LabW11 – User Management

Objectives:

1. Understand how to use Firebase for user management.
2. Understand login using a social account.

Tasks:

1. Setup Facebook Login for the Firebase app
2. Add Facebook authentication to the Firebase app.

One benefit of storing data in the cloud is that every user of the app can see the same content. We have learned how to enable basic user authentication in Firebase app using “Email/Password” (refer Lab Week 4). In addition, we have also learned how to set basic Security Rules on our data to restrict access only to users who are signed in, so that we can prevent unauthenticated users from reading or writing to the Cloud Firestore.

Supporting user login and imposing data access permission is important so different users can only view and modify their own data. As login using social account is very common nowadays, in this tutorial we will learn how to enable authentication in Firebase app using Facebook account. Login with other types of social accounts such as Google and Twitter utilise a similar approach like this tutorial.

The following are required in order to complete this tutorial:

- A valid Facebook account. To create a new Facebook account, go to https://www.facebook.com/ and sign up for one.
- A Firebase project. We will use a functional Friendly Eats app previously built with Cloud Firestore in Lab Week 4.

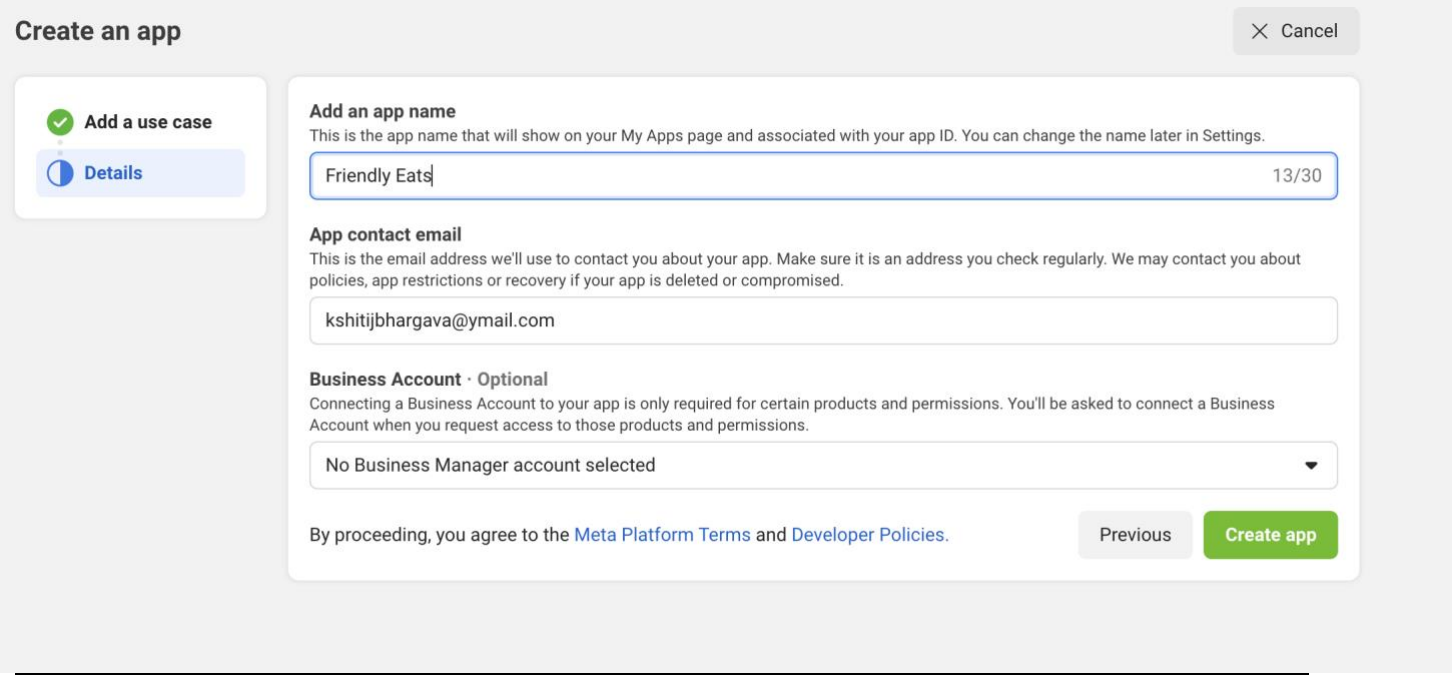
Task 1: Setup Facebook Login for Firebase app

Authentication is critical to applications success. Look at your app, and usually the very first thing that your users will do is to sign up or log in to their account. This will probably leverage some form of social or other types of authentication. Each social authentication provider is a little bit different, and some, such as Facebook, offer a native SDK to simplify the login process and offer additional functionalities specific to their service.

