

Lab 8: Tìm hiểu về FireBase



MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Tìm hiểu về FireBase của Google
- ✓ Tạo Project Trên FireBase
- ✓ Sử dụng được FireBase trong android

BÀI 1 (2 ĐIỂM)

Viết ứng dụng Android đơn giản sử dụng firebase authentication và profile management. Ta sẽ xây dựng chức năng Login, Registration, Forgot password, Change Email, Change Password & cuối cùng là Sign Out option.

HƯỚNG DẪN:

- ✓ Đầu tiên sinh viên cần đi đến liên kết sau <https://firebase.google.com/> tạo tài khoản để truy cập vào console firebase. Sau khi truy cập console tiến hành tạo project mới.

Learn more'. At the bottom right are two buttons: 'CANCEL' and 'CREATE PROJECT'." data-bbox="279 583 775 905"/>

Create a project ✕

Project name

andoid-poly

Country/region ?

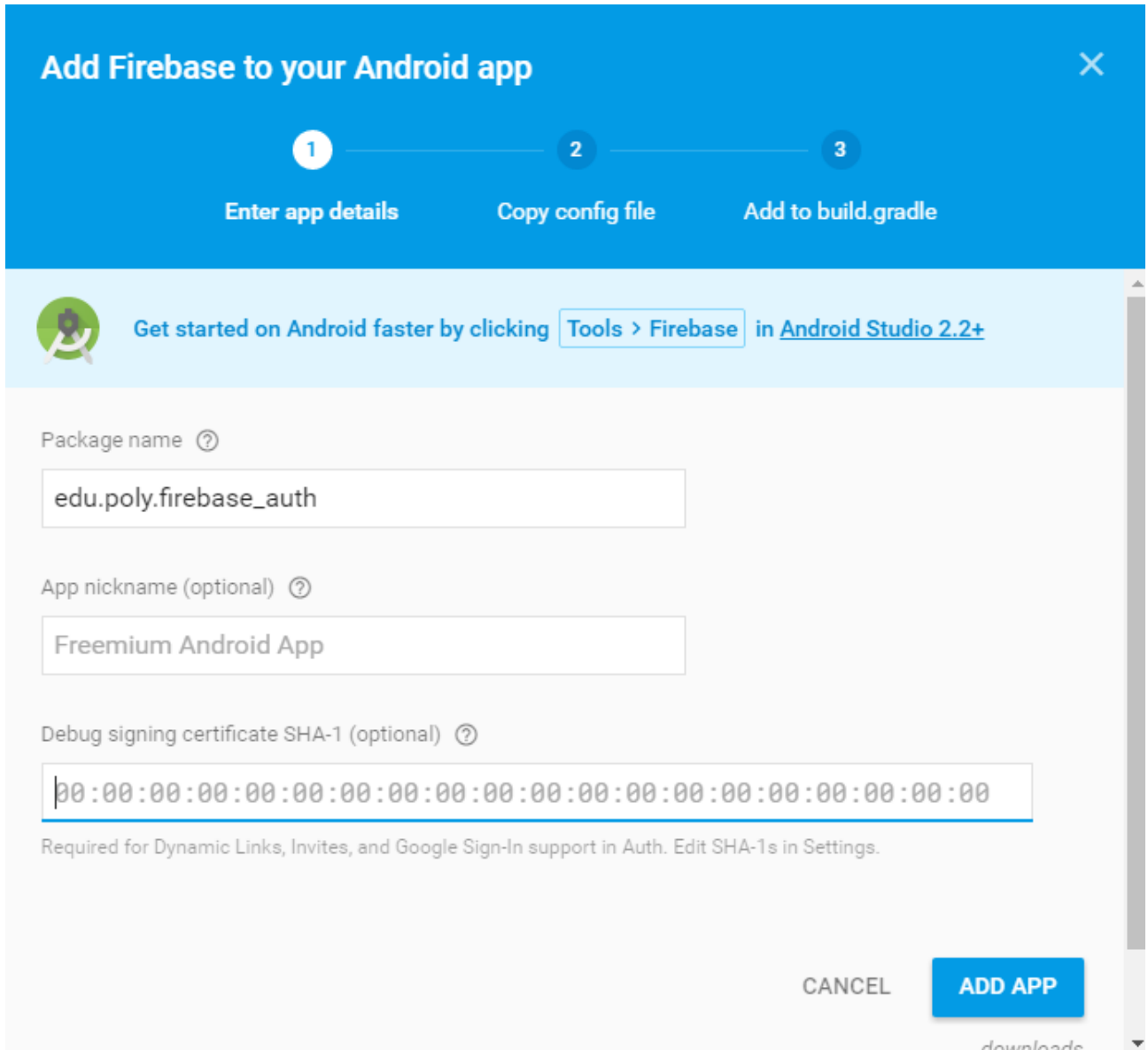
Vietnam

By default, your Firebase Analytics data will enhance other Firebase features and Google products. You can control how your Firebase Analytics data is shared in your settings at anytime. [Learn more](#)

CANCEL CREATE PROJECT

FPT POLYTECHNIC

- ✓ Nhập tên package ứng dụng android của sinh viên (mine is edu.poly.firebase_auth) ứng dụng mà ta sẽ tương tác với Firebase. Ở bước tiếp theo file google-service.json sẽ được tải khi sinh viên nhấn nút ADD APP copy file đó vào thư mục project theo hướng dẫn.



Add Firebase to your Android app ✕

1 Enter app details 2 Copy config file 3 Add to build.gradle

Get started on Android faster by clicking [Tools > Firebase](#) in [Android Studio 2.2+](#)

Package name ⓘ

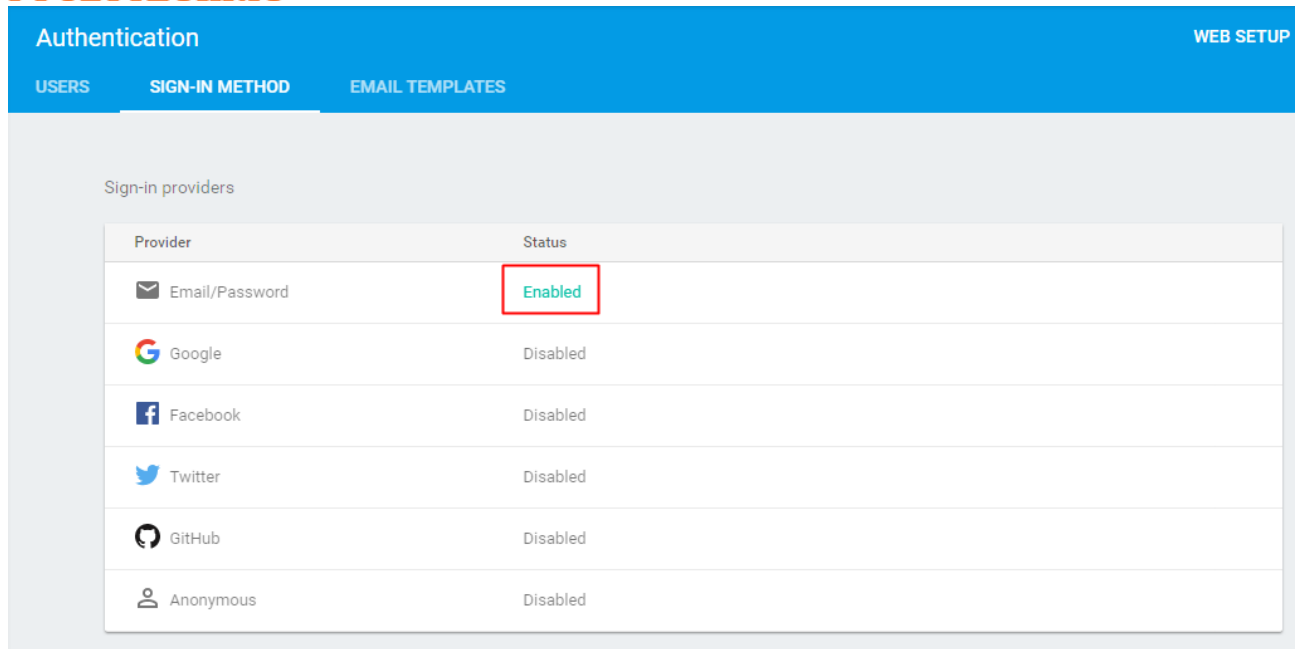
App nickname (optional) ⓘ

Debug signing certificate SHA-1 (optional) ⓘ

Required for Dynamic Links, Invites, and Google Sign-In support in Auth. Edit SHA-1s in Settings.

