
Assignment 3.0 Mobile Application Development

Overview

This is the first week of a multi-week assignment where you will be developing a mobile application either for the iOS or the Android platform. The application you will develop will basically be a portfolio application for your Github profile.

You can also choose the following:

- iOS application
- Android application
- [React Native](#) application

If you have never done mobile development before, I would suggest you use React Native for this assignment as it would be a huge timesaver.

Don't fall behind!

This week is the first of several for this assignment and contains several different technologies, many of which may be unfamiliar to you. Start early, make sure you understand the task and hand and get your development environment setup as soon as possible so that if you run into problems, you can get unblocked early.

The ramp up for this week will be longer than the previous assignments, so please start early!

You would have to run your application on an emulator or on your actual device. You can buy an Apple Developer Account for demo purposes for iOS development, else just using the emulator is fine.

Part I: Tutorials

Some great iOS tutorials:

- <https://www.raywenderlich.com/>
- <https://developer.apple.com/library/content/referencelibrary/GettingStarted/DevelopiOSAppsSwift/>
- <https://www.swiftbysundell.com/>
- <https://www.appcoda.com/>
- <https://www.udacity.com/>

Some great Android tutorials:

- <https://developer.android.com/training/basics/firstapp/index.html?hl=p>

-
- <https://www.youtube.com/playlist?list=PLB03EA9545DD188C3>

Some great React Native tutorials:

- <https://facebook.github.io/react-native/docs/tutorial.html>
- <https://www.toptal.com/ios/cold-dive-into-react-native-a-beginners-tutorial>
- <https://www.raywenderlich.com/165140/react-native-tutorial-building-ios-android-apps-javascript>

Part II: Create profile page, repositories, following, and followers page

For the first week, you would have to create the basic layout for your profile page and your repositories page. The profile page view should have a custom design and you **would be graded on your creativity**. For the repositories page, you would use a list/table view and you would be recycling each cell view of the list/table. Your pages should be accessible via tabs which are present at the bottom of your app.

Functional Requirements

The structure of your application is up to you, but it should contain the following data and should be able to **switch/navigate** pages easily (hint: tab bar navigation or drawer navigation):

Profile page:

- Profile avatar image view
- Name
- GitHub username
- Bio
 - Optional field, nice to have
- Website
 - Optional field, nice to have
- Email
 - Optional field, nice to have
- Public Repos count
 - Clickable: able to navigate to Repositories page
- Followers count
 - Clickable: able to navigate to Followers page
- Following count
 - Clickable: able to navigate to Following page

-
- Profile create date

All of these items should be properly formatted

Repositories page:

- A list/table of all your public repositories
 - Each item of the list/table should have:
 - Repository name
 - Owner's GitHub username
 - Repository description
 - Each item/cell of the list should be recycled/reused in a way
 - Clickable: able to open the GitHub webpage of the repo

Following page:

- An empty page, to be implemented next week (can do it this week as well for EC)

Followers page:

- An empty page, to be implemented next week (can do it this week as well for EC)

Assignment Project Exam Help

<https://powcoder.com>

Part III: Interacting with Github API

Since you already have some experiences in building an API in assignment 2.1, in this assignment, we require you to interact with the Github API.

Add WeChat powcoder

Documentation for the API here:

<https://developer.github.com/v3/> (REST)

<https://developer.github.com/v4/> (GraphQL - should only use this if you know what GraphQL is)

For fetching your own public GitHub user information, you may find this doc useful:

<https://developer.github.com/v3/users/>

Parse the information and display this information on the profile page you designed.

For repositories you can follow this link:

<https://developer.github.com/v3/repos/>

If you do it properly, you would get an array of repo JSON data. Parse the data and try displaying the information you need in a table/list in the repositories page you designed.

For this week, you are not storing the data anywhere in your application but you will just be fetching the data.

You can use the following libraries to make API requests:

- iOS - <https://github.com/Alamofire/Alamofire>
- Android - <http://square.github.io/retrofit/>
- React Native - <https://github.com/axios/axios>

Now, for this week as mobile development is kind of a new topic for many of you, you can just fetch the data but not display it in the application. If you do however manage to parse and display the data in your application, you will get extra credit!

Assignment Project Exam Help

We will be talking about storing data for next week, so it wouldn't be a bad idea for you to learn up a bit about iOS persistence storage / Android persistence storage / cloud storage solutions (e.g. Firebase)

<https://powcoder.com>

Add WeChat powcoder

Part IV: Adaptive Layout:

Your final task is to make the layout adaptive, which basically means your application should adapt to different screen sizes and orientations (portrait and landscape). If you look at tutorials closely above, some of them walk you through with it, so this should be an easy requirement.

Part V: Testing

As always, testing is an important part of this week's assignment, particularly for the parsing portion of the assignment.

You will create a manual test plan for this week. Write a test plan including screenshots and specific steps for a human tester to follow - what a tester should do and what he/she should observe

Summary

Table of Contents

- [Overview](#)
- [Part I: Tutorials](#)
- [Part II: Create profile page, repositories, following, and followers page](#)
- [Part III: Interacting with Github API](#)
- [Part IV: Adaptive Layout:](#)
- [Part V: Testing](#)

Reading

- The Pragmatic Programmer: Chapter 2
- Optional: Code Complete, Chapters 3 and 4

The readings are due before lecture as usual, on Oct 22nd, 2018.

Submission

This assignment is due at you discussion section for the week of Oct 22nd, **2018**. Please be sure to submit in Gitlab, and ask your moderator or TA before the deadline if you have any questions.

Objectives

- Create basic UI for a mobile application
- Interacting with a 3rd party API

Resources

See the links in tutorials above

Grading

Category	Weight	Scoring	Requirements
Basic Preparation	2	0-1	Ready to go at the start of section

Cleverness	2	0-2	The hardest points on the rubric
Code Submission	4	0-2	Submitted correct content on time and to the correct location in the repository
Decomposition	4	0-2	Project is adequately decomposed to different classes and methods
Documentation	4	0-2	Comments for each class and each function are clear and are following style guides
Effort	2	0-2	Perform considerable amount of work
Naming	2	0-2	Variable names and method names are readable and are following the naming conventions of the language you chose
Overall Design	5	0-2.5	Have nice approaches and structures in overall
Participation	5	0-2.5	Interact with the group 2 times (ask a question, make a comment, help answer a question, etc.)
Presentation	4	0-2	Present the code clearly
Requirements - Output	5	0-2.5	<ul style="list-style-type: none"> 2 points: Basic UI's for all the views have been defined. 2.5 points: Able to recycle views properly and have an application with tabs/drawer/etc. for quicker access to different views.
Requirements - Github API	5	0-2.5	<ul style="list-style-type: none"> 2 points: Ability to interact with the API without repeated calls 2.5 points: Parsed the data and was able to display the data in the application
Requirement - Adaptive	4	0-2	<ul style="list-style-type: none"> 1 point: Use some styling to make the application look clear and presentable. Use a nontrivial amount of styling 2 points: Modify the styling or theme to make those fit the different categories of pages, and make the application look nice, beautiful and adaptive
Testing	4	0-2	<ul style="list-style-type: none"> 0 points: No test plan 2 points: Manual test plan
Total	52		

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder