6COSSC004W Mobile Native Development Coursework 2 – SwiftUI Weather & POIs Application (2022/23)	
Module leader	PHILIP TRWOGA
Unit	Coursework 2: Real-time data app in SwiftUI
Weighting:	**6**0%
Qualifying mark	30%
Description	A tourist weather utility app
Learning Outcomes Covered in this Assignment:	This assignment contributes towards the following Learning Outcomes (LOs): LO1 understand language features and programming practice required for native development LO2 apply industry standard tools for design and development LO3 communicate and defend work by both written and oral means
Handed Out:	15th Nov 2024
Due Date	10th Jan 2024 13:00 Demo & Viva 17th Jan 2024 - Room & Time TBC
Expected deliverables	Submit on Blackboard a zip file containing: **Complete XCode 14.X solution - must be in Swift 5.7 or above**
Method of Submission:	Electronic submission on BB via a provided link close to the submission time. The file you upload should have the following naming format: E.g. **6COSC021W**_StudentNumber_firstName_lastName.zip
Type of Feedback and Due Date:	Formative feedback will be provided during tutorial sessions. Verbal feedback on the submitted CW will be provided during an online viva. Students are encouraged to record this feedback at this time. Feedback and marks are due by the TBD. Feedback shall also be given on the Blackboard. Note: All marks will remain provisional until formally agreed by an Assessment Board.

Assessment regulations

See the Assessment guidelines https://www.westminster.ac.uk/current-students/guides-and-policies/assessment-guidelines

for a clarification of how you are assessed, penalties and late submissions, **what constitutes plagiarism etc.**

Penalty for Late Submission

If you submit your coursework late but within 24 hours or one working day of the specified deadline, 10 marks will be deducted from the final mark, as a penalty for late submission, except for work which obtains a mark in the range 40 – 49%, in which case the mark will be capped at the pass mark (40%). If you submit your coursework more than 24 hours or more than one working day after the specified deadline you will be given a mark of zero for the work in question unless a claim of Mitigating Circumstances has been submitted and accepted as valid.

It is recognised that on occasion, illness or a personal crisis can mean that you fail to submit a piece of work on time. In such cases you must inform the Campus Office in writing on a mitigating circumstances form, giving the reason for your late or non-submission. You must provide relevant documentary evidence with the form. This information will be reported to the relevant Assessment Board that will decide whether the mark of zero shall stand. For more detailed information regarding University Assessment Regulations, please refer to the following website http://www.westminster.ac.uk/study/current-students/resources/academic-regulations

Note: By submitting the work through Blackboard you are acknowledging that this is solely your own work. Any code which is not created by you MUST be clearly commented as such. Any code discovered to not have been created by you will mean that the work will be submitted to academic standards for a potential assessment offence, which may result in a zero mark in the component or whole module.

6COSC021W Native Programming Coursework 2
Individual coursework

Introduction

This coursework is about using **SwiftUI** Framework to build a tourist utility app that will provide weather information and show tourist places on a map for that location. The app weather data must be fetched from OpenWeatherMap using the api listed below:

Weather-data:
https://api.openweathermap.org/data/3.0/onecall?lat={lat}&lon={lon}&exclude={part}&appid={API key}

Tourist-Places of interest will be loaded from a json file (places.json) that has been included in the starter template. The assets folder has file with images of tourist places in the folder 'Locations'.

You will need to register with openweather.org to obtain the respective API key.

Note that the openweather api is a subscription-based API and payment details will have to be added, however, you are allowed 1000 free calls per day, and you can set a limit to number of calls (900) so as not incur any charges

Very Important Note

The application must be built using SwiftUI, CoreLocation, MapKit frameworks and the OpenWeather API.

Other frameworks or libraries such as Pods, Lottie, SDWebImage, WeatherKit, UIKit (including any not listed here are) ARE NOT ALLOWED, and any use of these frameworks will result in your mark being capped to a maximum of 30.

You must use CWK2Starter template that has been provided and add where necessary additional models, view-models and views to build the app to provide complete functionality as required.