CANCEL **ADD APP**

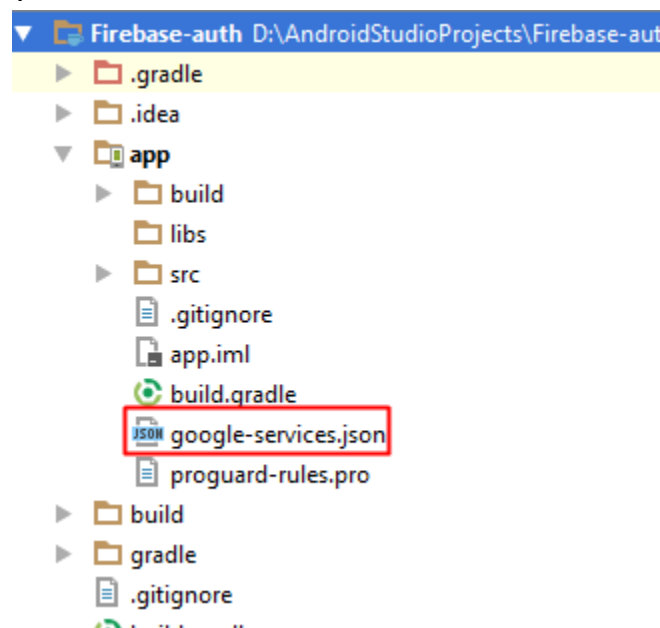
- ✓ Tiếp theo sinh viên tới project dashboard. Tìm Auth và nhấn started. Đi tới phần thiết lập sign in method và chọn Email & Password và enable nó lên.



- ✓ Tạo project mới trong Android Studio. Trong khi điền project details, sử dụng cùng tên package mà sinh viên đã nhập ở firebase console.
- ✓ Mở AndroidManifest.xml và thêm INTERNET permission

```
<uses-permission android:name="android.permission.INTERNET" />
```

- ✓ Copy file google-services.json mà sinh viên đã tải lúc tạo project vào thư mục app trong project. Bước này rất quan trọng. project sẽ không build nếu không có file này



FPT POLYTECHNIC

- ✓ Bây giờ mở build.gradle nằm ở mục project và thêm firebase dependency.

```
dependencies {
    classpath 'com.android.tools.build:gradle:2.2.3'
    classpath 'com.google.gms:google-services:3.0.0'
}
```

- ✓ Mở app/build.gradle và add firebase auth dependency. ở cuối file , add plugin: 'com.google.gms.google-services'

```
dependencies {
    compile "com.google.firebase:firebase-auth:9.0.2"
}
apply plugin: 'com.google.gms.google-services'
```

- ✓ Thêm các resources bên dưới trong file dims.xml, colors.xml và strings.xml.

dims.xml

```
<resources>
    <!-- Default screen margins, per the Android Design guidelines. -->
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
    <dimen name="fab_margin">16dp</dimen>
    <dimen name="logo_w_h">100dp</dimen>
</resources>
```

colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>

    <color name="bg_login">#26ae90</color>
    <color name="bg_register">#2e3237</color>
    <color name="bg_main">#428bca</color>
    <color name="white">#ffffff</color>
    <color name="input_login">#222222</color>
    <color name="input_login_hint">#999999</color>
    <color name="input_register">#888888</color>
    <color name="input_register_bg">#3b4148</color>
    <color name="input_register_hint">#5e6266</color>
    <color name="btn_login">#26ae90</color>
    <color name="btn_login_bg">#ecef1</color>
    <color name="lbl_name">#333333</color>
    <color name="btn_logout_bg">#ff6861</color>
</resources>
```

strings.xml

```
<resources>
    <string name="app_name">Firebase Auth</string>
```

```

<string name="action_sign_in_short">Register</string>
<string name="email">Email</string>
<string name="minimum_password">Password too short, enter minimum 6
characters!</string>
<string name="auth_failed">Authentication failed, check your email and
password or sign up</string>
<string name="change_email">Change Email</string>
<string name="change_password">Change Password</string>
<string name="send_password_reset_email">Send Password reset
email</string>
<string name="remove_user">Remove user</string>
<string name="new_pass">New Password</string>
<string name="title_activity_profile">Firebase</string>
<string name="title_activity_login">Sign in</string>
<string name="hint_email">Email</string>
<string name="hint_password">Password</string>
<string name="prompt_password">Password</string>

<string name="hint_name">Fullname</string>
<string name="btn_login">LOGIN</string>
<string name="btn_link_to_register">Not a member? Get registered in
Firebase now!</string>
<string name="btn_link_to_login">Already registered. Login Me!</string>
<string
name="title_activity_reset_password">ResetPasswordActivity</string>
<string name="btn_forgot_password">Forgot your password?</string>
<string name="btn_reset_password">Reset Password</string>
<string name="btn_back"><![CDATA[<< Back]]></string>
<string name="hint_new_email">New Email</string>
<string name="btn_change">Change</string>
<string name="btn_send">Send</string>
<string name="btn_remove">Remove</string>
<string name="btn_sign_out">Sign Out</string>
<string name="lbl_forgot_password">Forgot password?</string>
<string name="forgot_password_msg">We just need your registered Email Id
to sent you password reset instructions.</string>
</resources>

```

- ✓ Bây giờ ta sẽ tạo Sing up Screen chứa 2 EditText fields cho email và password. Và một số nữa để điều hướng tới login và forgot password screens
- ✓ Tạo activity mới đặt tên là SignupActivity.java và thêm đoạn code bên dưới cho file activity_signup.xml

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_gravity="center"
android:background="@color/colorPrimary"

```

```

android:fitsSystemWindows="true"
tools:context="edu.poly.firebase_auth.LoginActivity">

<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="30dp"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="@dimen/activity_horizontal_margin">

    <ImageView
        android:layout_width="@dimen/logo_w_h"
        android:layout_height="@dimen/logo_w_h"
        android:layout_gravity="center_horizontal"
        android:layout_marginBottom="10dp"
        android:src="@mipmap/ic_launcher" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:padding="10dp"
        android:text="@string/lbl_forgot_password"
        android:textColor="@android:color/white"
        android:textSize="20dp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="10dp"
        android:gravity="center_horizontal"
        android:padding="@dimen/activity_horizontal_margin"
        android:text="@string/forgot_password_msg"
        android:textColor="@android:color/white"
        android:textSize="14dp" />

    <android.support.design.widget.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <EditText
            android:id="@+id/email"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="10dp"
            android:layout_marginTop="20dp"
            android:hint="@string/hint_email"
            android:inputType="textEmailAddress"
            android:textColor="@android:color/white"
            android:textColorHint="@android:color/white" />
    </android.support.design.widget.TextInputLayout>

    <!-- Login Button -->

    <Button
        android:id="@+id/btn_reset_password"

```

```

        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dip"
        android:background="@color/colorAccent"
        android:text="@string/btn_reset_password"
        android:textColor="@android:color/black" />

        <Button
            android:id="@+id/btn_back"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:background="@null"
            android:text="@string/btn_back"
            android:textColor="@color/colorAccent" />

    </LinearLayout>

    <ProgressBar
        android:id="@+id/progressBar"
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:layout_gravity="center|bottom"
        android:layout_marginBottom="20dp"
        android:visibility="gone" />
</android.support.design.widget.CoordinatorLayout>

