

Lab 5 Instructions

Authors: Junwen Shen (junwen5@ualberta.ca), Meharpreet Singh Nanda (meharpre@ualberta.ca)

Step 1: Preparations

1. Download the starter code from eClass. The code should look similar to Lab 3.
2. Open the project in Android Studio.

Step 2: Create a Firestore project

1. Go to [Firebase Console \(https://console.firebase.google.com/\)](https://console.firebase.google.com/) and sign in with your Google account.
2. Click "Create a project", and follow the steps given below.
 1. Give it a name, ListyCity, for example.
 2. Disable **Google Analytics** for the lab project.
 3. Create it.
3. Click "**Continue**" when the process is finished.

Step 3: Add Firebase to the Application

1. Under "**Get started by adding Firebase to your app**", click on the Android icon.
2. Enter the package name. In this lab, it's `com.example.listycity5`. You can also find it on the first line of `MainActivity.java`. Click "**Register app**".
3. Download `google-services.json`.
4. Move `google-services.json` to the `ListyCity/app/` folder.
5. In Android Studio, go to **Gradle Scripts**.
 1. Open `build.gradle.kts` (`project: ListyCity`), and add `id("com.google.gms.google-services")` version "4.4.0" apply false after line 3.
 2. Open `build.gradle.kts` (`Module :app`), and add `id("com.google.gms.google-services")` after line 2. Then add `implementation(platform("com.google.firebase:firebase-bom:32.7.1"))` and `implementation("com.google.firebase:firebase-firestore")` after line 35.
 3. Sync the project.
6. Go back to the browser and click "**Next**".
7. Click "**Continue to console**".

Step 4: Create a Firestore Database

1. On the left, under "**Product categories**", choose "**Build → Firestore Database**".
2. Click "**Create database**".
3. Choose "**Start in test mode**" and click "**Next**".
4. You should now see an empty database.

Step 5: Add a New City

1. Update `activity_main.xml` to add two `EditText`s and a `Button` to get the name of the city and province from the user. NOTE: You can customize the layouts in any way you want.

```

<!--Other code here-->

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    >
    <EditText
        android:id="@+id/city_name_edit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="7"/>
    <EditText
        android:id="@+id/province_name_edit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="7" />
    <Button
        android:id="@+id/add_city_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Add City"/>
</LinearLayout>

<!--Other code here-->

```

2. Declare the attributes for the EditTexts and Button we added and initialize them with their respective objects in the onCreate method of MainActivity.

```

private Button addCityButton;
private EditText addCityEditText;
private EditText addProvinceEditText;

protected void onCreate(Bundle savedInstanceState) {
    // other code here

    addCityEditText = findViewById(R.id.city_name_edit);
    addProvinceEditText = findViewById(R.id.province_name_edit);
    addCityButton = findViewById(R.id.add_city_button);

    // other code here
}

```

3. Add a click listener to the addCityButton to read the EditTexts and make a new City object.

Step 6: Utilize Firestore Database

1. Remove the hard-coded data in the list (Remove the call to the addCitiesInit method in onCreate method of MainActivity).
2. Add Firestore instance in MainActivity.java.

```

private FirebaseFirestore db;

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    db = FirebaseFirestore.getInstance();

    cityDataList= new ArrayList<>();
    // Other code omitted
}

```

Step 6.1: Add data

1. Get a collection reference, in this lab, we store all data in a collection called “cities”.

```

private FirebaseFirestore db;
private CollectionReference citiesRef;

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    db = FirebaseFirestore.getInstance();
    citiesRef = db.collection("cities");

    cityDataList = new ArrayList<>();
    // Other code omitted
}

```

2. Create a method to add the new city to the collection. **Call** this method from the click listener for addCityButton added in step 5.3.

```

private void addNewCity(City city) {
    // Add the city to the local list
    cityDataList.add(city);
    cityArrayAdapter.notifyDataSetChanged();

    // Add the city to the Firestore collection with the city name as the document Id
    HashMap<String, String> data = new HashMap<>();
    data.put("Province", city.getProvinceName());
    citiesRef.document(city.getCityName()).set(data);
}

```

3. Optionally, you can add listeners for logging while saving a city. You can find more listeners in the [API references](https://developers.google.com/android/reference/com/google/android/gms/tasks/Task.html) (<https://developers.google.com/android/reference/com/google/android/gms/tasks/Task.html>).

```

citiesRef
    .document(city.getCityName())
    .set(data)
    .addOnSuccessListener(new OnSuccessListener<Void>() {
        @Override
        public void onSuccess(Void aVoid) {
            Log.d("Firestore", "DocumentSnapshot successfully written!");
        }
    });

```

Step 6.2: Get real-time updates

1. Add a `SnapshotListener` in `onCreate()` method for updating `cityDataList` in real time with the Firestore database.

```

citiesRef.addSnapshotListener(new EventListener<QuerySnapshot>() {
    @Override
    public void onEvent(@Nullable QuerySnapshot querySnapshots, @Nullable FirebaseFirestoreException error) {
        if (error != null) {
            Log.e("Firestore", error.toString());
            return;
        }
        if (querySnapshots != null) {
            cityDataList.clear();
            for (QueryDocumentSnapshot doc: querySnapshots) {
                String city = doc.getId();
                String province = doc.getString("Province");
                Log.d("Firestore", String.format("City(%s, %s) fetched", city, province));
                cityDataList.add(new City(city, province));
            }
            cityArrayAdapter.notifyDataSetChanged();
        }
    }
});

```

2. Now you can try to add some cities and verify that they appear in your app and the database.

References

- [Get started with Cloud Firestore \(https://firebase.google.com/docs/firestore/quickstart\)](https://firebase.google.com/docs/firestore/quickstart)