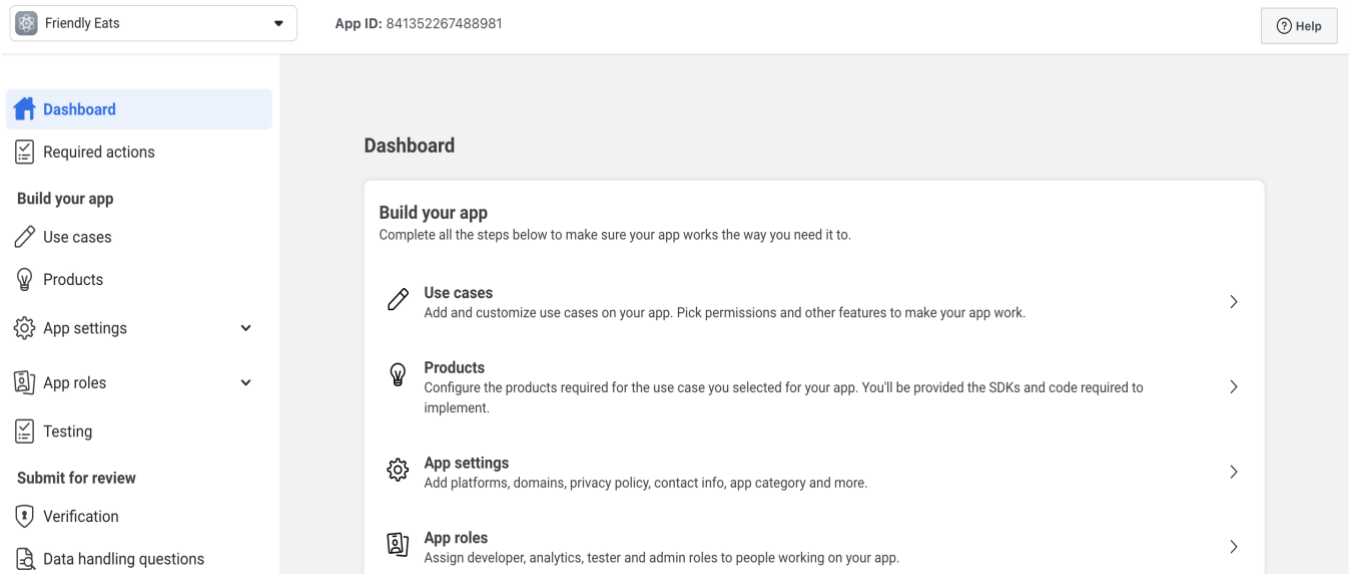
1. If you have not logged into your Facebook account on your browser, you will be asked to logging to FB account first and then redirected to the console to register as a developer. Follow the prompts given in the registration. Select “Developer” as the option.

2. To integrate Facebook authentication into our app, we must register a new application on the Facebook for Developers site: https://developers.facebook.com/. Log in using your Facebook account.

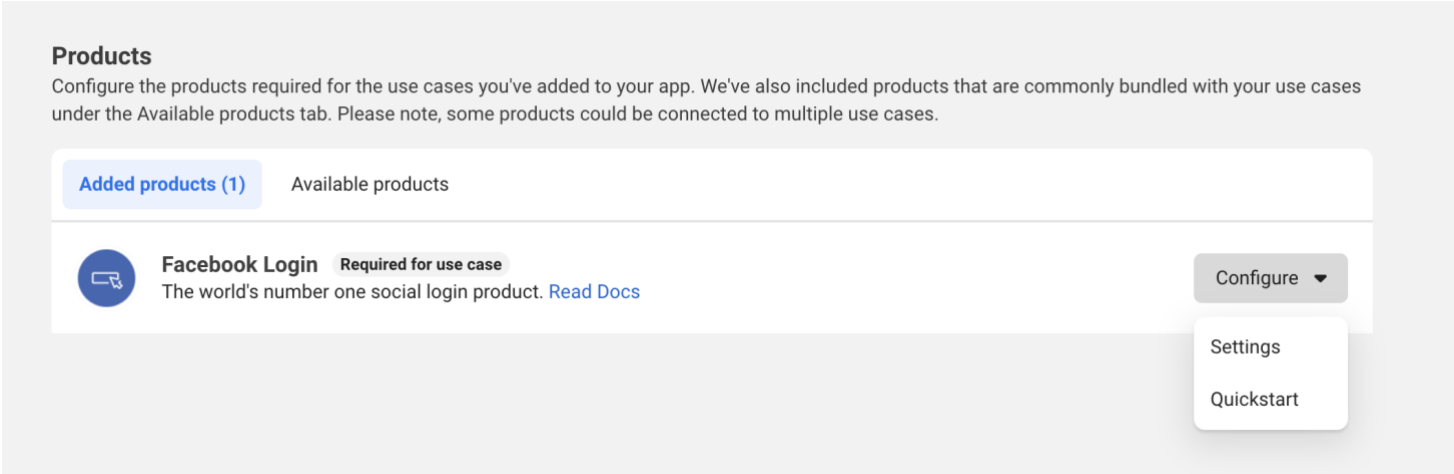
3. Click “My Apps” on the top right hand corner -> “Create App”.
4. You will be asked to select **What do you want your app to do?** Select “Allow people to log in with their Facebook account”, then click **Next.**
5. On the “**Create an App**” screen, enter “**Friendly Eats**” as the App Name, and your email address as the App Contact Email. Click “**Create App**” to create the app.



6. Re-enter your Facebook password to complete a security check and click Submit.
7. Next, on the dashboard screen select Products option.

