**MPD marks Breakdown of Features**

**Features**

|  |  |
| --- | --- |
| **Feature** | *Marks* |
| A 3-day weather forecast (basic info only) for the GCU Campus locations and associated institutions around the world in a suitable format that exploits the screen area.  This must provide a way to *select* one particular location to show either *current* weather or *detailed* forecast. | *8 marks*    Up to 5 marks for display format  Up to 3 marks for selection approach |
| Some means of displaying the *current* *weather* for the selected location (as available in the “latest observations” RSS feed). | *4 marks*  2 marks for selection – date and location  2 marks for display of information |
| Some means of displaying the more detailed weather forecast information for a particular day (as available in the “3-day forecast” RSS feed).  The user should be able to click on an entry and see the extended forecast information. | *6 marks* |
| Displaying the day of the week, date and place the forecast in view is for. This should be shown on each screen. | *4 marks* |
| The ability to navigate from campus location to location in a sequence. The display should wrap around to the start of the locations when the end of the sequence is reached. | *5 marks* |
| Good use of graphics to convey information about the weather for a specific campus location, including weather icons.  This might include a map view as part of the more detailed info display. | *10 marks*  5 marks for a suitable use of weather icons  5 marks for a suitable use of a map view for the locations |
| Other functionality you might wish to implement | Up to 5 marks depending on complexity. |
| **Total** | ***42 marks*** |

**Processing XML data**

|  |  |
| --- | --- |
| Use of the PullParser approach -and further parsing. | *6 marks*  For the full 6 marks here, the XML data needs to be fully parser and broken down into component parts.  This includes the further breaking down of the text information inside tags (like <description>) as necessary. |
| Storing of the data into data structures. | *4 marks*  2 marks for use of ArrayList (or similar) for basic exchange rates data.  2 marks for the handling of additional data (location IDs, weather icons, map coords, etc.) |

|  |  |
| --- | --- |
| **Portrait and Landscape** |  |
| 2 *distinct* layouts with different arrangements of components. | *8 marks*  For full marks here, 2 separate "*activity\_main*" xml layouts must be produced, with the graphical components’ containers in *different* positions. Changing from one layout to another should keep the current displayed data in a way transparent to the user.  *activity\_mains* with the same views in the same relative positions in portrait and landscape will score a maximum of 3 marks. |

**Architecture (*10 marks*)**

|  |  |
| --- | --- |
| Approach based on the use the Activity class | *Up to 5 marks*  Approach which utilises Intents/Activities (this is specifically the case for approaches which have one class/Activity per data stream): 3 marks |
| Approach which uses Fragments | *Up to 8 marks* |
| Approach which *explicitly* uses MVVM (Model, View, ViewModel) | *Up to 10 marks* |

**Threading and Auto Update**

|  |  |
| --- | --- |
| Auto Update of fetching of Data.  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). | *4 marks*  If data is only updated at the press of a button (basic code), this will yield 0 marks.  Auto-update at start-up only: 2 marks |
| Updated Threading  This should ideally involve replacing the *runOnUiThread* with an alternative approach.  There is scope for the code to be tidied up as well. Better use of methods and/or class approach. | *Up to 6 marks*  Replacing *runOnUiThread* plus a better organisation of code would get full 6 marks.  Use starter code yields 0 marks here.  Using *AsynTask* score 0 |

**Documented testing**

|  |  |
| --- | --- |
| This should be presented as a **table** showing what tests you carried out, and what the expected outcome was along with the actual outcome.  The testing should **clearly demonstrate** (with screen captures, data dumps, etc. as suitable):   * that the data in the RSS source is consistent with the data that is displayed in the various views provided by the app. *Up to 5 marks.* * that the user can navigate around the app and select the required functionality as identified in the specification. *Up to 4 marks.* * that the functionality required on the features asked for in the specification have been tested. *Up to 4 marks.* * A suitable behaviour when not connected to the internet (or internet connection is lost). *Up to 2 marks.*   Applications not implementing all required functionality will not score full marks here. | *15 marks* |

**Video-demo**

|  |  |
| --- | --- |
| Video-demo creation, demonstrating the operation of the app | *5 marks*  For full marks the video must show the full app functionality running in portrait and landscape, with a suitable voice over describing the operation.  Note very minimal solutions with very few features implemented cannot score full here. |