CS230 Developing Mobile Apps
Final project specifications
Due: Friday December 13th 2019 at 11:59pm
Prof. R. Butler, Johnson 114a, Office Hours MWF 9:00-11:00am

Some important comments:

1) The final project can be developed on your platform of choice (Android, Unity, Flutter, iOS, React Native, etc.), but I recommend using Android. Multi-platform solutions typically don't perform as well and are declining in popularity (Lecture 1, Slide 30).
2) The final project should be published on the Google Play Store (Assuming you use Android) by the due date, and you must send your Android Studio project by the same date. If you feel your project has long term potential and would like to keep the project to yourself/wait longer to publish, you may send me a preliminary version via .apk file and some code samples, and we will go over the app in-person.
3) Max group size has been increased to 3 people for the final project.

Evaluation:

Originality/usefulness (33%) – your app should be both original and perform some useful function. Do not simply create a notepad app, or a calculator app. Your app should fill some niche, so it doesn't get buried in the app store under the thousands of other similar apps. If you do replicate some existing app, it should not be something trivial.

Use of best practices, look and feel (33%) – if developed in Android, your app should make use of the concepts learned in the course. The app should be robust to device configuration changes and back button presses, UI/navigation should be smooth and intuitive, and the app should not crash. The app should be aesthetically pleasing, and have an overall theme. The app should have a settings screen where the user can adjust relevant parameters.

Monetization/App store presence/User data collection (33%)
Monetization - implement one of the following monetization strategies, or a combination of the three:

a) In-app purchases
b) Banner add
c) Interstitial add

App store presence – The app store presence will be optimized to maximize the number of downloads the app receives (app store optimization, we will discuss this in a lecture). Briefly: keywords, screenshots, description, icon.
Authentication - The app should also implement some type of login/authentication mechanism (facebook login, gmail login, etc.), the app will send data to an online database (Firebase) and store information about the user.

The complexity of the app is up to you, but rough guidelines:
~1000-2000+ lines of code, 5+ separate classes
At least 2-3 separate activities, each performing their own function
Make use of Service, Broadcast, and/or Content Provider
Make use of online database (Firebase) to store/synchronize data across different users and devices

Note – do not need to fulfill all the above, but try to get most of them.