

CPSC x150 Mobile Device Software Development FA19

Homework 3 – 40 points

Work on this step-by-step. You may want to first write some simple Java code to test out the Dark Sky API and get back values before tackling the UI (or vice-versa).

Assignment Objectives:

- Plan, design, build, and test an Android app in Java that uses fragments for adaptive design.
- Implement a program that gets API calls using Volley
- Use a `ListAdapter` or `RecyclerViewAdapter`

Delivery Instructions:

- This assignment is due by 11PM on **Friday, October 11, 2019** and will not be accepted late.
- Upload a .zip file containing the following to appropriate assignment on Canvas:
 - Your Android project directory containing all source code.
 - A `lastname - readme .txt` containing your names, CID numbers, and precise directions on how to load and run your program.
 - Your Android project should be titled `team_number - hw3` where `team_number` is the team number of your submission group on Canvas.
 - One shot each of the portrait and landscape views in .jpg or .png format for competition.

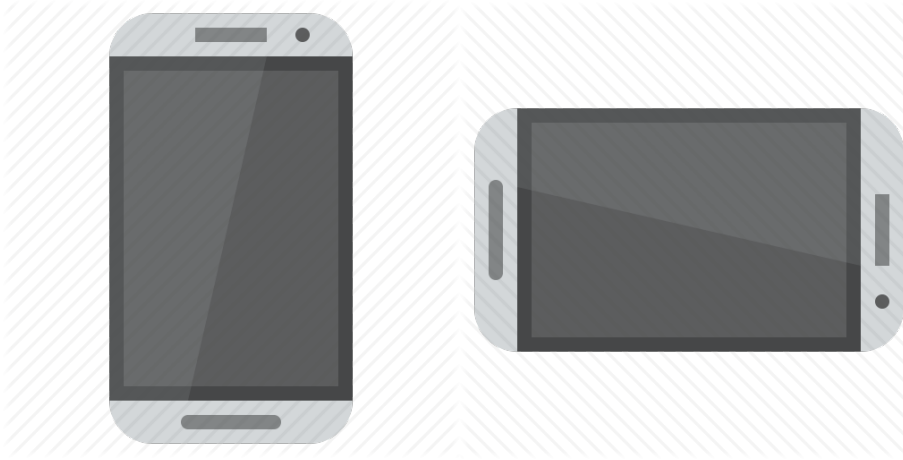
Approximate Grading Distribution:

- **Test Cases** (20 points): Checking to see if your program functions efficiently and correctly on an Android Emulator simulating a Google Pixel 2 device running Oreo 8.0. Code should compile cleanly (e.g., without errors or warnings). All code should be written in Java (not Kotlin). Project should meet specifications.
- **Proper Documentation** (5 points): Code should have consistent and proper indentation. Functions should have appropriate comments (including pre/post comments). Any code inspired by zyBooks or the official Android tutorials should have clear citations within your code comments (both in the header and where the code is used). Your name, CID, and email address should appear in a header in each file.
- **User Interface / User Experience** (15 points): App has a nicely-defined user interface as per guidelines discussed in class and/or readings.

Collaboration Policy:

- For this assignment, you may work with up to one other person in the class as was announced in-class and on Canvas. If you work with another person, this cannot be the same person that you are working with for the main project. The opportunity for collaboration is to combine brainpower to design, implement, and trouble-shoot.
- You may consult the following resources:
 - zyBooks Chapters 1 – 6 (and the associated sample code) (Participation Activity 6.9.4)
 - <https://developer.android.com/guide> (including Code Labs)
 - <https://docs.oracle.com/javase/8/docs/api>
 - <https://darksky.net/dev>
 - <http://getpostman.com> is a useful tool for visualizing API calls
- You may use Piazza for any of the following:
 - Asking conceptual questions about how your code functions (or should function)
 - Asking high-level questions related to debugging or errors (but clearly indicate in your post what you've already done to attempt to debug and/or trouble-shoot)
 - Referencing code from zyBooks, Android's Developer's Guide, or the Java API.
- As a student in a 4000/6000-level course, you should have ample experience in debugging object-oriented code. Expecting extensive one-on-one assistance from anyone else (course instructor, teaching assistant, classmate both in-person or online) is not reasonable.

The Project: Create a Responsive Weather App



Most smartphones come with some sort of weather app as part of the standard installation. For this project, you will create an account for the Dark Sky API (<http://darksky.net>). With a registered account, you will get 1,000 free API calls each day (resetting at midnight UTC). The following resources are very helpful on Dark Sky:

<https://darksky.net/dev/docs#api-request-types>
<https://darksky.net/dev/docs/libraries>

In this project, you will create the Blue Skies app that displays a list of *at least* 10 locations¹ to retrieve and display weather information. You will be creating this by using fragments in order to do adaptive design for a Pixel 2 phone held in orientation or portrait mode. To make this fun, choose locations that include Clemson, your home town(s), and then places that are on your bucket-list to visit!

You have leeway in this project to create an interface that is functional, user-friendly, and attractive. The TA and I will choose our top 5 designs and the class will vote on 1) best overall project functionality, and 2) best overall design for extra credit opportunities.

Specifications:

- You will need to use at least two fragments to make two *unique* user interfaces for whether the phone is in portrait or landscape.
- You should use JSON frameworks (Note, even though various wrappers for other methods are available, you should use the JSON/ Rest framework. See zyBooks participation exercise 6.9.4.)
 - It is critical for you to keep in mind the structure of the API response (e.g., the information returned uses a hierarchical format. You may want to download and use the application Postman to see how the information is structured)
- You need to have locations for at least 10 different geographic locations (e.g., there should be scrolling off of the home screen).
 - When a user clicks on any of the locations, a more detailed fragment should display. At a minimum, you want to have the location's name, current conditions, and current temperature. Feel free to go above and beyond.
- Your list should be implemented using a `ListAdapter` or `RecyclerViewAdapter` and having an individual item layout (e.g., each row should contain a "rowView" (you may use your own terminology) that contains the layout for the information in the row.
- Your readme.txt should also contain design justifications as well as credit for any code that's being borrowed or adapted from one of the approved resources.

¹ The goal of this project is to have more elements on the main screen than can fit, thus scrolling is required. If your UI allows all 10 required location to neatly fit on the screen, you must add additional locations in order to scroll multiple items off the screen.

Hints / Sage Advice:

- Make sure you read your zyBooks notes on the dependencies that need to be added to Gradle and your manifest in order to make Volley work.
- Troubleshooting internet connections via the emulator can be annoying. If you are having issues, check a browser on the emulator to see if it can make connections to sites. If you are having issues, check with your main computer/laptop connected to the internet in different ways (e.g., switch from wireless to a wired connection or a lab computer).
- Separation of concerns is helpful in designing code. You may want to follow a “real world” scenario of writing your Valley API calls separately and then integrate into your project later (and temporarily use test/stub hard-coding of values to test the interaction of the app.)

Extra Credit Opportunities:

- Implement a responsive design that has a *unique* layout for a Slate tablet. You will need to document your extra credit the same way in your readme file.
- Incorporate images to make the app look great!
- Any graphics used should be original or else permitted for use (typically with modification) royalty-free.
 - Please include sources for these images in your `readme.txt` file

Note, various choices related to graphical design (e.g., font selection, color choices, and layouts) are up to you. Just make sure you leverage good design practices as discussed in class.

Additional References:

<https://zyBooks.com>

<https://developer.android.com/studio/install>

<https://developer.android.com/studio/projects/create-project>

<https://developer.apple.com/design/tips/>

<https://jsoneditoronline.org>

<https://unsplash.com/license>

Sample Screenshot:

