing coursework from another student and submitting it as your own work.  Suspected plagiarism will be investigated and if found to have occurred will be dealt with according to the procedures set down by the University. Please see your student handbook for further details of what is / isn't plagiarism.** All material copied or amended from any source (e.g. internet, books) must be referenced correctly according to the reference style you are using.  Your work will be submitted for plagiarism checking.  Any attempt to bypass our plagiarism detection systems will be treated as a severe Assessment Offence. |

#### Coursework Submission Requirements

#### **An electronic copy of your work for this coursework must be fully uploaded on the Deadline Date**

#### **For this coursework you must submit a single PDF document.  In general, any text in the document must not be an image (i.e. must not be scanned) and would normally be generated from other documents (e.g. MS Office using "Save As .. PDF"). An exception to this is hand written mathematical notation, but when scanning do ensure the file size is not excessive.**

#### **For this coursework you must also upload a single ZIP file containing supporting evidence.**

#### **There are limits on the file size (see the relevant course Moodle page).**

* Make sure that any files you upload are virus-free and not protected by a password or corrupted otherwise they will be treated as null submissions.
* Your work will not be printed in colour. Please ensure that any pages with colour are acceptable when printed in Black and White.

#### **You must NOT submit a paper copy of this coursework.**

#### **All coursework’s must be submitted as above. Under no circumstances can they be accepted by academic staff**

The University website has details of the current Coursework Regulations, including details of penalties for late submission, procedures for Extenuating Circumstances, and penalties for Assessment Offences.

See <https://www.gre.ac.uk/policies/undergraduate-and-postgraduate-taught>

## **Detailed Specification**

**Please read the entire coursework specification before starting work.**

**This assignment consists of two parts:**

* **Part A (implementation) will be completed individually**
* **Part B (report) must be completed individually**

**Universal Yoga Apps**

Universal Yoga is a yoga studio which offers different types of classes in a studio in London. The studio has approached you to create a mobile app which would allow them to set up class schedules.

You are to create a mobile app for use by employees of Universal Yoga that allows them to record details of their planned class schedule, as well as adding individual instances. Details are stored on a phone but can then be uploaded to a server so they can later be viewed and booked by customers. A separate hybrid app should be developed for customers. The features the apps should support are given below:

* Features **a)** to **f)** are to be implemented as a **native Android app coded in Java**.
* Features **g)** and **h)** are to be implemented as a **hybrid app coded using** Xamarin/MAUI.

Your final app is the culmination of all your hard work on this course, which should become a strong addition to your programming portfolio. You should produce an app that is well-designed, robust and useful. The GUI design should be clean, simple to navigate, and operate smoothly without sluggishness or crashes. The app should not require instructions or a manual to use.