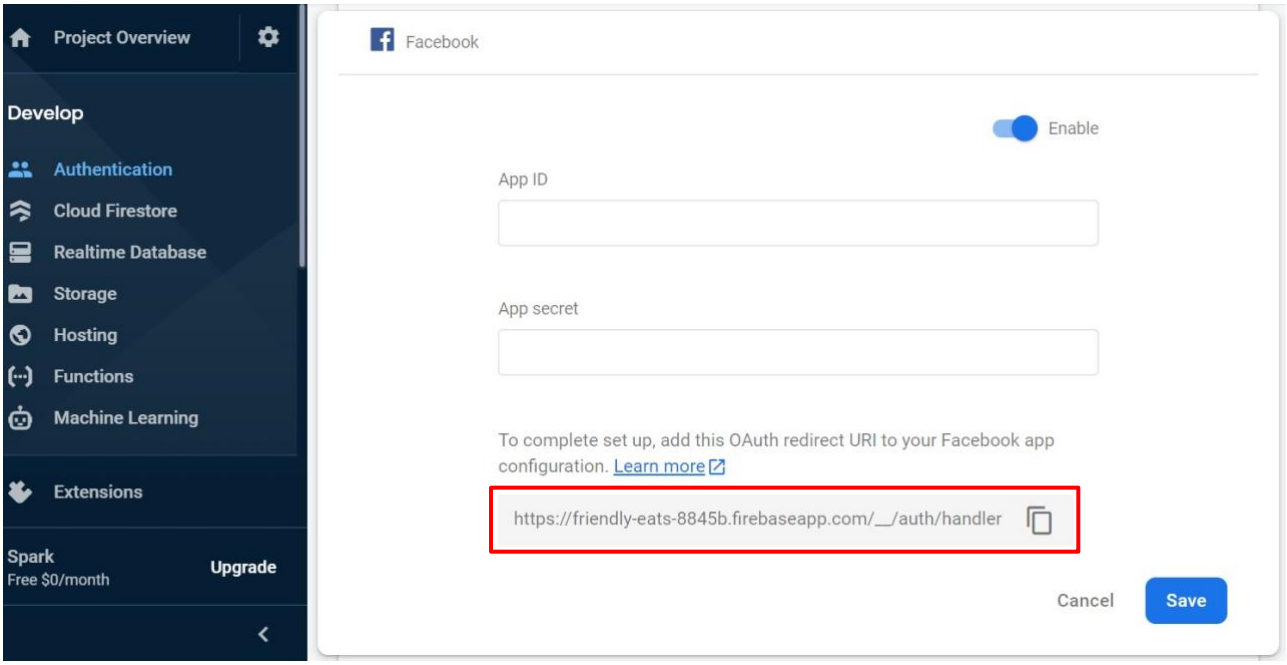


8. On the next screen click the **Configure** dropdown and select **Quickstart.** On the next screen select **Android.**

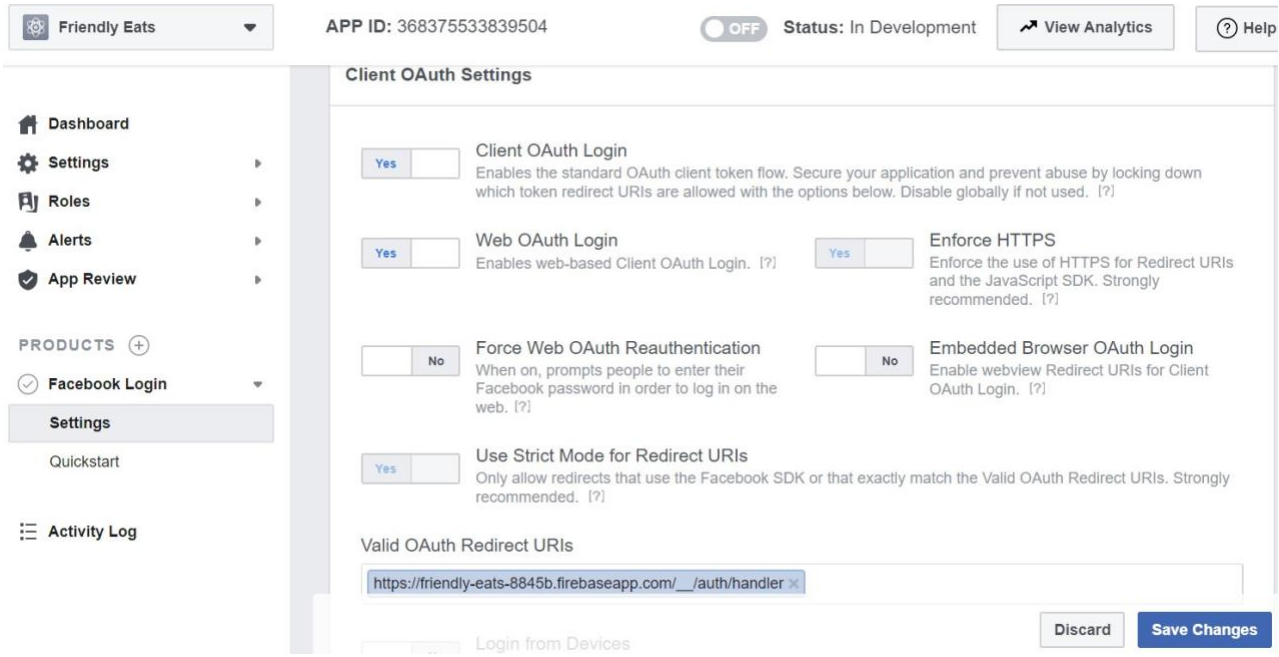


9. Click Settings to add your Valid OAuth Redirect URI, which can be obtained from your Firebase project.
10. Go to your Firebase console for “Friendly Eats” app -> Authentication -> “Sign-in method” -> Add new enable Facebook authentication.
11. Copy the OAuth redirect URI to add to your Facebook app configuration.

    e.g. https://friendly-eats-8845b.firebaseapp.com/ /auth/handler.

