

# Mobile Platform Development Assignment

Session 2024/2025

MHI326841/MHI326842/M3I326836

## Assignment – Coursework 1 (CW1)

This is the first assessment component for Mobile Platform Development. The second component (a class test) will run by the end of the trimester.

The purpose of this assignment is to test your design and programming ability in relation to the development of an application for a mobile computing device. The target device is a mobile computing device such as a mobile Smartphone running Android 7.0 (Nougat) or later (*minimum SDK: API level 24*). This means that the application will run on approximately 96.3% of current devices.

The marking scheme shown below indicates how the application will be assessed. At this stage, you are expected to show an in-depth level of understanding of the software lifecycle and the activities that you need to carry out as part of that lifecycle in producing your application.

Some form of iterative development is therefore most likely to be appropriate. However, your final submission and reports should only cover your final iteration.

### Scenario

BBC weather produces RSS feeds for many locations in the UK and around the world. A description of how this works can be seen on the following page:

<http://www.bbc.co.uk/weather/about/17543675>

Each location is represented by a numerical ID code. You can get a list of all available IDs from [this link](#); for convenience, the locations of interest for this assignment are given in *table 1*:

Location	Location ID
Glasgow	2648579
London	2643743
NewYork	5128581
Oman	287286
Mauritius	934154
Bangladesh	1185241

**Table 1: BBC weather RSS feeds location IDs**

The RSS feed links terminate with the code for each location. There are 2 separate feeds:

- **a 3-day forecast.** For example, for Manchester location (ID 2643123), the link is:  
<https://weather-broker-cdn.api.bbc.co.uk/en/forecast/rss/3day/2643123>
- **the latest observations.** For the same example location, the link is:  
<https://weather-broker-cdn.api.bbc.co.uk/en/observation/rss/2643123>

You are required to develop an Android application which parses the RSS feeds and reports the information in the forecast in a meaningful way suitable for a mobile device. There are a number of decisions that you will need to make regarding how this information is displayed.

This is a programming assignment and you must use Android Studio and write your application in Java/XML.

- Linking back to the BBC Weather website as part of the developed app is **NOT PERMITTED** and will score 0 marks for any functionality that this provides.
- Producing functionality based on HTML in a web view in Android is **NOT PERMITTED** either.
- You can make use of Google Maps APIs (or other free mapping services) in your application BUT you **ARE NOT** permitted to use any other external frameworks.

## Application Specification

The Android Application that you are developing should provide the user with appropriate functionality. The Android Application that you are developing should allow the user to select some option(s) related to the available weather information and display the information to the user in a useful and meaningful way, in line with the functionality outlined below. You will need to *parse* the raw data from the feed to extract the relevant information fields, and store the information in such a way that the app can exploit it. The parsing **MUST** be done in the mobile app on the downloaded data `String`, using the `PullParser` class.

The Android Application that you are developing should provide functionality such as:

- Displaying a *3-day weather forecast* for the GCU campus locations and associated institutions around the world. This should display only basic information: min/max temperature and the classification of the weather (Sun, Light cloud, Rain, Snow, etc.).
- Displaying the day of the week, date and place the forecast in view is for.
- Some means of displaying the more detailed weather forecast information for a particular day (as available in the “3-day forecast” RSS feed).
- Some means of displaying the *current* weather for the selected location (as available in the “latest observations” RSS feed).

- Good use of graphics and text to convey information about the weather for a specific campus location. You can use maps and weather icons to decorate your visuals
- The ability to navigate from location to location in a *sequence* as determined by the order of locations in a suitable data structure such as a list. The display should wrap around to the start of the locations when the end of the sequence is reached.
- The application will update the data when it starts, and also at regular intervals as set by the user (with the default being 08:00 and 20:00).

You might also wish to implement some other additional functionality you think useful.

You may want to identify specific Use Cases or XP style user stories to help you to work out what the indicated functionality needs to do (although this is not part of the assignment).

A good solution will make good use of graphics, maps and text to convey information about the weather. There is scope for having multiple views of them. You may find it useful to review a number of web and mobile applications that display this type of information.

You should design your interface to display appropriately in **portrait** and **landscape** mode. This means that you are expected to design screen layouts making appropriate use of the screen space to display the required information when the screen is in a particular orientation. This will require you to create different xml arrangements for landscape and portrait.

## Getting Started

You will be provided with some code that you can use in the creation of an Android Studio project as a starting point. The code that you will use in the Android Studio Project will allow you to access an example RSS resource and display the raw data in a `TextView`. This project is designed as a starting point and “test bed” for your assignment.

To access different locations, you will need to generate the corresponding URLs (for instance using standard `String` methods). Once the data is read, you will have to parse it to extract the info and then display it using appropriate Android graphical components. There are a number of possible ways to parse the XML data, but for this assignment **you must use an approach based on the `PullParser` class available in Android** (that has been covered in lectures and labs). *Other solutions will result in 0 marks for that part of the assignment.*

## Creating an Android Studio Project and Using the Supplied Starter Code

Please create your Android Studio Project using the naming convention below:

`Surname_GivenName_StudentID`

e.g. `smith_jimmy_S123456`

then you can then copy and paste the supplied code in your newly created project.

Also, make sure to **add the entries in bold** below into your **manifest** file:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="org.me.gcu.smith_john_s123456">
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:usesCleartextTraffic="true"
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Smith_John_s123456">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

See the links below for more information on what these entries do.

