

**6COSSC004W Mobile Native Development**  
**Coursework 2 – SwiftUI Weather Application (2021/22)- referred deferred version**

Module leader	PHILIP TRWOGA
Unit	Coursework 2 – APIs in SwiftUI
Weighting:	<b>60%</b>
Qualifying mark	30%
Description	<b>A SwiftUI weather app using a openweather API</b>
Learning Outcomes Covered in this Assignment:	<p>This assignment contributes towards the following Learning Outcomes (LOs):</p> <p>LO1 understand language features and programming practice required for native development</p> <p>LO2 apply industry standard tools for design and development</p> <p>LO3 communicate and defend work by both written and oral means</p>
Handed Out:	12 <sup>th</sup> June 2022
Due Date	07 <sup>th</sup> July 2022: 1:00 pm
Expected deliverables	<p>Submit on Blackboard a zip file containing:</p> <p><b>Complete XCode 13.X solution - <u>must</u> be in Swift 5.x</b></p>
Method of Submission:	<p>Electronic submission on BB via a provided link close to the submission time. The file you upload should have the following naming format:</p> <p>E.g.  <b>6COSC004W_StudentNumber_firstName_lastName.zip</b></p>
Type of Feedback and Due Date:	<p>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.</p> <p>Note: All marks will remain provisional until formally agreed by an Assessment Board.</p>

## 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.

# 6COSC004W Native Programming Coursework 2

## Individual coursework

### Introduction

This coursework is about using **SwiftUI** Framework to build a weather application. The weather data is to be fetched from <https://openweathermap.org/api> using APIs provided by openweather and listed below.

The initial application view will show weather for London that displays the current temperature and humidity (Figure1) with option to enter a new location. There will be options to view more detailed weather (Figure2) and a 5 Day forecast (Figure 3)

### Very Important Note

The weather application design is 3 views as shown in the Figure1, Figure2 and Figure3. This is the application design and must be implemented as such, any different implementation will be graded as '**not meeting the specification**'

The APIs that must be used are (use your own keys):

[https://api.openweathermap.org/data2.5/weather?q=\\(city\)&appid=\\("YOUR KEY"\)&units=metric"](https://api.openweathermap.org/data2.5/weather?q=\(city)&appid=\()

[https://api.openweathermap.org/data2.5/forecast?q=\\(city\)&appid=\\("YOUR KEY"\)&units=metric"](https://api.openweathermap.org/data2.5/forecast?q=\(city)&appid=\()

The API implementation must be as demonstrated in week 11 lecture and/or as per Phil's recording – [CW2 tips](#). Any different implementation of the APIs will be graded 0.

Frameworks such as Combine, Pods, Lottie, SDWebImage are not needed and **NOT ALLOWED**.

The coursework implementation must be solely based on the module content coverage.

The functional requirements are:

- User can input a place name and get the current weather data displayed in real time – temperature and humidity
- User can get detailed current weather that will include weather summary icon, temperature, humidity pressure and windspeed.
- User can get a 5-day forecast with 3 hourly predictions.
- The app must safely handle non-existent locations and non-alphabetic entries.
- Location names with spaces must be correctly processed, e.g., San Francisco, etc
- Transition from any one view to any other view is managed correctly by use of buttons or navigation links

The non-functional requirements are:

- The app interface follows Apple guidelines.
- The app interaction is consistent.
- The app should render on any iPhone

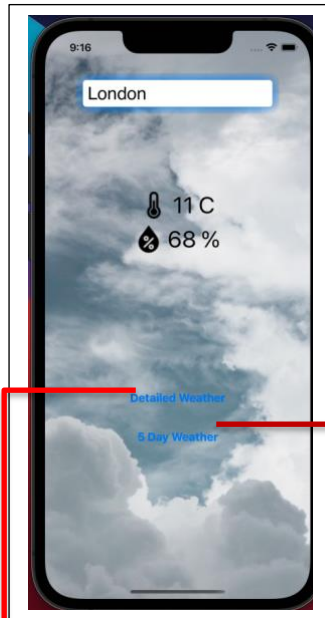


Figure 1



Figure 2

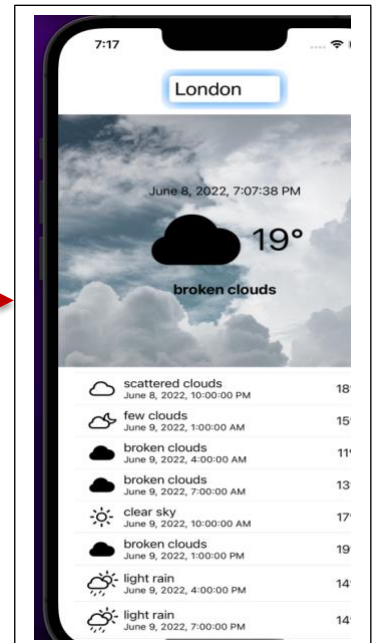


Figure 3

- Initial view rendered with some data either from a file or API call (15 marks)
- Detailed view rendered with some data either from file or API call (20 marks)
- 5-Day forecast view rendered with some data from file or API call (10 marks)
- User location entry – handle locations with space in names (5 marks)
- User location entry for non-existent places -suitable error message (5 marks)
- Use of icons to represent weather element (10 marks)
- Interface design demonstrating good use of stacks and modifiers (10 marks)
- Transition between views (10 marks)
- API(s) construction and usage (5 marks)
- Error management in network connections and safely unwrapping data (5 marks)
- Coding standards and naming conventions (5 marks)

**End of document**