#### **Part A – Implementation (80%)**

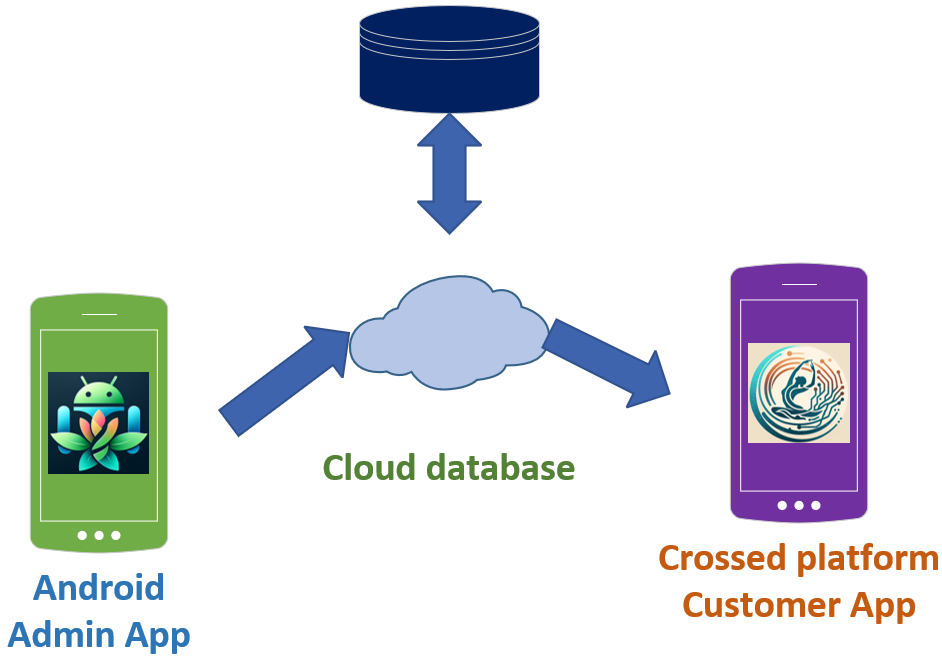


Figure 1 A diagram to show the interactions among different components

**Description of the application**

The overall application contains two mobile apps: one for Admin and one for Customers.

1. **Admin App (Android Native)**
2. **Enter details of yoga class and overall design of the App (10%):**

Note that users must be able to enter all of the following fields. ***Required*** fields mean that the user must enter something in this field; otherwise they will be prompted with an error message. ***Optional*** fields mean that the user can enter something if they wish, but they will not get an error message if nothing is entered.

The user should be able to enter the following details of a yoga course:

* Day of the week (e.g. Monday, Tuesday) - Required field
* Time of course (e.g. 10:00, 11:00) - Required field
* Capacity (how many persons can attend) - Required field
* Duration (e.g. 60 minutes) - Required field
* Price per class (e.g. £10) - Required field
* Type of class (Flow Yoga, Aerial Yoga, Family Yoga) - Required field
* Description – Optional field
* Any additional fields of your own invention – be creative!

The app will check the input and if the user doesn’t enter anything in one of the required fields, the app should display an error message.

Once the details have been accepted by the app (e.g. no required fields were missing), it should display the details back to the user for confirmation and allow them to go back and change any details that they wish.

1. **Store, view and delete yoga class details or reset the database (10%)**

All the details entered by the user should initially be stored on the device in an SQLite database.

The user should be able to **list all** the **details of all courses** that have been entered into the app, edit or delete individual courses.

1. **Manage class instances (15%)**

The course represents a schedule of classes taking place. A user should then be possible to add/edit/delete instances of classes to a schedule, including the following details:

* Date (e.g. 17/10/2023 – the date should match the day of the week – in the case of 17th of October, it would have to be a course running on Tuesday) - Required field
* Teacher (who is teaching this class) – Required field
* Additional comments - Optional field

The user must be able to enter multiple class instances for a single course. The app should store all details entered on the device in an SQLite database. It should be possible for a user to select a course and display all instances.

1. **Search (10%)**

The user should be able to search for a class in the database by teacher name. Ideally the user should be able to enter the first few letters of the name and display all matching classes.

Advanced search options will allow searching for all classes with the following criteria: date or day of the week. It should be possible for a user to select an item from the resulting search list and to display its full details.

1. **Upload details to a cloud-based web service (10%)**

The user should be able to upload all the details of yoga courses stored on the mobile device to a cloud- based web service. It should be possible for all courses to be uploaded at once.

The app should check the status if there is a network and be able to make connection to the internet before uploading data.

Note that: your application should handle the situation that if data in the local database is changed, it should be synchronised with the cloud database.

**Extra features to Admin App**

1. **Add additional features to the Android app (5%)**

**Features a) to e) are the core requirements for the app.** If you have implemented these and want to add some additional features, then you may. Any enhancements should be implemented **in addition to** **NOT instead of** the core requirements. The idea is that these features **stretch** your skills, so be prepared to do your own research and feel free to show off! You can think of your own enhancements. Here are some possible examples:

* Allow photos taken by the camera to be added to the data stored
* Pick up the location automatically from the user's location
* Anything you can think of – again be creative!