<https://developer.android.com/guide/topics/manifest/application-element>

<https://developer.android.com/training/basics/network-ops/connecting>

You must also **include your student ID** as follows:

- hard-code your student ID number into every **screen** that you produce in the mobile application
- every **Java class** that you produce should have a header as a Java comment containing your name and student ID number
- every **xml file** should have a comment line containing your name and student ID number.

These are **STRICT** anti-plagiarism requirements.

## Marking

Marks are allocated as follows.

Category	Mark
Design and coding of the <b>features</b> outlined in the specification ( <i>please check rubric available on GCULearn</i> )	42%
Provision of a separate distinct <b>landscape</b> layout. This must be different from the arrangement used for the <i>portrait</i> view (and not the default one).	8%

Code for parsing and processing of the data from the XML feed and storing the information in appropriate Java classes.	10%
Software Architecture that you utilised in order to build the Android application. There are a number of ways that you can create a data-driven application in Android like this one. These approaches range from the simple through clumsy/inefficient, to sophisticated approaches. The marks here are intended to encourage you to explore the different approaches that you can use.	10%
Background threads code (different from the provided starter code): 6% Automatic updating of the data: 4% <ul style="list-style-type: none"><li>You are required to improve on the threading approach that is used in the starter code provided. If you simply reuse the provided threading code, you will receive 0 marks for this section.</li></ul>	10%
Documented Testing. This should be presented as a table showing what tests you carried out, and <b>clearly demonstrating</b> what the expected outcome was along with the actual outcome.  The testing should demonstrate:  <ul style="list-style-type: none"><li>that the data in the link is consistent with the data that is displayed in the various views provided by the app.</li><li>that the user can navigate around the app and select the required functionality as identified in the specification.</li><li>that the functionality required on the features asked for in the specification have been tested.</li><li>A suitable behaviour when not connected to the internet or internet connection is lost.</li></ul> Minimal implementations will not score full marks here.	15%
Video demo, produced using <i>ScreenPal</i>	5%
<b>Total</b>	<b>100%</b>

Note that minimal (or very simple) implementations will not attract full marks in any particular section above.

## Recommended pace of Work

Begin with the starter code to demonstrate that you can access the data stream(s) and focus on having the data parsed with `PullParser` and stored into appropriate data structures in your own project.

You can then start to add features in to address the coursework specification provided. It is recommended that you keep a separate copy of each completed version before moving on to adding more functionality. This way, if something goes wrong and you run out of time you can always submit latest working version (instead of none). Even a very basic version can be enough to get over the pass mark; you can use the rubric to self-assess if your current version would be enough, and to direct your efforts efficiently.

## Submission

The full assignment should be submitted **NO** later than **12:00 (midnight) on Friday the 12<sup>th</sup> of April 2024**. You will be required to produce a video of your application running using [ScreenPal](#).

All work must be carried out using Android Studio, and handed in via GitHub ensuring that it is ready to be imported into Android Studio. The application must be written in Java with XML used to describe the interface as appropriate.

You are required to make 3 separate submissions. Separate areas on GCULearn will be set up to allow this.

### Submission 1: document

A Microsoft Word document (.doc or .docx) containing:

- a link to your video (**in mp4 format maximum 5mins**) stored on your OneDrive; please make sure to allow access to anyone with the link (only the module team will get this link).
- a link to your Android project stored on GitHub. Please make sure that in can be directly imported into Android Studio by testing it yourself.
- a link to the project .apk file stored on GitHub
- your written **documented test strategy**

This file **MUST** be named with your student identification following the format:

*Surname\_GivenName\_StudentId.docx*

### Submission 2: video

The Video of your application running in the emulator using [ScreenPal](#) (formerly *ScreenCastOMatic*). Use video formats .wmv or .mp4

This file **MUST** be named with your student info followed by \_Video, i.e.

*Surname\_GivenName\_StudentId\_Video.wmv* or

*Surname\_GivenName\_StudentId\_Video.mp4*

You can also submit a link to the video stored in your OneDrive -please ensure to allow access to anyone with the link.

### Submission 3: source code and video (packed for the External Examiner)

You should create a folder using the following folder naming convention:

***Surname\_GivenName\_StudentId***

Then you should copy all of those files into this folder:

- .apk file
- ScreenPal video
- Android Studio project (as its own zip file). Please use “*Clean project*” before producing the zip file to reduce its size.

Then compress the whole folder to a zip file (NOT a .rar, .7z or any other compressed file format, and NOT password protected). Finally, upload the resulting .zip file.

This zip file is a backup, and to facilitate sending coursework to the External Examiner.

## **Marking**

The module team marking the assignment will follow the provided rubric, looking for:

- conformance to the specification
- appropriate use of Android graphical components
- appropriate screen displays in Portrait and Landscape layout which utilise the space appropriately
- appropriate use of layout managers
- appropriate use of Java classes to hold for example information about the roadworks after it has been “parsed” from the XML data.
- appropriate use of menus/toolbars/action bars
- appropriate use of Threads
- an appropriate architecture for the overall application
- appropriate application of testing

A specific number of marks are allocated for the use of **threads** to handle the processing of the data from the RSS Feed. This code **MUST** be different from the supplied started code.

The number of students tackling this module is large. Please make every effort to ensure that you follow the instructions above and submit **WHAT** you have been told **WHERE** you have been told to submit it and in the **SPECIFICIED** format.

Please take additional care if you are a **Mac** user to ensure that you submit in windows-friendly formats as specified.

Marks will be deducted for late submission in line with University Policy.