Firebase

What is Firebase?

- Firebase is a cloud services provider and backend as a service company based in San Francisco, California
- Used to maintain app’s backend
  - Data storage
  - User authentication
  - Static hosting

Setting Up Firebase

- Sign up for an account on Firebase
- Add the following gradle dependency
  - compile 'com.firebase:firebase-client-android:2.5.2+’
- Add the permission
  - <uses-permission android:name="android.permission.INTERNET" />
- Add this line to your onCreate()
  - Firebase.setAndroidContext(this);

How data is stored

Data is stored in JSON Format

```
{ user :
          { posts:
               { post0:
                    { title: “Title1”,
                      description: “Description1” },
                 post1:
                    { title: “Title2”,
                      description: “Description2”} } }
```

Authentication

- Firebase can authenticate with Google, Facebook, Twitter, or an email/password
- When authenticating with an email/password, you can store in SharedPreferences an authentication token which can be used later to restore a session even after the app is killed
- If you are creating a new user, you need to register that user and then log the user in
- It is really easy to logout of a session by using the unauth() function, and resetting your session token in SharedPreferences to “”

Security Rules for Email/Password Authentication

```
{
  "rules": {
    "users": {
      "$uid": {
        // grants write access to the owner of this user account whose uid must exactly match the key ($uid)
        ".write": "auth !== null && auth.uid === $uid",

        // grants read access to any user who is logged in with an email and password
        ".read": "auth !== null && auth.provider === 'password'"
      }
    }
  }
}
```

Inserting New Data

- To insert new data you can either use an object class or a map
- Once you create the object, you navigate your Firebase reference to the position where you want to add a child
- If you are creating a list and do not have specific names for each child, you can use the push() method before you setValue()
- Call: ref.push().setValue(object) or ref.setValue(object)

Updating Data

- Navigate your firebase reference to the parent of the item that you want to update
- Create a map containing the update values
  - You do not have to put all of the children's data in the map again, only the values that you want to update
- Call ref.updateChildren(map)

Removing Data

- Navigate your firebase reference to the item that you want to remove
- Call ref.removeValue()