UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

COS 216 Practical Assignment 5

- Date Issued: **17 May 2021**
- Date Due: **21 June 2021** before **08:00**
- Submission Procedure: **ClickUP**
- Submission Format: **zip or tar + gzip/bzip2 archive**
- This assignment consists of **7 tasks** for a total of **80 marks**.

1 Introduction

During this practical assignment, you will be taking your Practical Assignment 4 with login functionality to make an Android mobile application. You will rely on your API and login functionality to work properly. **You do not need to cater for registration**, assume that the user is an already registered user. **Note:** Since this practical relies on PA4, all previously implemented functionality still needs to work.

After successful completion of this assignment, you should be able to create an Android mobile application version of the video game website you have developed thus far.

The specific Android pages for this assignment will showcase the following functionality:

- A copy of the Practical Assignment 4 'Trending' page as the Home page of your app
- The ability to refresh the page to get the latest data
- Login and Logout functionality

2 Constraints

1. You must complete this assignment individually.
2. You may ask the Teaching Assistants for help but they will not be able to give you the solutions.
3. You must produce all of the source files yourself; you may not use any tool to generate source files or fragments thereof automatically.
4. You may not use web containers.
5. You can use any SDK of your choice with the minimum Android version 4.4 KitKat.

3 Submission Instructions

You are required to upload all your source files (e.g. Java, APK and images) to ClickUP. Make sure that you test that your application works.

NB: You must also submit a ReadMe.txt file. It should include default login details (username and password) for a user you have on your API, as well as any bonus features you have implemented.

4 Online resources

Android Studio - `https://developer.android.com/studio/index.html`

Andriod UI - `https://developer.android.com/guide/topics/ui/index.html`

Android HTTP - `https://developer.android.com/training/volley/simple.html`

Android Sensors - `https://developer.android.com/guide/topics/sensors/sensors_overview.html`

Android UI Toasts - `https://developer.android.com/guide/topics/ui/notifiers/toasts.html`

Ionic - `https://ionicframework.com/`, `https://ionicframework.com/docs/components`, `https://ionicframework.com/docs/native`

5 Rubric for marking

Login and Logout	10
Menu	
Routing	5
Navigation	5
Video Games	
Dynamic	5
Layout	15
Refresh	
Layout	3
Refreshing	7
HTTP	20
Design and Aesthetics	10
Upload	
Not uploaded to ClickUP	**-80**
Bonus	5
Total	**80**

6 Uploading

Since Android studio builds multiple object files, you must only upload your **source code** as well as your **APK file**. No object/project files should be uploaded, your total upload zip should not exceed 50MB. If it does, omit the APK file, but ensure that you upload source code. You will need to upload to clickUP.
Note: If you do not upload you will receive 0 marks.

7 Assignment Instructions

NB: You may NOT use any web containers (e.g. WebView). Everything must be done in either Android native or Ionic. Violating this will cause you to get **zero** for this practical.

Task 1: Login and Logout ..... (10 marks)

Once a user opens your app they should be shown the login page. This page should contain 2 text-boxes, 1 for the username and 1 for the password, and a login button. Your app will test against this through your login API functionality you designed in PA4. You may modify your login functionality on the PHP side to cater for this if you have not implemented it correctly in PA4. Once a user logs out the generated API key from the login should be removed from memory.

You may also include a Splash page that appears before the login page when launching the app for bonus marks.

Task 2: Menu Tabs Navigation ..... (10 marks)

You will make use of the Android Bottom Navigation or Android Side Menu and have 2 tabs [Trending, Logout]. The Trending tab will be the default page shown once the user has logged in. If the user is not logged in, the login page should appear. For this you may use the "Android Bottom Navigation Activity" template as a jump start to get your application running.

Task 3: Video Game Layout ..... (20 marks)

Here you will need to mimic the PA4 design in Android. Your video game information should not be hard-coded, but rather added dynamically. Make sure to include the filter and search functionality as well. You must make use of native features and **you may not use a web container**. You do not need to display the Calendar like in the previous practical, but that can earn you bonus marks.

Task 4: Refresh ..... (10 marks)

The refresh functionality is used to retrieve the latest data from your PHP API followed by reloading the Trending page. This functionality is triggered using a pull down swipe on the phone to trigger the page data to be refreshed. You should also make use of a toast message to notify the user that the latest data has been fetched.

Task 5: HTTP and Data Manipulation ..... (20 marks)

As with any app, data needs to be requested from the server. Android has a simple approach to doing this. You will make use of HTTP POST and GET methods to fetch the data from your PHP API that you have developed. **Note:** Remember that each request needs to have the API key. You will also need to display any error messages or problems using Android UI toasts.

You may use any Andorid Native HTTP library or wrappers like OkHTTP (`https://medium.com/@sotti/android-networking-ii-okhttp-retrofit-moshi-and-picasso-c381f6c0efd8`). You may also make use of the native http request if you are using Ionic (`https://ionicframework.com/docs/native/http`).

Task 6: Design and Aesthetics ..... (10 marks)

Your app needs to look well-designed and be aesthetically appealing, therefore pay special attention to the following:

- Colour
- Font size
- Padding
- Material Design
- Overflow of text
- Responsiveness

- Use of UI elements

Task 7: Bonus ..... (5 marks)

In order to receive any marks here you need to have all the tasks and functionality implemented.

What you need to show here in order to get marks is additional functionality, mobile standardization and UI conformity or nice-to-have "wow" features. Simply having nice colours will not get you extra marks.

You can also earn extra marks for the following:

- displaying a Splash screen with cool animations on the app launch.
- including the Calendar part of the practical.
- creating local storage for the API key such that the user does not need to login every time the app is launched. The storage should be cleared when the user logs out.
- search functionality the same as the previous practicals and if the search is cancelled the view should not be reloaded, but the previously displayed data should be shown again.