```

- ✓ Mở SignupActivity.java và thêm code bên dưới. Firebase cung cấp createUserWithEmailAndPassword() method để tạo một user mới với email và password data.

```

package edu.poly.firebase_auth;

import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;

public class SignupActivity extends AppCompatActivity {
    private EditText inputEmail, inputPassword;
    private Button btnSignIn, btnSignUp, btnResetPassword;
    private ProgressBar progressBar;
    private FirebaseAuth auth;

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_signup);
    //Get Firebase auth instance
    auth = FirebaseAuth.getInstance();

    btnSignIn = (Button) findViewById(R.id.sign_in_button);
    btnSignUp = (Button) findViewById(R.id.sign_up_button);
    inputEmail = (EditText) findViewById(R.id.email);
    inputPassword = (EditText) findViewById(R.id.password);
    progressBar = (ProgressBar) findViewById(R.id.progressBar);
    btnResetPassword = (Button) findViewById(R.id.btn_reset_password);

    btnResetPassword.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            startActivity(new Intent(SignupActivity.this,
ResetPasswordActivity.class));
        }
    });

    btnSignIn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            finish();
        }
    });

    btnSignUp.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            String email = inputEmail.getText().toString().trim();
            String password = inputPassword.getText().toString().trim();

            if (TextUtils.isEmpty(email)) {
                Toast.makeText(getApplicationContext(), "Enter email
address!", Toast.LENGTH_SHORT).show();
                return;
            }

            if (TextUtils.isEmpty(password)) {
                Toast.makeText(getApplicationContext(), "Enter
password!", Toast.LENGTH_SHORT).show();
                return;
            }

            if (password.length() < 6) {
                Toast.makeText(getApplicationContext(), "Password too
short, enter minimum 6 characters!", Toast.LENGTH_SHORT).show();
                return;
            }

            progressBar.setVisibility(View.VISIBLE);
            //create user
            auth.createUserWithEmailAndPassword(email, password)
                .addOnCompleteListener(SignupActivity.this, new

```


FPT POLYTECHNIC

```

OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull Task<AuthResult>
task) {
        Toast.makeText(SignupActivity.this,
"createUserWithEmail:onComplete:" + task.isSuccessful(),
Toast.LENGTH_SHORT).show();

        progressBar.setVisibility(View.GONE);
        // If sign in fails, display a message to
the user. If sign in succeeds
        // the auth state listener will be notified
and logic to handle the
        // signed in user can be handled in the
listener.

        if (!task.isSuccessful()) {
            Toast.makeText(SignupActivity.this,
"Authentication failed." + task.getException(),
            Toast.LENGTH_SHORT).show();
        } else {
            startActivity(new
Intent(SignupActivity.this, MainActivity.class));
            finish();
        }
    }
});

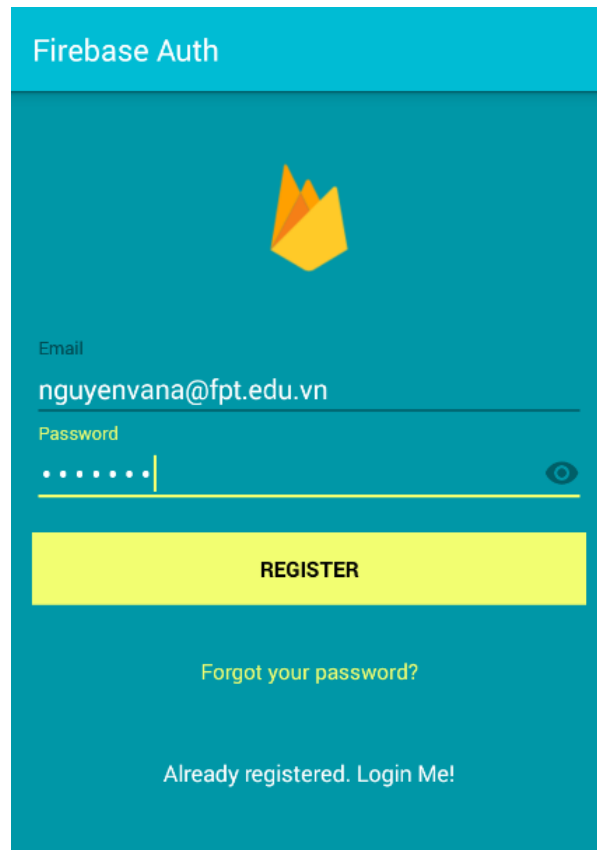
}

@Override
protected void onResume() {
    super.onResume();
    progressBar.setVisibility(View.GONE);
}

}

```

✓ Tiến hành test chức năng đăng ký



Registration screen for Firebase Auth. The screen has a teal background with the Firebase logo at the top. It contains input fields for Email (nguyenvana@fpt.edu.vn) and Password (masked with dots). Below the fields is a yellow 'REGISTER' button. At the bottom, there are links for 'Forgot your password?' and 'Already registered. Login Me!'.

- ✓ Vào firebase và kiểm tra tài khoản đã được thêm

Search by email address or user UID					ADD USER	↻
Email	Providers	Created	Signed In	User UID ↑		
nguyenvana@fpt.edu.vn	✉	Jan 2, 2017	Jan 2, 2017	1Tuk18Z0PHXI0i6WTD4fDUe...		
Rows per page: 50					1-1 of 1	< >

- ✓ Bây giờ ta tạo tiếp Login screen và kiểm tra credentials mà ta đã tạo trong sign up screen
- ✓ Tạo activity khác đặt tên là LoginActivity.java và thêm code bên dưới cho file activity_login.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<android.support.design.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:fitsSystemWindows="true"
tools:context="edu.poly.firebase_auth.LoginActivity">
```

```

<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@color/colorPrimary"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="@dimen/activity_horizontal_margin">

    <ImageView
        android:layout_width="@dimen/logo_w_h"
        android:layout_height="@dimen/logo_w_h"
        android:layout_gravity="center_horizontal"
        android:layout_marginBottom="30dp"
        android:src="@drawable/ic_launcher" />

    <android.support.design.widget.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <EditText
            android:id="@+id/email"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="10dp"
            android:hint="@string/hint_email"
            android:inputType="textEmailAddress"
            android:textColor="@android:color/white"
            android:textColorHint="@android:color/white" />
    </android.support.design.widget.TextInputLayout>

    <android.support.design.widget.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <EditText
            android:id="@+id/password"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="10dp"
            android:hint="@string/hint_password"
            android:inputType="textPassword"
            android:textColor="@android:color/white"
            android:textColorHint="@android:color/white" />
    </android.support.design.widget.TextInputLayout>

    <!-- Login Button -->

    <Button
        android:id="@+id/btn_login"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:background="@color/colorAccent"
        android:text="@string/btn_login"
        android:textColor="@android:color/black" />

    <Button

```

```

        android:id="@+id/btn_reset_password"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dip"
        android:background="@null"
        android:text="@string/btn_forgot_password"
        android:textAllCaps="false"
        android:textColor="@color/colorAccent" />

<!-- Link to Login Screen -->

<Button
    android:id="@+id/btn_signup"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dip"
    android:background="@null"
    android:text="@string/btn_link_to_register"
    android:textAllCaps="false"
    android:textColor="@color/white"
    android:textSize="15dp" />
</LinearLayout>

<ProgressBar
    android:id="@+id/progressBar"
    android:layout_width="30dp"
    android:layout_height="30dp"
    android:layout_gravity="center|bottom"
    android:layout_marginBottom="20dp"
    android:visibility="gone" />
</android.support.design.widget.CoordinatorLayout>

```

- ✓ Mở LoginActivity.java và code nội dung như sau. Firebase cung cấp `signInWithEmailAndPassword()` method để user đăng nhập

```

package edu.poly.firebase_auth;

import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.Toolbar;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;

public class LoginActivity extends AppCompatActivity {

```

FPT POLYTECHNIC

```

private EditText inputEmail, inputPassword;
private FirebaseAuth auth;
private ProgressBar progressBar;
private Button btnSignup, btnLogin, btnReset;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);
    //Get Firebase auth instance
    auth = FirebaseAuth.getInstance();

    if (auth.getCurrentUser() != null) {
        startActivity(new Intent(LoginActivity.this,
MainActivity.class));
        finish();
    }

    // set the view now
    setContentView(R.layout.activity_login);

    inputEmail = (EditText) findViewById(R.id.email);
    inputPassword = (EditText) findViewById(R.id.password);
    progressBar = (ProgressBar) findViewById(R.id.progressBar);
    btnSignup = (Button) findViewById(R.id.btn_signup);
    btnLogin = (Button) findViewById(R.id.btn_login);
    btnReset = (Button) findViewById(R.id.btn_reset_password);

