

Figure 1

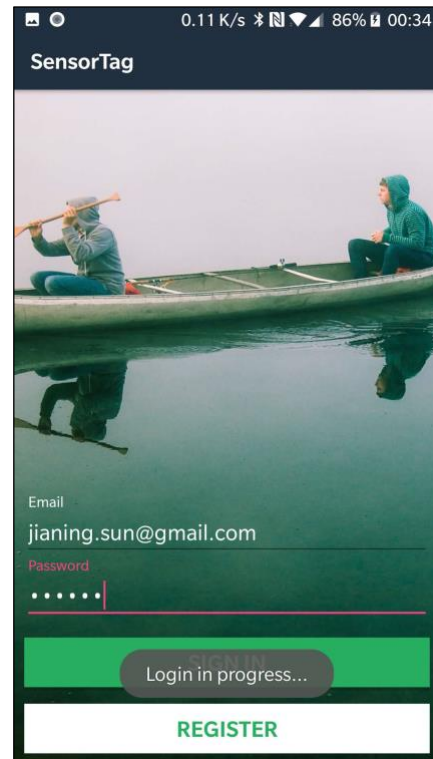


Figure 2



Figure 3

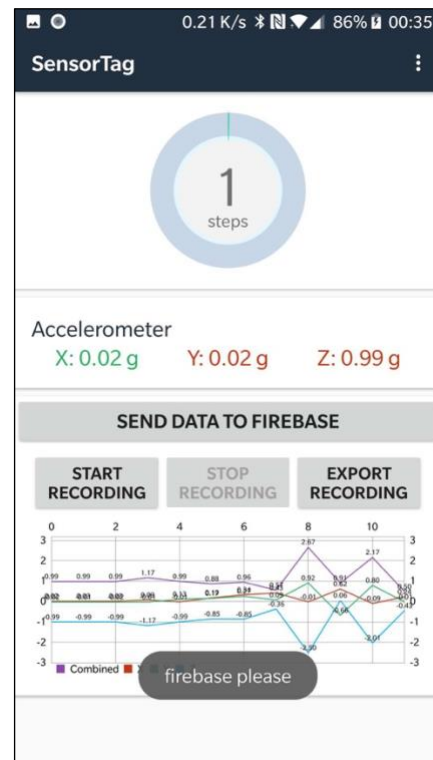


Figure 4

- **Firebase Authentication**

Most apps need to know the identity of a user. Knowing a user's identity allows an app to securely save user data in the cloud and provide the same personalized experience across all of the user's devices.

Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more.

In my case, I enabled "email" with address and name in order to identify among users, eventually establish Firebase authentication. Figure 1 illustrates my user identity – jianing.sun@gmail.com and password is registered before login, or you can also add a user with an email address and name in your Firebase Console. Then once I enter correct email address and its corresponding password, there would be a progress bar displaying in the screen with "Login in progress...". Then the app will navigate to another activity, which is basically inherited from assignment 2.

- **Real-time Database**

Store and sync data with our NoSQL cloud database. Data is synced across all clients in real-time and remains available when your app goes offline. The Firebase Real-time Database is a cloud-hosted database. Data is stored as JSON and synchronized in real-time to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of your clients share one Real-time Database instance and automatically receive updates with the newest data. Below is a snippet of my code about the key idea of storing data:

a) Use SharedPreferences to save my accelerometer data acquired from SensorTag:

```
private SharedPreferences pref;
private DatabaseReference mDatabaseReference;
private FirebaseAuth mAuth;
...

@Override
    public void onCharacteristicRead(BluetoothGatt gatt, BluetoothGattCharacteristic
characteristic, int status) {
    super.onCharacteristicRead(gatt, characteristic, status);
    // convert raw byte array to G unit values for xyz axes
    result = Util.convertAccel(characteristic.getValue());

    pref = getContext().getSharedPreferences("Firebase", Context.MODE_PRIVATE);
    SharedPreferences.Editor edit = pref.edit();
    edit.putFloat("X value", (float) result[0]);
    edit.putFloat("Y value", (float) result[1]);
    edit.putFloat("Z value", (float) result[2]);
    edit.apply();

    ...
}
```

- b) Set the click listener for *SEND DATA TO FIREBASE* Button then implement data store function (upload data to Firbase):

```
@Override
public void onClick(View view) {
    switch (view.getId()) {
        case R.id.bFirebase:
            mDatabaseReference = FirebaseDatabase.getInstance().getReference();
            pref = getContext().getSharedPreferences("Firebase",
Context.MODE_PRIVATE);

            float x_value = pref.getFloat("X value", 0);
            float y_value = pref.getFloat("Y value", 0);
            float z_value = pref.getFloat("Z value", 0);
            Toast.makeText(getContext(), "Fire|ebase", Toast.LENGTH_SHORT).show();
            Log.d("FirebaseDebug", String.valueOf(x_value));
            Log.d("FirebaseDebug", String.valueOf(y_value));
            Log.d("FirebaseDebug", String.valueOf(z_value));
            // save data to firebase
            mDatabaseReference.child("Accelerometer:X axis").push().setValue(x_value);
            mDatabaseReference.child("Accelerometer:Y axis").push().setValue(y_value);
            mDatabaseReference.child("Accelerometer:Z axis").push().setValue(z_value);

            break;
            ...
    }
}
```

Then you see the acquired accelerometer data with X, Y, Z axes respectively in the Console of Firebase. Below is the snapshot of my uploaded data from Sensortag to Firebase:

<https://sensortagdata-bda7b.firebaseio.com/>

★ Default security rules require users to be authenticated.

[LEARN MORE](#)[DISMISS](#)

sensortagdata-bda7b

Accelerometer: X axis

- L8LGvXm2TH9l0OX7P20: -0.01123046875
- L8LGvck7jNwszhFaHf6: -0.007568359375
- L8LGvfSLeOc0CyANQ10: -0.0087890625
- L8LGvi2Z1G6-l4LE7dL: -0.0107421875
- L8LGvjU2M-cRfeHZOEs: -0.004638671875
- L8LHVyBCYlrdipVJu9W: 0.02490234375
- L8LHWdQri5nljlLsz4z: 0.02001953125

Accelerometer: Y axis

- L8LGvXobN32uYH93AUH: 0.17626953125
- L8LGvcluxCsqe4fzYK6: 0.174072265625
- L8LGvfT5_u_0au-W42L: 0.1728515625
- L8LGvi2Z1G6-l4LE7dM: 0.177734375
- L8LGvjU2M-cRfeHZOEt: 0.17822265625
- L8LHVyDWVPwl2Ao5jl9: -0.011962890625
- L8LHWdRg_QYlq0dkeEc: -0.012451171875

Accelerometer: Z axis

- L8LGvXobN32uYH93AUI: -0.983642578125
- L8LGvcluxCsqe4fzYK7: -0.969482421875
- L8LGvfUrOS_GAEL6lXw: -0.973388671875
- L8LGvi3opz_B5M8qd3D: -0.96923828125
- L8LGjVthLg9phJEKHw: -1.023193359375
- L8LHVyFVx_VV21jdegY: -0.990966796875
- L8LHWdS6RtYbukEOdUg: -0.98974609375