1. **Customer App (Hybrid App)**
2. **Create a hybrid app of the app using .NET MAUI or other hybrid technology (10%)**

Implement an app using a hybrid technology for use by customers wishing to book yoga classes. This app should connect to the cloud service that you created for the Android Admin app. The customer should be able to view a list of available classes by connecting to the cloud service.

It should also be possible to search for yoga class instances by the day of the week and/or the time of day.

**Extra features to Customer App**

1. **Implement booking functionality in the hybrid app (10%)**

The user should be able to add classes to a shopping cart and then submit the shopping cart to the cloud service, along with their email, so there is a record of who booked the class.

The user can have an ability to view all classes that they booked.

#### **Part B – Report (20%)**

The report is to be completed **individually.**

Write a report consisting of **all** the following sections:

* **Section 1. (2%)** A **concise table** containing a checklist of the features you have been able to implement and your comments for each feature. Please refer to the features list given above in the specification.

**Note that**, giving the false information, it can give a penalty for your coursework.

For example, you might write:

|  |  |  |
| --- | --- | --- |
| **Feature** | **Status** | **Your Comments** |
| **Functionality A** | Fully completed **☑**  Partially completed ☐  Having bugs/Not working ☐  Not implemented ☐ | I used Constraint layout for the activities. Also, I used Material, Styles for all controls. |
| **Functionality B** | Fully completed **☑**  Partially completed ☐  Having bugs/Not working ☐  Not implemented ☐ |  |
| **Functionality C** | Fully completed☐  Partially completed ☐  Having bugs/Not working ☐  Not implemented ☐ | I have created the user interface for data entry, but the data is not being stored |
| **Functionality D** | Fully completed☐  Partially completed ☐  Having bugs/Not working **☑**  Not implemented ☐ | Implemented but the app throws an exception if no matching result is found. |
| **Functionality E** | Fully completed☐  Partially completed **☑**  Having bugs/Not working ☐  Not implemented ☐ |  |
| **Functionality F** | Fully completed☐  Partially completed ☐  Having bugs/Not working ☐  Not implemented **☑** |  |
| **Functionality G** | Fully completed☐  Partially completed ☐  Having bugs/Not working ☐  Not implemented **☑** | No additional features implemented |
| **Functionality H** | Fully completed☐  Partially completed ☐  Having bugs/Not working ☐  Not implemented **☑** | Implemented but the app throws an exception if no matching result is found. |

* **Section 2. (4%)** Write a reflection (approximately 350 words) on how the Apps, both Android and crossed platform, were developed. Discuss lessons learnt, what you think went well and what you think could have been improved and how.
* **Section 3. (8%)** An evaluation of your app(s). Write between 700 to 1000 words evaluating the app(s) that you have produced**.** Be specific and justify any statements you make. Just saying things like "my app is well designed" without justifying the statement will not gain you any marks. Also, explain how your app could be improved. Again, you need to try to be specific e.g. saying something like "It needs to be made more secure by adding security features" will not gain marks. Your evaluation should include, but need not be limited to, the following aspects of your app:
  + 1. Human computer interaction (you will have a lecture about this)
    2. Security (you will need to research this yourself)
    3. Ability of the app to run on a range of screen sizes and how this could be improved (we will touch on this in the course, but you will need to do additional research)
    4. Changes that would need to be made in order for the app(s) to be deployed for live use.

This sort of discussion will form an important part of your BSc project report so use this opportunity as a way of practicing your skills in writing an evaluation.

**English Proficiency + report format (2%)** – marks will be allocated to the level of language used in the individual report, including correct spelling and use of grammar

* **Section 4. (2%)** Screen shots demonstrating each of the features that you have implemented. Give captions or annotations to explain which features are being demonstrated.
* **Section 5. (2%)** Code listing of any code files you have written. You do not need to include generated code. Please clearly label the code, so it indicates the source file and programming language.