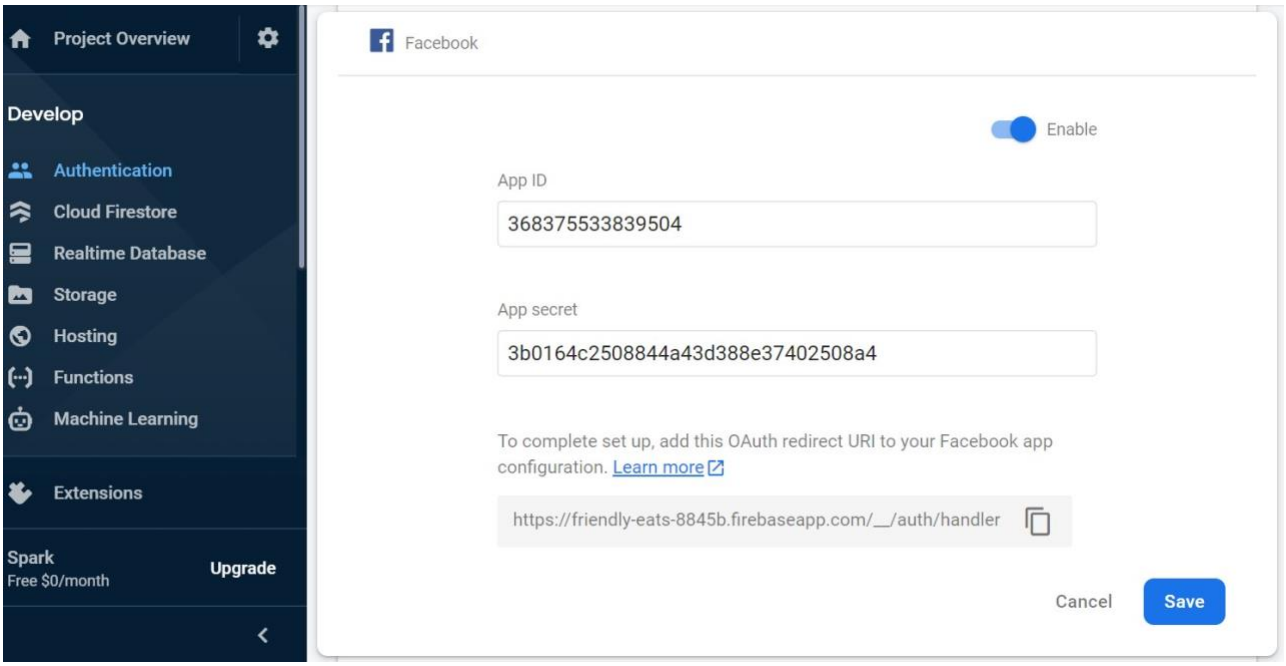


12. Add the URI to your Facebook app's "Valid OAuth Redirect URIs" /configuration. Make sure that you are using the HTTPS scheme. Click "Save Changes" to save your changes.



13. To enable Facebook authentication in Firebase, we need to obtain the App ID and App Secret. Click Settings -> Basic -> get the App ID and App Secret.

14. Enter the App ID and App Secret from step 8 into your Facebook authentication configuration for your Firebase project. Click Save.



15. Facebook authentication is now configured for your app.

Task 2: Add Facebook authentication to Firebase app

In this task, you will add authentication to the Friendly Eats Android app by using the Facebook login setup in the previous task.

Using FirebaseUI Auth is the recommended way to add a complete sign-in system to your app. It provides a drop-in auth solution that handles the UI flows for signing in users with email addresses and passwords, phone numbers, and with popular federated identity providers, including Facebook Login and Google Sign-In.

1. In your project-level build.gradle file, make sure to include Google's Maven repository in both your buildscript and allprojects sections.

```
buildscript {
    repositories {
        // Add this line
        google()
    }
    ...
}

allprojects {
    repositories {
        // Add this line
        google()
    }
    ...
}
```

2. Add the dependencies for FirebaseUI to your app-level build.gradle file. To support sign-in with Facebook, also include the Facebook SDK. The FirebaseUI Auth SDK has transitive dependencies on the Firebase SDK and the Google Play services SDK.

```
dependencies {
    ...

    // FirebaseUI (for authentication)
    implementation 'com.firebaseui:firebase-ui-auth:7.1.1'

    // Required only if Facebook login support is required
    // Find the latest Facebook SDK releases here: https://goo.gl/Ce5L94
    implementation 'com.facebook.android:facebook-android-sdk:8.2.0'

    ...
}
```

3. Add string resources to strings.xml that specify the identifying information required by Facebook.

```
<resources>
    ...

    <!-- Facebook application ID and custom URL scheme (app ID prefixed by 'fb'). -->
    <string name="facebook_application_id" translatable="false">YOUR_APP_ID</string>
    <string name="facebook_login_protocol_scheme"
translatable="false">fbYOUR_APP_ID</string>

    ...
</resources>
```

4. To kick off the FirebaseUI sign in flow, create a sign in intent with your preferred sign-in methods. Because our Friendly Eats app has previously

supported login using “Email/Password”, we can simply add Facebook authentication provider as another sign-in method to startSignIn() method.

```
private void startSignIn() {
    // Sign in with FirebaseUI
    Intent intent = AuthUI.getInstance().createSignInIntentBuilder()
            .setAvailableProviders(Arrays.asList(
                    new AuthUI.IdpConfig.EmailBuilder().build(),
                    new AuthUI.IdpConfig.FacebookBuilder().build()))
            .setIsSmartLockEnabled(false)
            .build();

    startActivityForResult(intent, RC_SIGN_IN);
    mViewModel.setIsSigningIn(true);
}
```