    //Get Firebase auth instance
    auth = FirebaseAuth.getInstance();

    btnSignup.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            startActivity(new Intent(LoginActivity.this,
SignupActivity.class));
        }
    });

    btnReset.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            startActivity(new Intent(LoginActivity.this,
ResetPasswordActivity.class));
        }
    });

    btnLogin.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String email = inputEmail.getText().toString();
            final String password = inputPassword.getText().toString();

            if (TextUtils.isEmpty(email)) {
                Toast.makeText(getApplicationContext(), "Enter email
address!", Toast.LENGTH_SHORT).show();

```

```

        return;
    }

    if (TextUtils.isEmpty(password)) {
        Toast.makeText(getApplicationContext(), "Enter
password!", Toast.LENGTH_SHORT).show();
        return;
    }

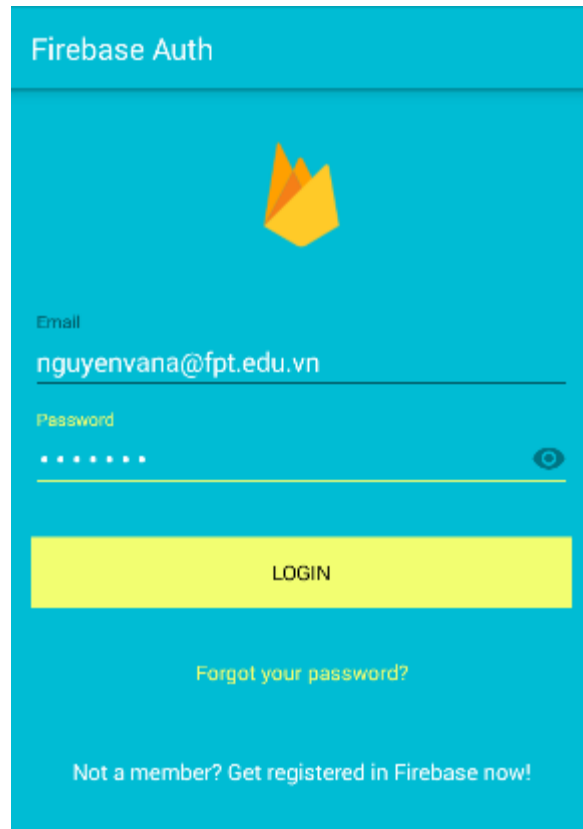
    progressBar.setVisibility(View.VISIBLE);

    //authenticate user
    auth.signInWithEmailAndPassword(email, password)
        .addOnCompleteListener(LoginActivity.this, new
OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult>
task) {
            // If sign in fails, display a message to
            the user. If sign in succeeds
            // the auth state listener will be notified
            and logic to handle the
            // signed in user can be handled in the
            listener.

            progressBar.setVisibility(View.GONE);
            if (!task.isSuccessful()) {
                // there was an error
                if (password.length() < 6) {
                    inputPassword.setError(getString(R.string.minimum_password));
                } else {
                    Toast.makeText(LoginActivity.this,
getString(R.string.auth_failed), Toast.LENGTH_LONG).show();
                }
            } else {
                Intent intent = new
Intent(LoginActivity.this, MainActivity.class);
                startActivity(intent);
                finish();
            }
        }
    });
}
}
}
}

```

✓ Tiến hành test chức năng đăng nhập



- ✓ Một tính năng tuyệt vời khác và firebase cung cấp là , gửi reset password email khi yêu cầu. Công việc rất là khó nếu bạn muốn có email server riêng
- ✓ Tạo activity khác đặt tên là ResetPasswordActivity.java và thêm đoạn code bên dưới cho file activity_reset_password.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_gravity="center"
    android:background="@color/colorPrimary"
    android:fitsSystemWindows="true"
    tools:context="edu.poly.firebase_auth.ResetPasswordActivity">

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="30dp"
        android:gravity="center"
        android:orientation="vertical"
        android:padding="@dimen/activity_horizontal_margin">

        <ImageView
            android:layout_width="@dimen/logo_w_h"
```

```

        android:layout_height="@dimen/logo_w_h"
        android:layout_gravity="center_horizontal"
        android:layout_marginBottom="10dp"
        android:src="@mipmap/ic_launcher" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:padding="10dp"
    android:text="@string/lbl_forgot_password"
    android:textColor="@android:color/white"
    android:textSize="20dp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:gravity="center_horizontal"
    android:padding="@dimen/activity_horizontal_margin"
    android:text="@string/forgot_password_msg"
    android:textColor="@android:color/white"
    android:textSize="14dp" />

<android.support.design.widget.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <EditText
        android:id="@+id/email"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="10dp"
        android:layout_marginTop="20dp"
        android:hint="@string/hint_email"
        android:inputType="textEmailAddress"
        android:textColor="@android:color/white"
        android:textColorHint="@android:color/white" />
</android.support.design.widget.TextInputLayout>

<!-- Login Button -->

<Button
    android:id="@+id/btn_reset_password"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:background="@color/colorAccent"
    android:text="@string/btn_reset_password"
    android:textColor="@android:color/black" />

<Button
    android:id="@+id/btn_back"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:background="@null"
    android:text="@string/btn_back"

```



```

        android:textColor="@color/colorAccent" />

    </LinearLayout>

    <ProgressBar
        android:id="@+id/progressBar"
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:layout_gravity="center|bottom"
        android:layout_marginBottom="20dp"
        android:visibility="gone" />
</android.support.design.widget.CoordinatorLayout>

```

- ✓ Mở ResetPasswordActivity.java và thêm code sau. Sinh viên có thể sử dụng sendPasswordResetEmail() method để gửi password reset email.

```

package edu.poly.firebase_auth;

import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;

public class ResetPasswordActivity extends AppCompatActivity {
    private EditText inputEmail;
    private Button btnReset, btnBack;
    private FirebaseAuth auth;
    private ProgressBar progressBar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_reset_password);
        inputEmail = (EditText) findViewById(R.id.email);
        btnReset = (Button) findViewById(R.id.btn_reset_password);
        btnBack = (Button) findViewById(R.id.btn_back);
        progressBar = (ProgressBar) findViewById(R.id.progressBar);

        auth = FirebaseAuth.getInstance();
        btnBack.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });

        btnReset.setOnClickListener(new View.OnClickListener() {