Coursework grading:

Your coursework will be graded as a combination of demonstration of app built and submitted on Blackboard and a compulsory viva.
The app build and demo is worth **70 marks**, viva **20 marks and independent enhancements 10 marks**. Enhancements must be clearly justified and add value to the app with regards to UI, UX and functionality implemented.

The demonstration will require you to download your submission from Blackboard and run that in the presence of your tutor.
Do make sure that your code compiles and runs in the Mac labs as non-compiling code will be **graded as 0**.
The viva will follow your demonstration. You will be asked questions about your submission and/or general questions to demonstrate your understanding of the work to demonstrate you have adequately met the learning outcomes of the module. You will also be expected to defend your code.
Viva marks will be the academic judgment of the marking tutor in respect of your overall output i.e., your coded solution and your understanding of the work.
Written evidence of any enhancements including any code, must be submitted as a pdf document with your coursework submission.

Coursework brief:

The application design is as depicted in the images shown below and **no changes** are allowed regarding background images nor navigation between different views. The information presented in each view must be the same as shown in the images. Weather icons must be fetched from openweather.
Background images and other images are in the starter template project.

There are many ways to manage an API call and you are only allowed to use the ones that will be covered and used in the seminar sessions and included in the starter template project.

The functional requirements are:

- When the app launches for the first time, a user will see the current weather for London including the date and time as shown in Figure1.

- Forecast tab gives a rich weather view of the current location, with a top part a scrollable view showing a summary of hourly weather for the next 48 hour and the bottom part a vertically scrollable view showing summarised weather for each of the following the next 8 days for the location from City tab (Figure 2).

- Places tab is a compound view with top part showing a map with pins of tourist attraction places and the bottom part a scrollable list of the tourist places loaded from file data, if there is data relating to the location (Figure 3).

- When the user changes the location in City tab, all other views must also change and load relevant data for this new place (Figures 4, 5, 6 and 7).

The non-functional requirements are:

- The app interface follows Apple guidelines.

- The app interaction is consistent.
- The app data should be correctly formatted.

Design Images

The images are based on iPhone 14 Pro Max, iOS16.0 Simulator.

Initial App Launch: Default location: London

Figure 1

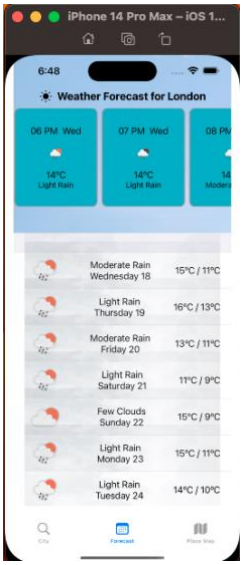

Figure 2

Figure 3

Figure 4

New Location: Rome

Figure 5

Figure 6

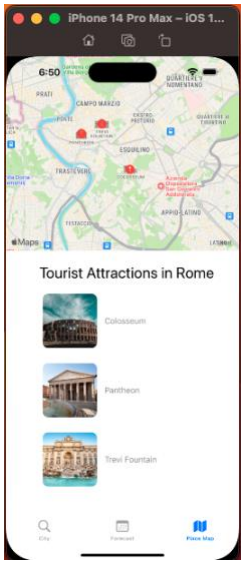

Figure 7

Marking Scheme

1. Weather summary view for any location both in terms of view design and correct current data as per images shown.

   15 marks
2. Detailed hourly and 8 day forecast correctly shown both in terms of view design and data accuracy as per images shown.

   20 marks
3. Composite Map view of the current location set to Rome with annotations of places of interest (POI) listed below the map, both in terms of design and correct information as per images shown.

   20 marks
4. Composite Map view of the current location set to Paris with annotations of places of interest (POI) listed below the map, both in terms of design and correct information as per images shown.

   15marks

Indicative Viva - code defence and explanation (20 marks)

1. Use of environment variables and data flow management.
2. Weather for any location - api call and management, conversion of json data to swift.
3. Map rendering and updating with new location.
4. Main thread and background threads and when to switch.

Enhancements must be clearly justified and add value to the app with regards to UI,UX and functionality implemented (10 marks).