5. When the sign-in flow is complete, you will receive the result in onActivityResult:

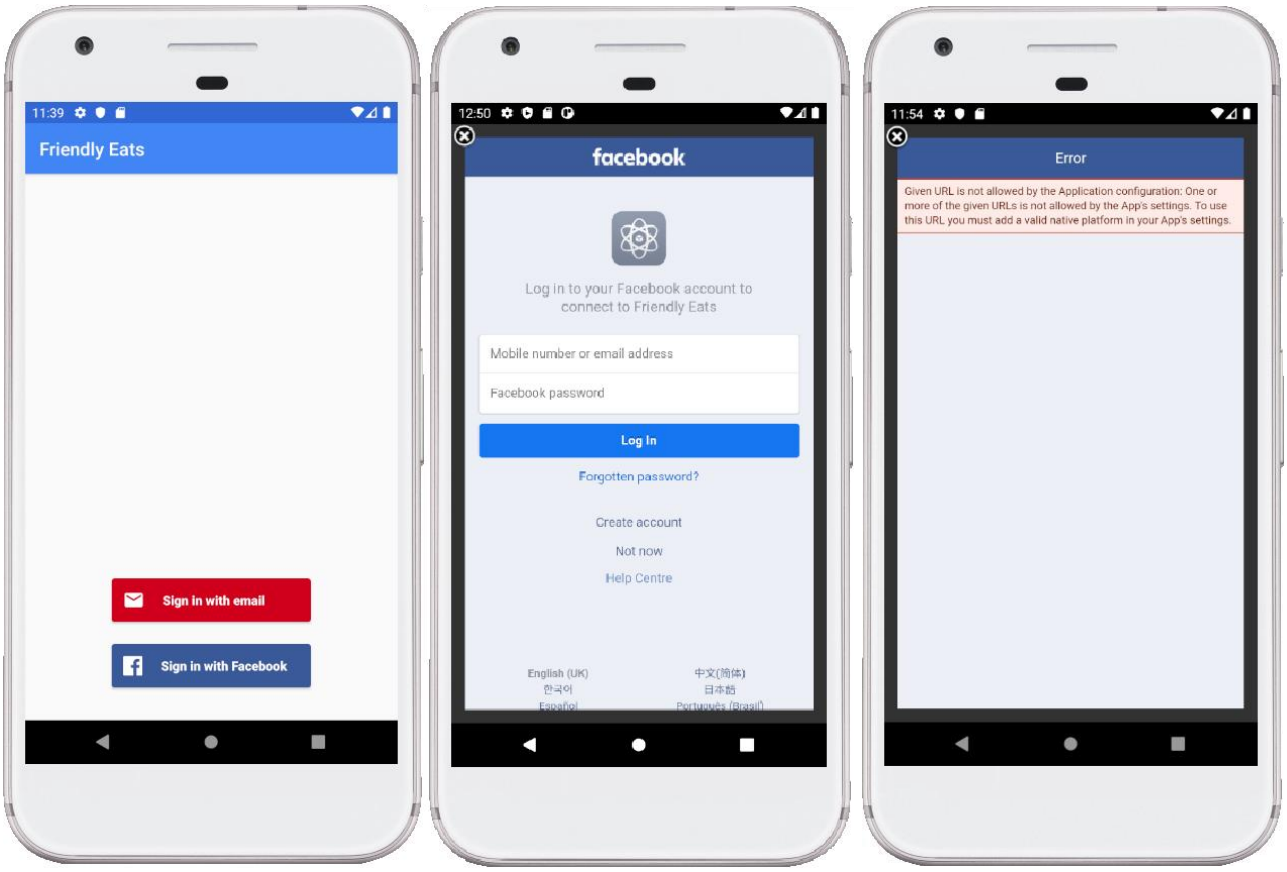
```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == RC_SIGN_IN) {
        mViewModel.setIsSigningIn(false);

        if (resultCode != RESULT_OK && shouldStartSignIn()) {
            startSignIn();
        }
    }
}
```

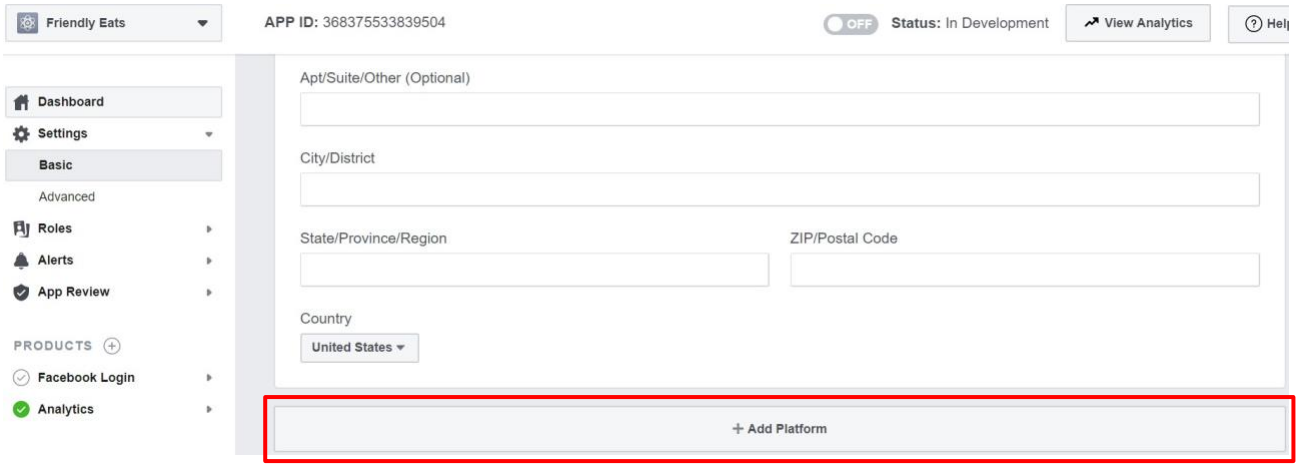
6. FirebaseUI provides convenience method signOut() to sign out of Firebase Authentication as well as social identity provider like Facebook:

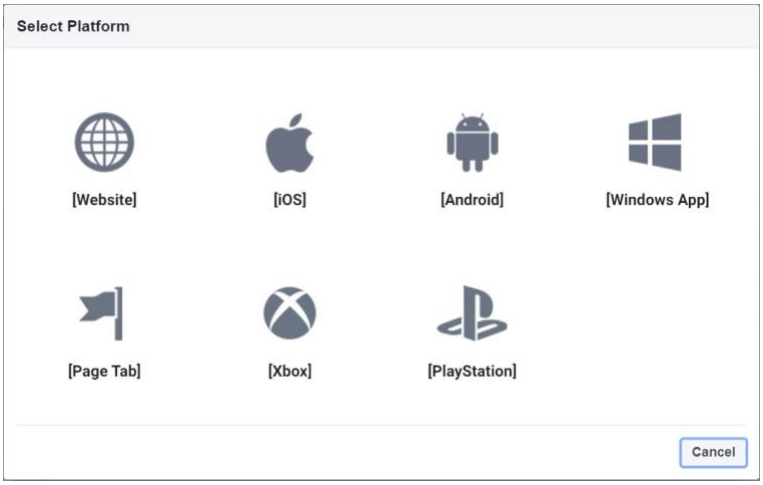
```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.menu_add_items:
            onAddItemsClicked();
            break;
        case R.id.menu_sign_out:
            AuthUI.getInstance().signOut(this);
            startSignIn();
            break;
    }
    return super.onOptionsItemSelected(item);
}
```

7. Build and run your app. You should get another option to “Sign in with Facebook” in addition to “Sign in with email”. Clicking “Sign in with Facebook” button will prompt you to sign in to the app using Facebook account. Use your Facebook account to login and you should notice an error message “… URL is not allowed …” – refer to screenshots below.



8. The error is because we have not completely integrated our Firebase app with Facebook. Open Facebook for Developers site and add an Android platform. Click Settings -> Basic -> +Add Platform -> Android.





9. Enter your app package name and default activity class name:
   - Google Play Package Name: com.google.firebase.example.fireeats
   - Class Name: com.google.firebase.example.fireeats.MainActivity

10. Generate a development key hash to ensure the authenticity of the interactions between your app and Facebook. You'll have a unique development key hash for each Android development environment. To generate a development key hash on Windows, run this command:

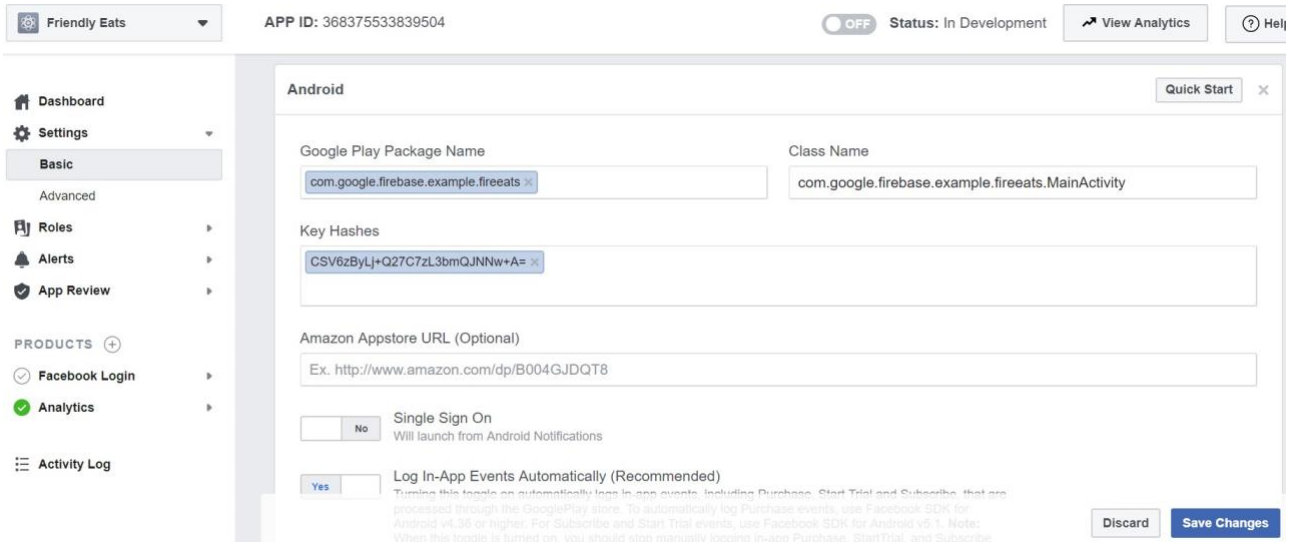
```
"C:\Program Files\Android\Android Studio\jre\bin\keytool" -exportcert -alias
androiddebugkey -keystore %HOMEPATH%\.android\debug.keystore | "C:\openssl-
0.9.8k_X64\bin\openssl" sha1 -binary | "C:\openssl-0.9.8k_X64\bin\openssl" base64
```