```

```

        @Override
        public void onClick(View v) {

            String email = inputEmail.getText().toString().trim();

            if (TextUtils.isEmpty(email)) {
                Toast.makeText(getApplicationContext(), "Enter your registered
email id", Toast.LENGTH_SHORT).show();
                return;
            }

            progressBar.setVisibility(View.VISIBLE);
            auth.sendPasswordResetEmail(email)
                .addOnCompleteListener(new
OnCompleteListener<Void>() {
                @Override
                public void onComplete(@NonNull Task<Void> task)
{
                    if (task.isSuccessful()) {

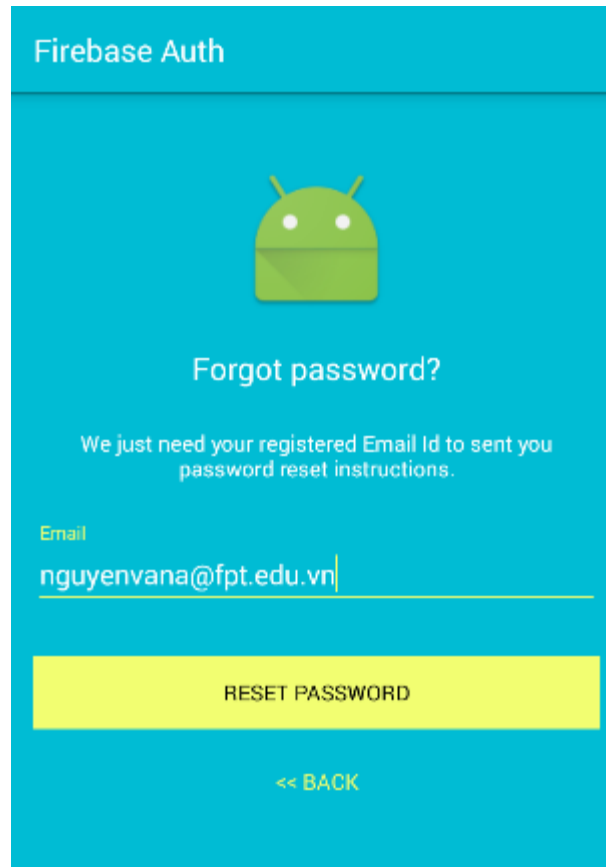
                        Toast.makeText(ResetPasswordActivity.this, "We have sent you instructions to
reset your password!", Toast.LENGTH_SHORT).show();
                    } else {

                        Toast.makeText(ResetPasswordActivity.this, "Failed to send reset email!",
                        Toast.LENGTH_SHORT).show();
                    }

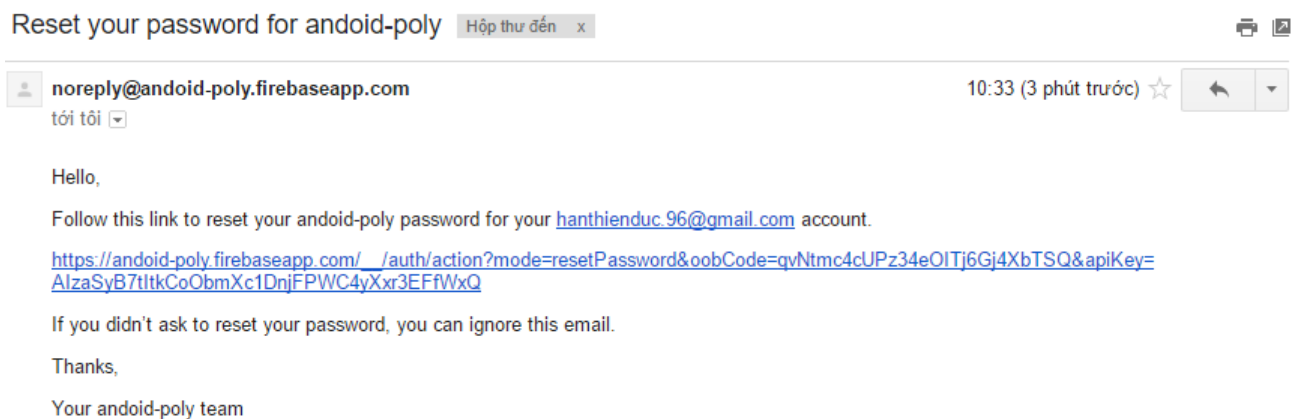
                    progressBar.setVisibility(View.GONE);
                }
            });
        }
    }
}

```

✓ Tiến hành test chức năng reset password



✓ Nhận email gửi tới



✓ Bây giờ ta mở file layout của main activity (activity_main.xml) và thêm đoạn code bên dưới.

```
<?xml version="1.0" encoding="utf-8" ?>
<android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
```

```

tools:context=".MainActivity">

<android.support.design.widget.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:elevation="0dp">

    <android.support.v7.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimaryDark"
        app:layout_scrollFlags="scroll|enterAlways"
        />

</android.support.design.widget.AppBarLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior">

    <EditText
        android:id="@+id/old_email"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/hint_email"
        android:inputType="textEmailAddress"
        android:maxLines="1"
        android:singleLine="true" />

    <EditText
        android:id="@+id/new_email"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/hint_new_email"
        android:inputType="textEmailAddress"
        android:maxLines="1"
        android:singleLine="true" />

    <EditText
        android:id="@+id/password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:focusableInTouchMode="true"
        android:hint="@string/prompt_password"
        android:imeActionId="@+id/login"
        android:imeOptions="actionUnspecified"
        android:inputType="textPassword"
        android:maxLines="1"
        android:singleLine="true" />

    <EditText

```

```

        android:id="@+id/newPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:focusableInTouchMode="true"
        android:hint="@string/new_pass"
        android:imeActionId="@+id/login"
        android:imeOptions="actionUnspecified"
        android:inputType="textPassword"
        android:maxLines="1"
        android:singleLine="true" />

<Button
    android:id="@+id/changeEmail"
    style="?android:textAppearanceSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:background="@color/colorPrimaryDark"
    android:text="@string/btn_change"
    android:textColor="@android:color/white"
    android:textStyle="bold" />

<Button
    android:id="@+id/changePass"
    style="?android:textAppearanceSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:background="@color/colorPrimaryDark"
    android:text="@string/btn_change"
    android:textColor="@android:color/white"
    android:textStyle="bold" />

<Button
    android:id="@+id/send"
    style="?android:textAppearanceSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:background="@color/colorPrimaryDark"
    android:text="@string/btn_send"
    android:textColor="@android:color/white"
    android:textStyle="bold" />

<ProgressBar
    android:id="@+id/progressBar"
    android:layout_width="30dp"
    android:layout_height="30dp"
    android:visibility="gone" />

<Button
    android:id="@+id/remove"
    style="?android:textAppearanceSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:background="@color/colorPrimaryDark"
    android:text="@string/btn_remove"

```

```

        android:textColor="@android:color/white"
        android:textStyle="bold" />

<Button
    android:id="@+id/change_email_button"
    style="?android:textAppearanceSmall"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="@string/change_email"
    android:textStyle="bold" />

<Button
    android:id="@+id/change_password_button"
    style="?android:textAppearanceSmall"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="@string/change_password"
    android:textStyle="bold" />

<Button
    android:id="@+id/sending_pass_reset_button"
    style="?android:textAppearanceSmall"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="@string/send_password_reset_email"
    android:textStyle="bold" />

<Button
    android:id="@+id/remove_user_button"
    style="?android:textAppearanceSmall"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="@string/remove_user"
    android:textStyle="bold" />

<Button
    android:id="@+id/sign_out"
    style="?android:textAppearanceSmall"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:background="@color/colorPrimary"
    android:text="@string/btn_sign_out"
    android:textColor="@android:color/white"
    android:textStyle="bold" />

</LinearLayout>

</android.support.design.widget.CoordinatorLayout>

```

✓ ở MainActivity.java và thêm code sau. Ta sẽ tiến tạo các chức năng để hoàn

FPT POLYTECHNIC

chính ứng dụng như ChangeEmail(), ChangePassword(), sendResetEmail(), RemoveUser() và signout().

```
package edu.poly.firebase_auth;

import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

public class MainActivity extends AppCompatActivity {

    private Button btnChangeEmail, btnChangePassword, btnSendResetEmail,
    btnRemoveUser,
        changeEmail, changePassword, sendEmail, remove, signOut;

    private EditText oldEmail, newEmail, password, newPassword;
    private ProgressBar progressBar;
    private FirebaseAuth.AuthStateListener authListener;
    private FirebaseAuth auth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //get firebase auth instance
        auth = FirebaseAuth.getInstance();

        //get current user
        final FirebaseUser user =
        FirebaseAuth.getInstance().getCurrentUser();

        authListener = new FirebaseAuth.AuthStateListener() {
            @Override
            public void onAuthStateChanged(@NonNull FirebaseAuth
        firebaseAuth) {
                FirebaseUser user = firebaseAuth.getCurrentUser();
                if (user == null) {
                    // user auth state is changed - user is null
                    // launch login activity
                    startActivity(new Intent(MainActivity.this,
        LoginActivity.class));
                    finish();
                }
            }
        };
    }
};
```

```

        btnChangeEmail = (Button) findViewById(R.id.change_email_button);
        btnChangePassword = (Button)
findViewById(R.id.change_password_button);
        btnSendResetEmail = (Button)
findViewById(R.id.sending_pass_reset_button);
        btnRemoveUser = (Button) findViewById(R.id.remove_user_button);
        changeEmail = (Button) findViewById(R.id.changeEmail);
        changePassword = (Button) findViewById(R.id.changePass);
        sendEmail = (Button) findViewById(R.id.send);
        remove = (Button) findViewById(R.id.remove);
        signOut = (Button) findViewById(R.id.sign_out);

        oldEmail = (EditText) findViewById(R.id.old_email);
        newEmail = (EditText) findViewById(R.id.new_email);
        password = (EditText) findViewById(R.id.password);
        newPassword = (EditText) findViewById(R.id.newPassword);

        oldEmail.setVisibility(View.GONE);
        newEmail.setVisibility(View.GONE);
        password.setVisibility(View.GONE);
        newPassword.setVisibility(View.GONE);
        changeEmail.setVisibility(View.GONE);
        changePassword.setVisibility(View.GONE);
        sendEmail.setVisibility(View.GONE);
        remove.setVisibility(View.GONE);

        progressBar = (ProgressBar) findViewById(R.id.progressBar);

        if (progressBar != null) {
            progressBar.setVisibility(View.GONE);
        }

        btnChangeEmail.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                oldEmail.setVisibility(View.GONE);
                newEmail.setVisibility(View.VISIBLE);
                password.setVisibility(View.GONE);
                newPassword.setVisibility(View.GONE);
                changeEmail.setVisibility(View.VISIBLE);
                changePassword.setVisibility(View.GONE);
                sendEmail.setVisibility(View.GONE);
                remove.setVisibility(View.GONE);
            }
        });
        changeEmail.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                progressBar.setVisibility(View.VISIBLE);
                if (user != null
&& !newEmail.getText().toString().trim().equals("")) {
                    user.updateEmail(newEmail.getText().toString().trim())
                        .addOnCompleteListener(new
OnCompleteListener<Void>() {
                                @Override
                                public void onComplete(@NonNull Task<Void>
task) {

