Frank Eslami
CS 496, Final Project

APP DOCUMENTATION

App's Demo Video Link
Please download the mp4 video titled, Frank Eslami Final Project Video Demo.mp4, from the following public link:
http://web.engr.oregonstate.edu/~eslamif/cs496/final-project/

App Name
Travel Memory

App Description
Imagine going on vacation to Alaska. You've never been there before, but always wanted to go to check out the beautiful sky and go deep ocean fishing. Your friends and family are envious that you have this opportunity and want to hear all about your trip when you return. How will you remember the most memorable moments, moments like the first time you smelled the Alaskan air, the time you saw a whale swim up right next to your little six foot boat, or the time when a shooting star waved its arc in the sky, as if a sign was given directly to you? Travel Memory allows you to capture these memorable experiences within seconds. Here is how it works.

Whenever a registered user wants to record an experience, they type a brief description of the experience and save it in order to reminisce or share it later. They can also include pictures or documents from their gallery, such a menu item, fliers, or a picture of a bird. Their information is saved on a database for their viewing pleasure later.

Future enhancements will include geolocation and a date stamp, so they can map their travel, and the ability to export the data in scrapbook format.

App Specifications
```
Android SDK
compileSdkVersion23
buildToolsVersion"23.0.2"
applicationId"com.cs496.eslami.assignment_4"
minSdkVersion8
targetSdkVersion23
versionCode1
versionName"1.0"
```
Phone: Nexus 5 API 21 x86 2GB

Assignment Requirements Met
The following assignment requirements were met:
1. Create a mobile app with cloud backed feature: I created a native Android app that reads/writes to a Non-relational database, RESTful database called Firebase.

2. User account registrations: the app forces users to register in order to only view their own content. Add, delete, and modify data features are implemented for each user.
3. Entities & relationships: Users are registered with an email and password. For each user, a text input called representing memories and file can be saved. A user can have many files associated with them, so a one-to-many relationship exists.
4. Mobile front-end: several mobile features were used: Android's camera, file system, and Gallery application, which was called through an intent.
5. Debugging: The following features were implemented: once logged out, you could not hit the back button to go back. Emails have to be in correct email format to be registered. Logging in as a different user properly shows the correct content.

Database

I used Firebase with its non-relational, RESTful features. Add, remove, and modify features were done as follows:

Add
```
Firebase pushRef = new Firebase("https://travel-memory.firebaseio.com/users/" +
userKey);
pushRef.push()
     .child("text")
     .setValue(text.getText().toString());
```

Delete
```
new Firebase("https://travel-memory.firebaseio.com/users/" + userKey)
          .orderByChild("text")
          .equalTo((String) listView.getItemAtPosition(position))
          .addListenerForSingleValueEvent(new ValueEventListener() {
               public void onDataChange(DataSnapshot dataSnapshot) {
                    if (dataSnapshot.hasChildren()) {
                         DataSnapshot firstChild =
dataSnapshot.getChildren().iterator().next();
                         firstChild.getRef().removeValue();
                    }
               }
```

Modify
```
final Firebase mylist = new Firebase("https://travel-memory.firebaseio.com/users/" +
userKey);
mylist.orderByChild("text")
          .equalTo((String) listView.getItemAtPosition(position))
          .addListenerForSingleValueEvent(new ValueEventListener() {
               public void onDataChange(DataSnapshot dataSnapshot) {

                    DataSnapshot firstChild =
dataSnapshot.getChildren().iterator().next();
                    //Update Firebase
                    Firebase dataRef = firstChild.getRef();
                    dataRef.child("text").setValue(userInput.getText().toString());

               }
```

User Accounts

Firebase's OAuth user account authentication was utilized in order to register users. Once each user was registered, Firebase created a unique Uid for that user. User data was saved to the database under each user's Uid. That is how I was able to provide content appropriately to each user.

References

https://www.youtube.com/watch?v=BXTanDpOTVU
https://www.firebase.com/docs/android/quickstart.html
https://cloud.google.com/solutions/articles#mobile