Demonstration (0% but see below)

You are required to prepare a brief video showing your implementation including Android app and cross platform app (one video per submission). The duration should be approximately 10 minutes. You then upload your recorded demonstrations onto Panopto. To help us identifying your **recorded videos** easier and quicker, please follow the **naming convention** “COMP1786-Firstname-Lastname-ID”. The links can be mentioned in your CW reports. If unable to identify your recorded video, it may result in a decrease in your overall marks.

You have a responsibility to make sure your tutors be able to access and view the videos.

In the video, you must clearly demonstrate the working version of the apps (Android and Xamarin/MAUI) and link those working functions to the correspondent code. You might attend a brief Q&A session with your tutor where you will be asked questions about your implementation and the code.

**Failure to demonstrate might lead to a failed assessment.**

**Deliverables**

1. A zip file containing all the files required to run your app(s). Please try to structure your work so that it is easy for the person marking your work to compile and run your app(s) if they need to. Any compilation, installation or running instructions should be included in a “readme” file.

If you have **borrowed code** **or ideas** from anywhere other than the lecture notes and tutorial examples (eg. from a book, somewhere on the web or another student) then include a reference showing where the code or ideas came from and comment your code very carefully to show which bits are yours and which bits are borrowed. This will protect you against accusations of plagiarism. Be aware that **the marker will look for similarities between your code and that submitted by other students** so please do not share your code with any other students as this is considered to be plagiarism. Note that **the upload of this zip** file **is MANDATORY**. It must be along with your report (deliverable 3) **or you will lose marks and are likely to fail the coursework**.

1. A Video Demonstration of your app and what you have achieved, showing the app in operation and displaying all functionalities you have implemented. This should be approximately 10 minutes in duration as mentioned above. There should be only one video submission per submission.
2. A brief Q&A session where you will be asked questions about your app(s) and be expected to show an understanding of the code that you have written and the design decisions that you have made. For instance, the instructor may ask you to talk them through the code that implements a particular feature. If you are unable to answer questions about your code, you may be investigated for plagiarism. The date, time and details of the Q&A meeting will be set by the instructors.
3. A report consisting of **all** the sections described in the specification. It must be **uploaded**

## **Formative Assessment**

You are encouraged to show the progress made with the implementation in the labs on a weekly basis to get informal feedback on how the work progresses and, on the design, and implementation decisions taken at every stage of the assignment.

There are also weekly lab exercises, which train you in the knowledge required to complete the assessment. Completed exercises do not need to be uploaded but shown to the instructor for feedback.

## **Grading Criteria**

**Please note that the PASS GRADE FOR THIS MODULE IS 40%.**

This coursework will not be marked anonymously.

There are 8 features. Percentage score for each feature is stated in the coursework specification. This doesn’t mean that you will automatically obtain the full marks for a feature for implementing it, but it will be graded based on the assessment criteria below. Section 3 of the report is also worth 20% and each remaining section of the report is graded as stated in the coursework. The below is applied to overall of the coursework.

**For a very high 1st class (90% to 100%)**

* A native Android application fully implementing all features. No bugs and negligible weaknesses. Exemplary quality code.
* Two or more creative additional features (h) to both apps.
* Demonstration: able to show good knowledge of code implemented. Not only able to answer questions about **how** you did something but also **why** you chose to do it in a particular way and **compare** it with possible alternative implementations.
* Report complete, accurate and easy to read. Section 5 within specified word count, logically structured and making some insightful points about **all four** of the issues specified.

**For a high 1st class (80% to 89%)**

* A native Android application fully implementing all features. No bugs and very minor weaknesses. Outstanding quality code.
* A working hybrid prototype and one or more creative additional features (h) to both apps.
* Demonstration: able to show good knowledge of code implemented. Not only able to answer questions about **how** you did something but also **why** you chose to do it in a particular way.
* Report complete, accurate and easy to read. Section 5 within specified word count, logically structured and making some insightful points about **all four** of the issues specified.