```



```

        if (task.isSuccessful()) {
            Toast.makeText(MainActivity.this,
                "Email address is updated. Please sign in with new email id!",
                Toast.LENGTH_LONG).show();

            signOut();

        progressBar.setVisibility(View.GONE);
        } else {
            Toast.makeText(MainActivity.this,
                "Failed to update email!", Toast.LENGTH_LONG).show();

        progressBar.setVisibility(View.GONE);
        }
    });
} else if (newEmail.getText().toString().trim().equals("")) {
    {
        newEmail.setError("Enter email");
        progressBar.setVisibility(View.GONE);
    }
}
});
btnChangePassword.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        oldEmail.setVisibility(View.GONE);
        newEmail.setVisibility(View.GONE);
        password.setVisibility(View.GONE);
        newPassword.setVisibility(View.VISIBLE);
        changeEmail.setVisibility(View.GONE);
        changePassword.setVisibility(View.VISIBLE);
        sendEmail.setVisibility(View.GONE);
        remove.setVisibility(View.GONE);
    }
});
changePassword.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        progressBar.setVisibility(View.VISIBLE);
        if (user != null
            && !newPassword.getText().toString().trim().equals("")) {
            if (newPassword.getText().toString().trim().length() <
                6) {
                newPassword.setError("Password too short, enter
                minimum 6 characters");
                progressBar.setVisibility(View.GONE);
            } else {
                user.updatePassword(newPassword.getText().toString().trim())
                    .addOnCompleteListener(new
                    OnCompleteListener<Void>() {
                        @Override
                        public void onComplete(@NonNull
                        Task<Void> task) {
                            if (task.isSuccessful()) {
                                Toast.makeText(MainActivity.this, "Password is updated, sign in with new
                                password!", Toast.LENGTH_SHORT).show();

```

```

signOut();

progressBar.setVisibility(View.GONE);
    } else {

Toast.makeText(MainActivity.this, "Failed to update password!",
Toast.LENGTH_SHORT).show();

progressBar.setVisibility(View.GONE);
    }
    }
    });
    } else if
(newPassword.getText().toString().trim().equals("")) {
    newPassword.setError("Enter password");
    progressBar.setVisibility(View.GONE);
}
});
btnSendResetEmail.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
    oldEmail.setVisibility(View.VISIBLE);
    newEmail.setVisibility(View.GONE);
    password.setVisibility(View.GONE);
    newPassword.setVisibility(View.GONE);
    changeEmail.setVisibility(View.GONE);
    changePassword.setVisibility(View.GONE);
    sendEmail.setVisibility(View.VISIBLE);
    remove.setVisibility(View.GONE);
}
});

sendEmail.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
    progressBar.setVisibility(View.VISIBLE);
    if (!oldEmail.getText().toString().trim().equals("")) {
auth.sendPasswordResetEmail(oldEmail.getText().toString().trim())
    .addOnCompleteListener(new
OnCompleteListener<Void>() {
@Override
public void onComplete(@NonNull Task<Void>
task) {
        if (task.isSuccessful()) {
            Toast.makeText(MainActivity.this,
"Reset password email is sent!", Toast.LENGTH_SHORT).show();

progressBar.setVisibility(View.GONE);
        } else {
            Toast.makeText(MainActivity.this,
"Failed to send reset email!", Toast.LENGTH_SHORT).show();

progressBar.setVisibility(View.GONE);
        }
    }
}
}

```

```

        });
    } else {
        oldEmail.setError("Enter email");
        progressBar.setVisibility(View.GONE);
    }
}
});

btnRemoveUser.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        progressBar.setVisibility(View.VISIBLE);
        if (user != null) {
            user.delete()
                .addOnCompleteListener(new
OnCompleteListener<Void>() {
                    @Override
                    public void onComplete(@NonNull Task<Void>
task) {
                        if (task.isSuccessful()) {
                            Toast.makeText(MainActivity.this,
"Your profile is deleted:( Create a account now!",
Toast.LENGTH_SHORT).show();
                            startActivity(new
Intent(MainActivity.this, SignupActivity.class));
                            finish();

progressBar.setVisibility(View.GONE);
                        } else {
                            Toast.makeText(MainActivity.this,
"Failed to delete your account!", Toast.LENGTH_SHORT).show();
progressBar.setVisibility(View.GONE);
                        }
                    }
                });
        }
    }
});

signOut.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        signOut();
    }
});

}

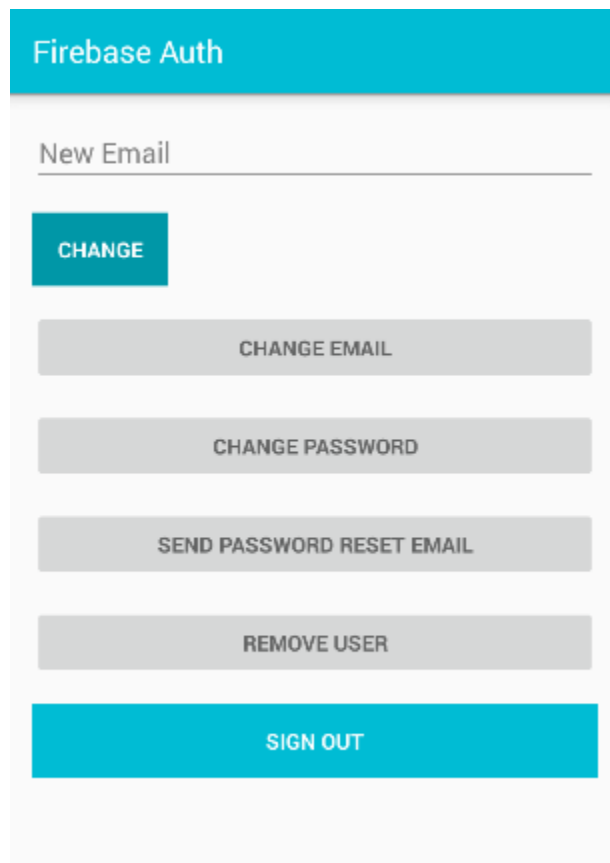
//sign out method
public void signOut() {
    auth.signOut();
}

@Override
protected void onResume() {
    super.onResume();
    progressBar.setVisibility(View.GONE);

```

```
}  
  
@Override  
public void onStart() {  
    super.onStart();  
    auth.addAuthStateListener(authListener);  
}  
  
@Override  
public void onStop() {  
    super.onStop();  
    if (authListener != null) {  
        auth.removeAuthStateListener(authListener);  
    }  
}  
}
```

✓ cuối cùng chạy và test ứng dụng.



BÀI 2 (2 ĐIỂM)

Viết ứng dụng android To-do app đơn giản kết hợp bài 1 sử dụng tính năng Realtime Database, Authentication, Messaging của Firebase.

