

Competency

In this project, you will demonstrate your mastery of the following competency:

- Develop and launch a fully functional mobile application

Scenario

Your app UI design has been received positively by the client, which means you are now ready to proceed to the final stages of the development process. The client is eager to see the completed product ready for launch. To make this happen, you know that most of your time will need to be spent creating functional, error-free code that will run the app.

However, you also know there are further steps that must be completed after the coding in order for the app to be made widely available. To support your client with this part of the process, you will need to create a recommended launch plan that details how to successfully reach the client's intended users.

Directions

Begin by reviewing any feedback you received on your previous Project One and Project Two work. Next, open the Android Studio project where you developed the UI for your app in Project Two. Before you begin coding, be sure you have incorporated necessary changes to your UI design as this will help enable your success moving forward. You will be using the Android Studio Editor Window to view and write your Java code.

App Code Design

1. **Develop code that allows a user to log in.** As you develop your code, use the Android Emulator feature of Android Studio to test your code. Your app should work for users that both do and do not have accounts. This means you should focus on the following functionality:
 - The app should check the username and password against the database when the user attempts to log in.
 - If the user has never logged into the application before, the user should be able to create a new login and password. The application needs to save these to a table in the database.
2. **Develop code to create a database shell and display that database's information on a user's screen using a grid.** The SQLite database you create will depend on the option you selected in Project One and will store either inventory items, event details, or daily weight. Any information needed for your application to function correctly should be held in the database, but remember you are only creating the shell for the database so

its contents can later be populated by a user. Note that you will need tables in the database to store user information when it is not being displayed in the grid. This database will be persistent so no user information is lost when the app is closed. Be sure to test your code often with the Android Emulator. Your completed database should allow a user to accomplish the following:

- Create: The user should be able to add items to a database.
- Delete: The user should be able to remove items from a database.
- Update: The user should be able to change the value associated with individual database items (e.g. the number of a specific item in an inventory or the date of an event).
- Read: The user should be able to view all of the database items displayed as a grid.

3. **Develop code that prompts users for permission to send SMS messaging notifications and makes the app function based on the individual user's response.**

Keep in mind that the user of your application might deny access to the permissions needed for sending text messages via SMS. If they do, your application should still continue to function, just without this feature. Test both possible options for user response by using the Android Emulator. When you do, you will be looking for the following outcomes:

- If the user grants permissions, the application should send alerts to the user as SMS messages. The alerts correspond to the specific notification trigger of the application you chose (low inventory, an upcoming event, or reaching a goal weight).
- If the user denies permission, then the rest of the application should still continue to function without the SMS messaging notification feature.

4. **Employ industry standard best practices such as in-line comments and appropriate naming conventions to enhance readability of code.** This should be evident throughout all of the code you create for this project and will be assessed comprehensively. Some things to ask yourself as you code are:

- Have I kept my classes concise?
- Is my style consistent throughout the code?
- Would my naming conventions make sense to anyone else who looked at my code?
- Do my in-line comments contain enough useful information?

App Launch Plan

5. **Determine a plan for launch of the application that outlines all necessary components to support your app.** Once you have completed the code for your app, there are a few more steps required to bring your app to the market. In this project you will not actually have to launch your app in the app store but you will need to outline a comprehensive plan for launch that will prepare you to bring your app to market. Later in this course you will have the opportunity to officially launch your app, if you choose,

since it will be an excellent accomplishment to share as you progress along your professional pathway. For now, address the following questions as you write your plan:

- What will be included in your app's description and what kind of icon will best represent your app once it is made available in the app store?
- Which version(s) of Android will your app successfully run on? Have you included the most current version? Note that with each version of Android, new components are introduced that add considerations and challenges in the development.
- What permissions will your app ask for? Be sure these are only permissions that are necessary for your app to run. For example, does your manifest ask for permission to record phone audio when your app does not use it?
- What is your plan for monetization of the app? Consider whether your app will include ads, require a one-time payment, both, or neither.

What to Submit

To complete this project, you must submit the following:

App Code Design

Submit a ZIP file of the completed Android Studio project folder for your fully functional, error-free application. Your professor will grade your work in the ZIP file but you may also compile and submit an APK file. This would be useful to you because if there are any issues with the code, Android Studio will not generate the APK file. Therefore you can use the APK file to check your work and demonstrate overall functionality to your instructor.

App Launch Plan

Submit your completed launch plan as a Microsoft Word document of 2-3 pages in length. Use double spacing, 12-point Times New Roman font, and one-inch margins. The file name should include your name and the app option you have been using for the projects.