**For 1st class (70% to 79%)** the following is required

* A native Android application fully implementing at least six features. Very few minor bugs or weaknesses. Excellent quality code.
* A working hybrid prototype and one or more very good additional features (f) or/and (g).
* Demonstration: able to show good knowledge of code implemented. Not only able to answer questions about **how** you did something but also **why** you chose to do it in a particular way.
* Report complete, accurate and easy to read. Section 5 within specified word count, logically structured and making some insightful points about at least three of the four issues specified.

**For a mark in the range 60 to 69%** the following are required:

* A native Android application fully implementing at least five features. Few minor bugs or some weaknesses. Very good quality code.
* A working hybrid prototype.
* Demonstration: able to show good knowledge of code implemented.
* Report complete, accurate and easy to read. Section 5 within specified word count logically structured and making sensible points about at least two of the four issues specified.

**For a mark in the range 50 to 59%** the following are required:

* At least five features have been completed across the native Android application and the hybrid prototype, possibly with some bugs and weaknesses. Good quality code.
* Demonstration: able to show good knowledge of code implemented.
* Report complete including some attempt at section 5

**For a mark in the range 40 to 49%** the following are required:

* At least four features have been completed across the native Android application and the hybrid prototype, possibly with some bugs and weaknesses. Good quality code.
* Demonstration: able to show reasonable knowledge of the attempted features.
* Report mostly complete.

**For a fail mark in the range below 40%**:

* A native or hybrid android application attempting three of less feature but buggy.
* Demonstration: unable to show reasonable knowledge or understanding of the attempted features.
* Report missing or mostly incomplete.

For a coursework that does not fit into any of these categories (e.g. features a) to d) are implemented but section 3 of the report is less than 700 words and weak) the grade will be determined by taking the lower mark applicable and making some upward adjustment for the other aspects of the coursework. Please be aware to pass you must submit a working app **and** an acceptable report.

**NOTE: Failure to do your Demonstration will normally result in you being awarded 0% for the coursework. Even if your implementation and report are excellent and would be awarded a mark of 80% but you don’t do a Demonstration then you may score 0% for the coursework.**

## **Assessment Criteria**

Your app(s) will be assessed on the following criteria.

* **Features implemented.**  The number of features (listed as a) to h) in the specification above) that you have successfully implemented will have a big effect on your overall mark.
* **The quality of the application code you produce**. Credit will be given for inclusion of meaningful comments in the code, use of the sensible naming standards (eg. for packages, classes, variables, and methods), code layout (e.g. indentation to make the structure of "if" statements and loops clear), avoidance of unnecessary duplicate code, for the Android app use of appropriate Java language features and Android APIs. Java coding conventions (covering naming and indentation etc) can be found at <http://www.oracle.com/technetwork/java/codeconvtoc-136057.html> – (please note that this Oracle page is no longer maintained as the java conventions are common standard, but the page continues to be used as a sources of reference).
* **The user interface.** This is not a course about user interface design, but credit will be given for making your application as pleasant an experience as possible for the user. Examples of good practice are: allowing the user to choose options rather than their having to type in input, sensible default values, validation of input, and meaningful messages. Credit will be given for showing the use of a range of appropriate features from the Android GUI API.

Your report will be assessed on the following criteria.

* Are all the required sections included and completed properly?
* Does the report give an accurate reflection of what you have achieved?
* Is the report clear and easy read? Does it follow the structure specified?
* Is the evaluation (section 3) realistic and does it show that you have really thought about your app(s) and the specified issues as well as how they may be enhanced to be ready for live deployment. Do you show insight into the complexities of app development and the challenges of balancing the various constraints involved?

## **Summative Feedback**

Feedback on the final submission will be provided in written format within 21 days of submission.