# Softwarica College of IT & E-Commerce

STW300CEM Android Applications Development

**Assignment Brief 2020/21**

| | | | |
|---|---|---|---|
| **Module Title:**<br>Android Applications Development | **Ind/Group:**<br>Individual | **Cohort:**<br>March 2022 - Regular | **Module Code:**<br>STW300CEM |
| **Coursework Title:**<br>Developing an android app | | | **Handout Date:**<br>TBD |
| **Lecturer:**<br>Kiran Rana | | | **Due Date:**<br>TBD |
| **Estimated Time (hrs.):**<br>**Word Limit:** | **Coursework Type:**<br>Assignment | | **% of Module Mark:**<br>100% |
| Submission arrangement online via Softwarica Moodle:<br>The link to your GitHub classroom repository containing app source code must be submitted on Softwarica Moodle. The YouTube video file link must also be submitted on Moodle. | | | |

**Failing to submit either the GitHub classroom repository link provided by Softwarica or the video file will result in a zero mark for this assessment.**

File types and method of recording: MP4 for the video
Mark and Feedback date: 2 weeks after submission
Mark and Feedback method: written feedback using Softwarica Moodle

**Module Learning Outcomes Assessed:**

1. Analyze the Android platform components and their uses.
2. Design software suitable for Android platform.
3. Develop apps that can interact with web service APIs.
4. Develop apps that can communicate with sensors built into the phone hardware.

**Task and Mark distribution:**

You are briefed on designing, building and testing an Android app that demonstrates your proficiency in the skills that have been taught during the module. Ultimately the choice of app is entirely up to you, however it is your responsibility to ensure that the app you decide to build will give you sufficient scope to demonstrate the required skills.

**Requirements**

For this task you must design and implement your own Android app. You are free to choose your own ideas for this app. You must:

1. You must be creative in designing your app. Your app needs to offer novel solutions or distinct features to other existing/similar websites.

2. You must use tools and technologies taught during this module such as Google Android SDK and Google Maps Android API.

3. You must NOT use commercial software and/or 3rd party packages/libraries. For example, using Butter Knife libraries or Adobe PhoneGap to create your app is not allowed.

Example app ideas include:

- Online quiz app.
- Online movie ticket app. Allow the user to browse the upcoming and currently showing movies and allow the user to buy or book the ticket online.
- Online Notes app, where user can create, update and delete their notes.
- Restaurant food ordering service apps.
- Chat application.
- Online Shopping app.

**Video**

You are required to record a screen cast in which you are expected to both demonstrate the features of your app and also explain key code behind these features. You will only be awarded marks for features demonstrated in the video. There are a number of restrictions:

1. The video must last no longer than 10 minutes
2. Show your app running on the screen and how to use it.
3. Explain features of your app and how those meet your design requirements.
4. You must include a clear voice for explanation purposes.
5. The video must be produced of high quality so that technical details such as source code can be clearly viewed. The video quality must be at least 720HD.
6. You must upload your video to YouTube as Unlisted category and upload the video link on Moodle.

**Repository**

You must also provide a link to your GitHub classroom repository that contains all source codes for your app. You must make sure that permissions are given to your tutors to view your repository for marking purposes, as well as to check for any plagiarism or collusion. Note this must be your GitHub classroom repository and NOT the public GitHub repository (**github.com**).

**Note**

Apps should be developed using the native development tools and languages taught in the labs. You must not use any other languages or tools. If you make use of any code from other sources (tutorials, help sites, etc.) this must be clearly indicated in your source code.

You should not share any of your assignment code with your fellow students. We use a sophisticated tool that can identify similarities between submissions. If these are detected both parties will be referred to the Academic Conduct Office accused of collusion.

You must make sure that the repository you submit are shared with academics. In other words, the repositories you submit are not protected.

**Notes:**

1. You are expected to use the CU Harvard referencing format. For support and advice on how this student can contact Centre for Academic Writing (CAW).

2. Please notify your registry course support team and module leader for disability support.

3. Any student requiring an extension or deferral should follow the university process as outlined here.

4. The University cannot take responsibility for any coursework lost or corrupted on disks, laptops or personal computers. Students should therefore regularly back up any work and are advised to save it on the University system.

5. If there are technical or performance issues that prevent students submitting coursework through the online coursework submission system on the day of a coursework deadline, an appropriate extension to the coursework submission deadline will be agreed. This extension will normally be 24 hours or the next working day if the deadline falls on a Friday or over the weekend period. This will be communicated via email and as a Softwarica Moodle announcement.

|  | 0 | 4 | 8 | 12 | 16 | 20 |
|---|---|---|---|---|---|---|
| **Version Control and Testing 20%** | No access given to the remote repository. No tests have been written or run. | Some code access provided. Limited attempt at flawed tests and added test cases. | Access provided to GitHub repository. An unsuccessful attempt has been made to write simple tests. | Access has been given to the code on GitHub with multiple commits. Evidence of a limited number of tests written and run. | Evidence of regular commits over an extended period of time together with a range of tests showing how they contribute to code quality. | Commits over an extended period of time, demonstrating the use of branching and merging. A full suite of automated tests ensuring full code coverage. |
| **Layout and Design 20%** | No evidence of the use of standard views. | The application makes use of some basic views with a fixed screen size/orientation. | The app makes use of a limited range of view but only works in a single screen size/orientation. | The app makes use of a wide range of appropriate views and layouts. It works effectively in different screen sizes/orientations. | The app adapts itself to work effectively on both phones and tablets or any screen sizes and orientation. | The app makes use of custom views and animations to produce a professional standard app that works on any device. |
| **Data Persistence 20%** | The app does not work. | Data is displayed with errors. | The correct data is displayed in the app but nothing is persisted if the app is fully closed and opened. | Data is persisted locally when the app is closed fully. | Data includes complex relationships and is persisted locally. | Data is both persisted locally and shared between users via a custom API. |
| **Programming Language 20%** | No code has been made available through online repository. | The app demonstrates only limited understanding of the programming language. | The application is fully functional, and the explanation of the code is clear. | Application uses complex class relations, coding follow Google recommended style and is fully documented | The application makes full use of the range of structures and APIs covered in the module. | The app makes appropriate use of advanced, cutting- edge APIs and techniques not mentioned during the course. |
| **Hardware and Sensors 20%** | The app makes no use of any phone sensors or wearables. | A (failed) attempt has been made to incorporate a phone sensor or wearable into the app. | One sensor or wearable has been incorporated into the app. | The app makes appropriate use of one or more phones sensors. | The app makes appropriate use of multiple sensors and/or wearables. | The app integrates a wide range of wearables and makes appropriate use of multiple phone sensors. |