Make sure you have downloaded OpenSSL for Windows from this link:
https://code.google.com/archive/p/openssl-for-windows/downloads

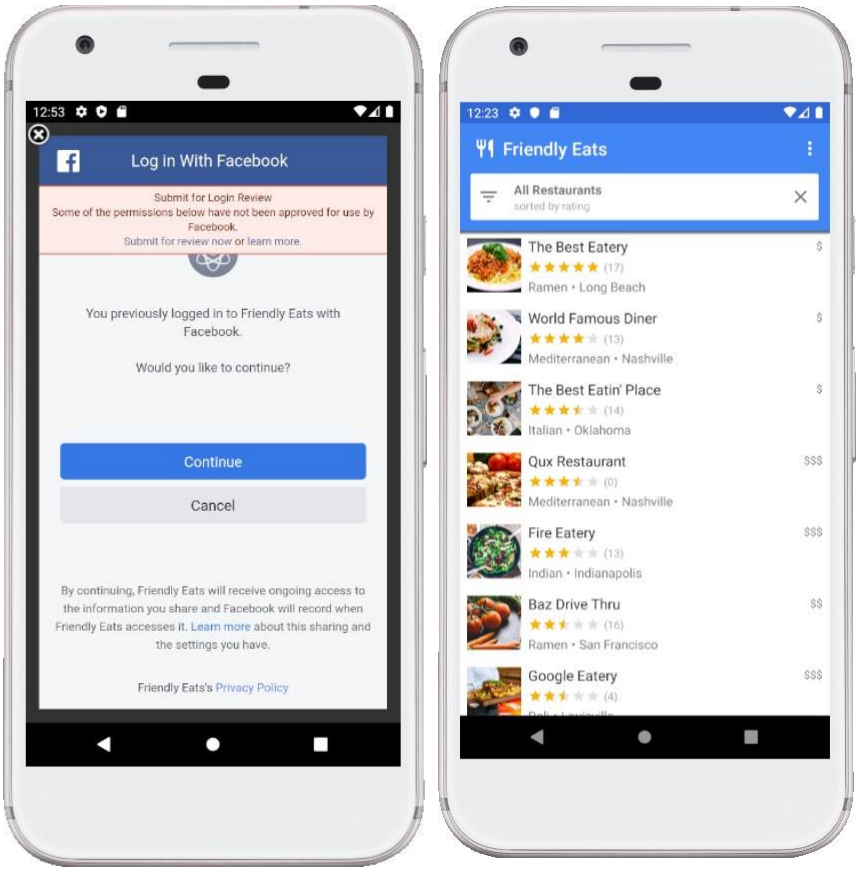
For further information on generating Key Hashes follow this link.

https://developers.facebook.com/docs/android/getting-started/

11. Add the hash to the “Key Hashes”. Save Changes.



12. Rebuild and run your app. This time you should be able to successfully sign in to your Firebase app using your Facebook account.



13. [Optional] By default FirebaseUI uses AppCompat for theming, which means it will naturally adopt the color scheme of your app. Customize the theme and logo of Facebook Sign In by passing a theme and a logo to the sign-in Intent builder.

References:

- Firebase Authentication - https://firebase.google.com/docs/auth/
- Easily add sign-in to your Android app with FirebaseUI - https://firebase.google.com/docs/auth/android/firebaseui
- Authenticate Using Facebook Login on Android - https://firebase.google.com/docs/auth/android/facebook-login
- OpenSSL for Windows - https://code.google.com/archive/p/openssl-for-windows/downloads

