**ECE 1778 - Creativity and Programming for Mobile Devices**
**February 2016**
**Programming Assignment P3**

# Mobile App Backends, Images, and Location

Mobile applications are increasingly web-connected and often require a server to authenticate users of an app, to store and backup files, and to send messages. The server and the set of software and APIs that perform these tasks are collectively called a *backend*. In this assignment you will set up and use a pre-made backend designed for use by mobile applications. You will be creating a photo-taking app that both provides a photo gallery on the phone, and sends copies of the photos to the server. In addition, you'll be learning how to use the location (GPS) services.

Also, please note, that **you will also be asked by your Specialist partners for instruction** on an aspect of computer engineering or science, relating to one of Search/Indexing, Databases, Digital Signal Processing, Optimization or Internet Communication.

## 1 Mobile "Backend as a Service"

The design and maintenance of a backend is a large amount of work, but fortunately many companies provide it as a service – this is called "Backend as a Service" or BaaS. Here, developers are given a pre-made backend that provides the necessary functions. In this assignment, you will be setting up and using a free account to Google's BaaS, called **Firebase**.

Go to https://firebase.google.com/ and click "Get Started for Free" to do so. To do this you will need a (free) Google account, so you will be asked to make one if you don't already have one. Once this is done and you are in the Firebase console, you should see the option to create a new project.

## 2 Reading & Learning

There is quite a bit of reading and learning to do in this assignment. The following links point you to documentation and setup procedures for both Firebase, and the relevant topics for the mobile device. You will need to choose what to read and do carefully to make good use of your time:

1. **Basic Firebase**
   Android: https://firebase.google.com/docs/android/setup
   - You will need to add the "Core", "Database", "Auth", and "Storage" libraries

iOS : https://firebase.google.com/docs/ios/setup

- You will need to add the "Core", "Database", "Auth", and "Storage" pods

2. **Specifics of Firebase storage, authentication and database.**
   i. https://firebase.google.com/docs/storage/
   ii. https://firebase.google.com/docs/auth/
   iii. https://firebase.google.com/docs/database/

3. **GPS Location Service/Sensor**

Android:

i. The Chapter titled "Accessing Location-Based Services" of the **The Busy Coder's Guide to Android Development** version 7.6. You may also find the next chapter on "The Fused Location Provider" worthwhile as well.
ii. Read through the Firebase guides on Storage, Realtime Database, and Authentication

iOS: from either **Beginning iPhone 7 Development Exploring the iOS SDK** (if you're using Objective C) or **Beginning iPhone Development with Swift 2** (if you're using Swift):

i. Chapter 19, "Where Am I? Finding Your Way with Core Location and Map Kit"
ii. Chapter 21, "The Camera and Photo Library."

## 3 Assignment

*NOTE: in writing your code for this assignment, please be sure to follow 'Braiden Brousseau's Guide To Quality Apps' that was given as part of Assignment P1. 20% of your grade will be assigned for fulfilling these guidelines.*

You are to write an application that will allow a user to take, view and delete pictures. All pictures taken by the app are to be copied up to a Firebase-based Storage server.

The main screen should be a photo gallery and it should display *thumbnails* (small versions) of all the photos stored on the device's DCIM folder *and* the GPS locations where the photos were taken. The DCIM folder is the traditional location for pictures and videos when mounting the mobile device as a digital camera. Note that there may already be photos in the device's DCIM folder and these must be displayed as thumbnails as well. Note also that photos in the gallery that were not taken by this application may not have associated location data, so the location can be missing in that case.

This photo and location data should be available over separate invocations of the app.
The main screen should have the following two buttons which should behave as follows:

1. Camera button: This should launch the device's camera and allow the user to take and save a picture. The picture must be saved to the device's DCIM directory and also uploaded to Firebase Storage.
2. Restore button: This should download all pictures that exist in the Firebase Storage, but not locally on the device, back to the device's DCIM directory. Only photos taken by this app need to be restored. This means that there is no need to upload pre-existing photos in the DCIM directory that were not created by this app.

The second screen of the app should be a detailed view mode that is invoked any time a thumbnail from the gallery is selected (i.e. touched). In this detailed view, the image that was selected from the gallery should be shown in fullscreen mode. Additionally, there should be a **Delete** button on the screen, with the following functionality: this button deletes the photo from the device. It should also ask the user if they wish to delete the backup that resides on Firebase Storage. If they only delete the local copy, they should be able to recover it through use of the Restore button. Note that only photos taken by this app need to be uploaded/downloaded/deleted from Firebase Storage.

You can see a video of a working version of the App here:
https://www.dropbox.com/s/iq8dyij3mf5env3/ece1778_p3_demo.mp4?dl=0

**Due date**: Thursday October 20th, 6pm, marked out of 10, 0.5 marks off every hour late. Submit your solution on the Blackboard portal, under the **Programmer Assignments** link and **Assignment P3**.

What to submit:
1. Android developers: a zip file containing your final Android application file (.apk); use your student number as the filename. Also submit the complete Android Studio project directory in a separate zip file.
2. iPhone developers: you must submit the complete project directory, including source, in a zip file. Use your student number as the filename. Please do your development on the Version 8.0 of the SDK, and make sure that you haven't included any files by reference. Please test your submitted zip file before sending it in.

## 4 Frequently Asked Questions

**1. I'm an Android programmer. Can I launch a camera intent to take new photos, or must I use the camera API directly?**

Yes, you can take photos simply by using an intent.

**2. I understand why we are using Firebase Storage, but why did you tell us to set up and read about Authentication and the Realtime Database?**

Authentication is necessary to upload files to Firebase Storage. Don't worry, you don't need to setup user login and management, just authenticate an anonymous user in the background without asking the person using the app the setup an account. See https://firebase.google.com/docs/auth/android/anonymous-auth

The Realtime Database is necessary to store the path of all the images that exist in Storage. This is because the Storage API doesn't allow you to simply list all the files that exist there – in order to get a reference to a file in Storage you need to know where it is beforehand. The workaround here is to maintain a piece of information (the file path) in the database for each image you upload to storage. Then, when you want to download or delete images from storage, you simply query the database for all image paths and then use those to get StorageReferences. Once you have the StorageReference for an image file, you can download or delete that image from Storage.

You could save the URLs using another method that is persistent (file saved locally on the device, or database running locally on the device), but the method described above makes more sense, for example, if you were planning on designed a companion website to view the pictures on the web.

**3. I set up anonymous authentication in my app but I still can't upload an image to storage. What's wrong?**

You also need to configure Firebase to allow incoming anonymous users. To set this up, do the following:

- Go to the firebase web console https://console.firebase.google.com/ and select your project
- Select "Auth" from the left-hand side (under "DEVELOP")
- Click the "SET UP SIGN-IN METHOD" button
- Click "Anonymous", slide the slider next to "Enable", and hit "SAVE"