Android Fundamentals Project Description

This document is intended for students enrolled in the Full Course Experience. Students who are taking the free courseware are encouraged to complete the project on their own, but will not submit their work to Udacity for assessment.

For this project, you will make your own app. You can either come up with your own idea or select from one of the ideas listed under “App Ideas.” You are welcome to repurpose code from your Sunshine app to create your new app.

Before you embark on this project, make sure to review the Project Rubric at the end of this document to understand how your work will be assessed.

This is what you will do (for students enrolled in the Full Course):

1. Pick an app to build, and draft a short (less than one page) summary of what the app does, and what problem it solves.
2. Create mocks for all user-facing screens, and at least one alternate mock for tablet screens. Hand-drawn sketches are OK, but they must be in a digital format that you can share with your Coach.
3. Build your app!
4. Zip source code and all accompanying documents.
5. Email your file to androidapps-project@udacity.com.

<u>Project Requirements</u>

Your Project must:

- Include a problem description of the problem your app solves.
- Include mocks for all user-facing screens.
- Include at least one alternate mock for tablet / large screens
- Have at least two distinct screens.
- Work properly with the app lifecycle (rotate screen changes)
- Use permissions responsibly.
- Use Intents to move inside your app or to an outside app.
- Create and use your own ContentProvider

For extra Udacious-ness, include at least two of the following (optional):

- Receive Broadcast events and do something meaningful
- Create and use a Custom View
- Implement ShareActionProvider
- Implement a SyncAdapter
- Use Notifications

App Ideas (or substitute your own):

- Scientific / simple calculator
- Food ordering app
- Teleprompter app
- Local news app
- App for your company, university or local club
- Education app
- Public transit schedule

Helpful Resources

If you're looking for action bar icons, you can download the pngs for different screen densities for holo light and holo dark themes from the Android Design Guide downloads page. The Adobe Illustrator source files are also included if you want to style them to match your app.

Action bar icon pack download: https://developer.android.com/design/downloads/index.html

Helpful tools to create assets: http://romannurik.github.io/AndroidAssetStudio/

Other resources:

- http://developer.android.com/distribute/essentials/quality/tablets.html
- http://androidniceties.tumblr.com/
- http://android-developers.blogspot.com/

Final Project Rubric

Overview

This rubric is here to help you understand the expectations for how your project will be evaluated. It is the same rubric that the person evaluating your project will use. You should look at the rubric **before you begin working** on this project **and before you submit it**.

The criteria included in this rubric are a subset of the Core App Quality Guidelines that your app would be evaluated against if you submitted it to the Google Play Store. You can find the full list of guidelines here: http://developer.android.com/distribute/essentials/quality/core.html

Before You Submit

1. Go through each rubric item below and do your best to honestly evaluate where you think your project falls.
2. If you think your project "does not meet specifications" for any criteria item in the Required Components section, you should make any necessary changes.
3. See the “How Grading Works” section below. Once you’re confident that your project "meets specifications" or "exceeds specifications," you can submit by emailing your Zip file to androidapps-project@udacity.com.

How Grading Works

1. Your project evaluator will be able to see all of your code submissions. They will use this rubric to evaluate your code as well as your written responses.
2. Your grade will simply be "pass, meets specifications," “pass, exceeds specifications,” or "doesn’t pass,"
    a. You earn “pass, meets specifications” if **all** criteria in the Required Components section “meet specifications”.
    b. You earn “pass, exceeds specifications” if **all** criteria in the Required Components section “meet specifications” and you receive “exceed specifications” in all criteria under at least 2 of the 4 categories of the Optional Components section.
    c. Your project “doesn’t pass” if **any** criteria in the Required Components section are graded as “does not meet specifications.” In this case, you will have the opportunity to revise and resubmit your work.

The Rubric

Required Components

To “meet specifications”, your app must fulfill all of the criteria listed in this section of the rubric.

Criteria	Does Not Meet Specifications	Meets Specifications
Standard Design		
App does not redefine the expected function of a system icon (such as the Back button).		
App does not replace a system icon with a completely different icon if it triggers the standard UI behavior.		
App does not redefine or misuse Android UI patterns, such that icons or behaviors could be misleading or confusing to users.		
Navigation		
App supports standard system Back button navigation and does not make use of any custom, on-screen "Back button" prompts.		
All dialogs are dismissible using the Back button.		
Pressing the Home button at any point navigates to the Home screen of the device.		
Permissions		
App requests only the absolute minimum permissions that it needs to support core functionality.		
App does not request permissions to access sensitive data or services that can cost the user money, unless related to a core capability of the app.		

Performance and Stability		
App does not crash, force close, freeze, or otherwise function abnormally on any targeted device.		
ContentProvider		
App retrieves and caches data from a server using a ContentProvider.		
App updates data in its cache at regular intervals using a SyncAdapter.		
User/App State		
App correctly preserves and restores user or app state.		
When the app is resumed after the device wakes from sleep (locked) state, the app returns the user to the exact state in which it was last used.		
When the app is relaunched from Home or All Apps, the app restores the app state as closely as possible to the previous state.		

Optional Components

To receive “exceeds specifications”, your app must fully implement all of the criteria listed under at least two of the four categories below (e.g. Notifications, ShareActionProvider, Broadcast Events, and Custom Views).

Criteria	Does Not Exceed Specifications	Exceeds Specifications
Notifications		
Notifications do not contain advertising or content unrelated to the core function of the app.		

Notifications are persistent only if related to ongoing events (such as music playback or a phone call).		
Multiple notifications are stacked into a single notification object, where possible.		
App uses notifications only to indicate a context change relating to the user personally (such as an incoming message).		
App uses notifications only to expose information/controls relating to an ongoing event (such as music playback or a phone call).		
ShareActionProvider		
Uses ShareActionProvider to share content with an outside application.		
Makes use of Intent Extras to send rich content.		
Broadcast Events		
App intercepts broadcast events.		
App responds to Broadcast events in a meaningful way.		
Custom Views		
App creates and uses a custom View.		
App uses a novel View that couldn’t sufficiently be satisfied by the core Views in Android.		