MainActivity.java

```
/**
 * Copyright 2017 Google Inc. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
package com.google.firebase.example.fireeats;

import android.content.Intent;
import android.os.Bundle;
import android.text.Html;
import  android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import  androidx.appcompat.widget.Toolbar;
import androidx.lifecycle.ViewModelProviders;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.firebase.ui.auth.AuthUI;
import com.google.android.material.snackbar.Snackbar;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.example.fireeats.adapter.RestaurantAdapter;
import com.google.firebase.example.fireeats.model.Restaurant;
import com.google.firebase.example.fireeats.util.RestaurantUtil;
import com.google.firebase.example.fireeats.viewmodel.MainActivityViewModel;
import com.google.firebase.firestore.CollectionReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.FirebaseFirestoreException;
import com.google.firebase.firestore.Query;

import java.util.Arrays;
import java.util.Collections;

public class MainActivity extends AppCompatActivity implements
        View.OnClickListener,
        FilterDialogFragment.FilterListener,
        RestaurantAdapter.OnRestaurantSelectedListener {

    private static final String TAG = "MainActivity";

    private static final int RC_SIGN_IN = 9001;

    private static final int LIMIT = 50;
```

```
    private Toolbar mToolbar;
    private TextView mCurrentSearchView;
    private TextView mCurrentSortByView;
    private RecyclerView mRestaurantsRecycler;
    private ViewGroup mEmptyView;

    private FirebaseFirestore mFirestore;
    private Query mQuery;

    private FilterDialogFragment mFilterDialog;
    private RestaurantAdapter mAdapter;

    private MainActivityViewModel mViewModel;

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

        mToolbar = findViewById(R.id.toolbar);
        setSupportActionBar(mToolbar);

        mCurrentSearchView = findViewById(R.id.text_current_search);
        mCurrentSortByView = findViewById(R.id.text_current_sort_by);
        mRestaurantsRecycler = findViewById(R.id.recycler_restaurants);
        mEmptyView = findViewById(R.id.view_empty);

        findViewById(R.id.filter_bar).setOnClickListener(this);
        findViewById(R.id.button_clear_filter).setOnClickListener(this);

        // View model
        mViewModel = ViewModelProviders.of(this).get(MainActivityViewModel.class);

        // Enable Firestore logging
        FirebaseFirestore.setLoggingEnabled(true);

        // Initialize Firestore and the main RecyclerView
        initFirestore();
        initRecyclerView();

        // Filter Dialog
        mFilterDialog = new FilterDialogFragment();
    }

    private void initFirestore() {
        // TODO(developer): Implement
        mFirestore = FirebaseFirestore.getInstance();

        mQuery = mFirestore.collection("restaurants")
                .orderBy("avgRating", Query.Direction.DESCENDING)
                .limit(LIMIT);

    }

    private void initRecyclerView() {
        if (mQuery == null) {
            Log.w(TAG, "No query, not initializing RecyclerView");
        }

        mAdapter = new RestaurantAdapter(mQuery, this) {

            @Override
            protected void onDataChanged() {
                // Show/hide content if the query returns empty.
```

```
                if (getItemCount() == 0) {
                    mRestaurantsRecycler.setVisibility(View.GONE);
                    mEmptyView.setVisibility(View.VISIBLE);
                } else {
                    mRestaurantsRecycler.setVisibility(View.VISIBLE);
                    mEmptyView.setVisibility(View.GONE);
                }
            }

            @Override
            protected void onError(FirebaseFirestoreException e) {
                // Show a snackbar on errors
                Snackbar.make(findViewById(android.R.id.content),
                        "Error: check logs for info.", Snackbar.LENGTH_LONG).show();
            }
        };

        mRestaurantsRecycler.setLayoutManager(new LinearLayoutManager(this));
        mRestaurantsRecycler.setAdapter(mAdapter);
    }

    @Override
    public void onStart() {
        super.onStart();

        // Start sign in if necessary
        if (shouldStartSignIn()) {
            startSignIn();
            return;
        }

        // Apply filters
        onFilter(mViewModel.getFilters());

        // Start listening for Firestore updates
        if (mAdapter != null) {
            mAdapter.startListening();
        }
    }

    @Override
    public void onStop() {
        super.onStop();
        if (mAdapter != null) {
            mAdapter.stopListening();
        }
    }

    private void onAddItemsClicked() {
        // TODO(developer): Add random restaurants
        // Get a reference to the restaurants collection

        CollectionReference restaurants = mFirestore.collection("restaurants");
        for (int i = 0; i < 10; i++) {
            // Get a random Restaurant POJO
            Restaurant restaurant = RestaurantUtil.getRandom(this);
            // Add a new document to the restaurants collection
            restaurants.add(restaurant);
        }

        showTodoToast();
    }

    @Override
    public void onFilter(Filters filters) {
```

```
        // TODO(developer): Construct new query
        // Construct query basic query

        showTodoToast();

    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return super.onCreateOptionsMenu(menu);
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.menu_add_items:
                onAddItemsClicked();
                break;
            case R.id.menu_sign_out:
                AuthUI.getInstance().signOut(this);
                startSignIn();
                break;
        }
        return super.onOptionsItemSelected(item);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == RC_SIGN_IN) {
            mViewModel.setIsSigningIn(false);

            if (resultCode != RESULT_OK && shouldStartSignIn()) {
                startSignIn();
            }
        }
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.filter_bar:
                onFilterClicked();
                break;
            case R.id.button_clear_filter:
                onClearFilterClicked();
        }
    }

    public void onFilterClicked() {
        // Show the dialog containing filter options
        mFilterDialog.show(getSupportFragmentManager(), FilterDialogFragment.TAG);
    }

    public void onClearFilterClicked() {
        mFilterDialog.resetFilters();

        onFilter(Filters.getDefault());
    }

    @Override
    public void onRestaurantSelected(DocumentSnapshot restaurant) {
        // Go to the details page for the selected restaurant
        Intent intent = new Intent(this, RestaurantDetailActivity.class);
```

```
        intent.putExtra(RestaurantDetailActivity.KEY_RESTAURANT_ID,  restaurant.getId());

        startActivity(intent);
    }

    private boolean shouldStartSignIn() {
        return (!mViewModel.getIsSigningIn() &&
FirebaseAuth.getInstance().getCurrentUser() == null);
    }

    private void startSignIn() {
        // Sign in with FirebaseUI
        /*Intent intent = AuthUI.getInstance().createSignInIntentBuilder()
                .setAvailableProviders(Collections.singletonList(
                        new AuthUI.IdpConfig.EmailBuilder().build()))
                .setIsSmartLockEnabled(false)
                .build();

        startActivityForResult(intent, RC_SIGN_IN);
        mViewModel.setIsSigningIn(true);*/

        // Sign in with FirebaseUI
        Intent intent = AuthUI.getInstance().createSignInIntentBuilder()
                .setAvailableProviders(Arrays.asList(
                        new AuthUI.IdpConfig.EmailBuilder().build(),
                        new AuthUI.IdpConfig.FacebookBuilder().build()))
                .setIsSmartLockEnabled(false)
                .build();

        startActivityForResult(intent, RC_SIGN_IN);
        mViewModel.setIsSigningIn(true);


    }

    private void showTodoToast() {
        Toast.makeText(this, "TODO: Implement", Toast.LENGTH_SHORT).show();
    }
}
```