HƯỚNG DẪN:

- ✓ Tạo project mới trên Firebase console các bước thực hiện đến tạo project android tương tự bài 1.
- ✓ Thêm INTERNET permission trong AndroidManifest.xml
- ✓ Mở app/build.gradle và thêm các thư viện sau

```
dependencies {
    compile 'com.google.firebase:firebase-database:9.4.0'
    compile 'com.google.firebase:firebase-auth:9.4.0'
    compile 'com.google.firebase:firebase-messaging:9.4.0'
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:design:24.2.1'
}
apply plugin: 'com.google.gms.google-services'
```

- ✓ Mở layout activity_main.xml và xây dựng giao diện sau:

```
<?xml version="1.0" encoding="utf-8" ?>
<android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:context=".MainActivity">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_behavior="@string/appbar_scrolling_view_behavior">

        <FrameLayout
            android:id="@+id/content"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            />

    </RelativeLayout>
</android.support.design.widget.CoordinatorLayout>
```

FPT POLYTECHNIC

- ✓ Tạo file layout mới đặt tên là home.xml. Layout này sẽ sử dụng cho HomeFragment.java

```
<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/home_relative_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="Large Text"
        android:id="@+id/textView"
        android:gravity="center"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"/>

    <ListView
        android:id="@+id/listView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textView"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true"/>

    <TextView
        android:id="@+id/emptyTxt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:text="No Todo Items, Add one." />

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentEnd="true"
        android:layout_gravity="end|bottom"
        android:layout_marginBottom="43dp"
        android:layout_marginEnd="43dp"
        android:layout_marginLeft="16dp"
        android:layout_marginRight="16dp"
        android:src="@android:drawable/ic_input_add" />
</RelativeLayout>
```

- ✓ Tiếp tục tạo layout login_signup.xml như sau

```
<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
```

```

        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:paddingBottom="@dimen/activity_vertical_margin"
        android:paddingLeft="@dimen/activity_horizontal_margin"
        android:paddingRight="@dimen/activity_horizontal_margin"
        android:paddingTop="@dimen/activity_vertical_margin"
    >

    <EditText
        android:id="@+id/email_edt"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="106dp"
        android:ems="10"
        android:hint="Email Id"
        android:inputType="textEmailAddress" />

    <EditText
        android:id="@+id/pwd_edt"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/email_edt"
        android:layout_centerHorizontal="true"
        android:ems="10"
        android:hint="Password"
        android:inputType="textPassword" />

    <LinearLayout
        android:id="@+id/linearLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/pwd_edt"
        android:layout_centerHorizontal="true"
        android:orientation="horizontal">

        <Button
            android:id="@+id/login_btn"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Login" />

        <Button
            android:id="@+id/signup_btn"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Sign Up" />
    </LinearLayout>

    <Button
        android:id="@+id/pwd_forgot_btn"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/linearLayout"

```

```

        android:layout_centerHorizontal="true"
        android:layout_marginTop="59dp"
        android:text="Forgot your password ?" />
</RelativeLayout>

```

✓ Tạo file layout mới đại diện cho 1 dòng trên listview, tên là list_row.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="@dimen/activity_horizontal_margin">

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight=".2"
        android:layout_gravity="center"
        android:text="Medium Text"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <ImageView
        android:id="@+id/imageView_done"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight=".4"
        android:src="@drawable/single_tick" />

    <ImageView
        android:id="@+id/imageView_delete"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight=".4"
        android:src="@drawable/delete" />
</LinearLayout>

```

✓ Tạo file layout cuối cùng đại diện cho custom dialog đặt tên là dialog_change_user_credential.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textEmailAddress"
        android:ems="10"
        android:id="@+id/edit1" />

```


✓ Tới class model mới đặt tên là Items.java.

```
package edu.poly.androidtodoapp;

import com.google.firebase.database.Exclude;

import java.util.HashMap;
import java.util.Map;

/**
 * Created by Han on 02/01/2017.
 */

public class Items {
    private String title;
    public String uid;
    public String author;
    public boolean isDone;

    public Items() {
        // Default constructor required for calls to
        DataSnapshot.getValue(Items.class)
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getUId() {
        return uid;
    }

    public void setUId(String uid) {
        this.uid = uid;
    }

    public String getAuthor() {
        return author;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    public boolean isDone() {
        return isDone;
    }
}
```

```

    public void setDone(boolean done) {
        isDone = done;
    }

    public Items(String uid, String author, String title, boolean isDone) {
        this.uid = uid;
        this.author = author;
        this.title = title;
        this.isDone = isDone;
    }

    @Exclude
    public Map<String, Object> toMap() {
        HashMap<String, Object> result = new HashMap<>();
        result.put("uid", uid);
        result.put("author", author);
        result.put("title", title);
        result.put("isDone", isDone);

        return result;
    }
}

```

✓ Qua thư mục menu tạo file menu mới đặt tên là menu_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <item
        android:id="@+id/action_change_email"
        app:showAsAction="never"
        android:title="Change Email" />

    <item
        android:id="@+id/action_change_password"
        app:showAsAction="never"
        android:title="Change Password" />

    <item
        android:id="@+id/action_delete_account"
        app:showAsAction="never"
        android:title="Delete account" />

    <item
        android:id="@+id/action_logout"
        app:showAsAction="never"
        android:title="Logout" />

</menu>

```

✓ Tạo class custom adapter cho listview đặt tên là CustomListAdapter.java,

code như sau.

```

package edu.poly.androidtodoapp;

import android.app.Activity;
import android.content.Context;
import android.graphics.Paint;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import android.widget.TextView;

import java.util.ArrayList;
import java.util.List;

/**
 * Created by Han on 02/01/2017.
 */

public class CustomListAdapter extends BaseAdapter{

    private final Activity context;
    private List<Items> items;
    private HomeFragment homeFragment;

    public CustomListAdapter(Context cont, ArrayList<Items> items,
HomeFragment homeFragment) {
        this.context = (Activity) cont;
        this.items = items;
        this.homeFragment = homeFragment;
    }

    static class ViewHolder {
        public TextView txtTitle;
        public ImageView imageViewDelete, imageViewDone;
    }

    @Override
    public int getCount() {
        return items.size();
    }

    @Override
    public Object getItem(int i) {
        return items.get(i);
    }

    @Override
    public long getItemId(int i) {
        return 0;
    }

    @Override
    public View getView(final int position, View rowView, ViewGroup parent)
{

```

```

ViewHolder holder = null;

if (rowView == null) {
    holder = new ViewHolder();
    LayoutInflater inflater = context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    rowView = inflater.inflate(R.layout.listitem_row, parent,
false);
    holder.txtTitle = (TextView)
rowView.findViewById(R.id.textView);
    holder.imageViewDone = (ImageView)
rowView.findViewById(R.id.imageView_done);
    holder.imageViewDelete = (ImageView)
rowView.findViewById(R.id.imageView_delete);
    rowView.setTag(holder);
} else {
    holder = (ViewHolder) rowView.getTag();
}

final Items todoItem = items.get(position);
holder.txtTitle.setText(todoItem.getTitle());
boolean isDone = todoItem.isDone();
if (isDone) {
    holder.imageViewDone.setImageResource(R.drawable.double_tick);
    holder.txtTitle.setPaintFlags(holder.txtTitle.getPaintFlags() |
Paint.STRIKE_THRU_TEXT_FLAG);
} else {
    holder.imageViewDone.setImageResource(R.drawable.single_tick);
    holder.txtTitle.setPaintFlags(holder.txtTitle.getPaintFlags() &
(~ Paint.STRIKE_THRU_TEXT_FLAG));
}
holder.imageViewDone.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (todoItem.isDone)
            todoItem.isDone = false;
        else
            todoItem.isDone = true;

        homeFragment.updateTodoItemDetails(todoItem, position);
    }
});
holder.imageViewDelete.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        homeFragment.deleteTodoItem(position);
    }
});
return rowView;
}
}

```

✓ Bây giờ ta sẽ tạo trang Home, tạo class mới đặt tên là HomeFragment.java.

FPT POLYTECHNIC

fragment này sẽ thực hiện các chức năng như là, add new todolist, change email, password, delete account và logout

```
package edu.poly.androidtodoapp;

import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.net.Uri;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentTransaction;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.ChildEventListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

/**
 * Created by Han on 02/01/2017.
 */

public class HomeFragment extends Fragment{
    private static final String TAG = "HomeFragment";
    //AuthStateListener that responds to changes in the user's sign-in state
    private FirebaseAuth.AuthStateListener fAuthListener;
    private FirebaseAuth fAuth;
    private FirebaseUser user;
    //Database references for handling CRUD operations
    private DatabaseReference mDatabase;

    public static HomeFragment homeFragment;
    private ProgressDialog mProgressDialog;
    private TextView msgTxt;
```

FPT POLYTECHNIC

```

private ListView listView;
private CustomListAdapter customListAdapter;
private static ArrayList<Items> arrayListTodo = new ArrayList<>();
private ArrayList<String> keysArray;

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                        Bundle savedInstanceState) {
    View rootView = inflater.inflate(R.layout.home, container, false);

    setHasOptionsMenu(true);
    return rootView;
}

@Override
public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {
    inflater.inflate(R.menu.menu_main, menu);
    super.onCreateOptionsMenu(menu, inflater);
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.action_change_email:
            showEditTextDialog(1); //To change email id
            break;
        case R.id.action_change_password:
            showEditTextDialog(0); //To change password
            break;
        case R.id.action_delete_account:
            deleteAccountAction(); //To delete user account
            break;
        case R.id.action_logout:
            signOutUser(); //To logout current user
            break;
    }
    return super.onOptionsItemSelected(item);
}

public static HomeFragment newInstance() {
    homeFragment = new HomeFragment();
    return homeFragment;
}

//Fragment transaction for changing fragment
private void mFragmentTransaction(Fragment mFragment) {
    FragmentManager fm = getFragmentManager();
    Fragment oldFragment =
fm.findFragmentByTag(MainActivity.FRAGMENT_TAG);
    FragmentTransaction ft = fm.beginTransaction();
    if (oldFragment != null)
        ft.remove(oldFragment);
    ft.replace(R.id.content, mFragment, MainActivity.FRAGMENT_TAG);
    ft.addToBackStack("replace");
    ft.commit();
}

```

```

//Used for showing progress dialog while deleting, changing email or
password
private void showProgressDialog(String title, String message) {
    if (mProgressDialog != null && mProgressDialog.isShowing())
        mProgressDialog.setMessage(message);
    else
        mProgressDialog = ProgressDialog.show(getActivity(), title,
message, true, false);
}
//Used to hide the progress dialog if displaying
private void hideProgressDialog() {
    if (mProgressDialog != null && mProgressDialog.isShowing()) {
        mProgressDialog.dismiss();
    }
}

@Override
public void onActivityCreated(@Nullable Bundle savedInstanceState) {
    super.onActivityCreated(savedInstanceState);
    msgTxt = (TextView) getView().findViewById(R.id.textView);
    //get firebase auth instance
    FirebaseAuth fAuth = FirebaseAuth.getInstance();
    //get firebase database instance
    FirebaseDatabase mDatabase = FirebaseDatabase.getInstance().getReference();

    //To get the currently signed-in user by calling getCurrentUser
    user = fAuth.getCurrentUser();
    //When a signed-out user uses an app feature that requires
authentication with Firebase, sign in the user anonymously
    FirebaseAuthListener = new FirebaseAuth.AuthStateListener() {
        @Override
        public void onAuthStateChanged(@NonNull FirebaseAuth
firebaseAuth) {
            if (user == null) {
                // user auth state is changed - user is null
                mFragmentManager(LoginSignUpFragment.newInstance());
            } else {
                // User is signed in
                // Name, email address, and profile photo Url
                String name = user.getDisplayName();
                String email = user.getEmail();
                Uri photoUrl = user.getPhotoUrl();

                // The user's ID, unique to the Firebase project. Do NOT
use this value to
                // authenticate with your backend server, if you have
one. Use

                // FirebaseUser.getToken() instead.
                String uid = user.getUid();
                msgTxt.setText("Hello " + email);
            }
        }
    };
    getView().findViewById(R.id.fab).setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {

```

```

        showEditTextDialog(2);
    }
});

    keysArray = new ArrayList<>();
    listView = (ListView) getView().findViewById(R.id.listView);
    //Custom Adapter for listview
    customListAdapter = new CustomListAdapter(getActivity(),
arrayListTodo, homeFragment);
    listView.setAdapter(customListAdapter);
    //Listview header text view
    TextView headerText = new TextView(getActivity());
    headerText.setText("ToDo Items");
    listView.addHeaderView(headerText);
    //Listview empty text view
    listView.setEmptyView(getView().findViewById(R.id.emptyTxt));
    listView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> adapterView, View view,
int i, long l) {
            updateTodoItemDetails(arrayListTodo.get(i), i);
        }
    });
}
//Child events are triggered in response to specific operations that
happen to the children of a node
ChildEventListener childEventListener = new ChildEventListener() {
    @Override
    public void onChildAdded(DataSnapshot dataSnapshot, String s) {
        Log.d(TAG, dataSnapshot.getKey() + ":" +
dataSnapshot.getValue().toString());
        String addedKey = dataSnapshot.getKey();
        if (!keysArray.contains(addedKey)) { //Checking for any
duplicate key entry
            Items todoItem = dataSnapshot.getValue(Items.class);
            arrayListTodo.add(todoItem);
            keysArray.add(dataSnapshot.getKey());
            updateView();
        }
    }

    @Override
    public void onChildChanged(DataSnapshot dataSnapshot, String s) {
        Log.d(TAG, dataSnapshot.getKey() + ":" +
dataSnapshot.getValue().toString());
        String changedKey = dataSnapshot.getKey();
        int changedIndex = keysArray.indexOf(changedKey);
        if (changedIndex > -1) { //Checking for key exist or not
            Items todoItem = dataSnapshot.getValue(Items.class);
            arrayListTodo.set(changedIndex, todoItem);
            updateView();
        }
    }

    @Override
    public void onChildRemoved(DataSnapshot dataSnapshot) {
        Log.d(TAG, dataSnapshot.getKey() + ":" +

```


FPT POLYTECHNIC

```

dataSnapshot.getValue().toString());
String deletedKey = dataSnapshot.getKey();
int removedIndex = keysArray.indexOf(deletedKey);
if (removedIndex > -1) { //Checking for key exist or not
    keysArray.remove(removedIndex);
    arrayListTodo.remove(removedIndex);
    updateView();
}

}

@Override
public void onChildMoved(DataSnapshot dataSnapshot, String s) {
}

@Override
public void onCancelled(DatabaseError databaseError) {
    Toast.makeText(getActivity(), "Could not update.",
Toast.LENGTH_SHORT).show();
    updateView();
}
};

//This method is used to refresh listview and list item has been
modified.
public void updateView() {
    customListAdapter.notifyDataSetChanged();
    listView.invalidate();
}

//This method is used to add an item to to-do list
public void addNewTodoItem(Items model) {
    // Create new to-do item at /user/$userid/$itemkey
    String key =
mDatabase.child("users").child(user.getUid()).push().getKey();
    Map<String, Object> todoValues = model.toMap();

    Map<String, Object> childUpdates = new HashMap<>();
    childUpdates.put("/users/" + user.getUid() + "/" + key, todoValues);
    mDatabase.updateChildren(childUpdates);
}

//This method is used to delete an item based on position
public void deleteTodoItem(int position) {
    String clickedKey = keysArray.get(position);

mDatabase.child("users").child(user.getUid()).child(clickedKey).removeValue(
);
}

//This method is used to update if any to-do item was changed based on
the position
public void updateTodoItemDetails(Items model, int position) {
    String clickedKey = keysArray.get(position);

mDatabase.child("users").child(user.getUid()).child(clickedKey).setValue(mod
el);
}

//This method is used to logout an user.

```

FPT POLYTECHNIC

```

private void signOutUser() {
    FirebaseAuth.signOut();
    mFragmentManager.beginTransaction(LoginSignUpFragment.newInstance());
}
//This method is used to delete current user account.
private void deleteAccountAction() {
    showProgressDialog("Delete Account", "Loading..");
    if (user != null) {
        //You can delete a user account with the delete method.
        user.delete().addOnCompleteListener(new
        OnCompleteListener<Void>() {
            @Override
            public void onComplete(@NonNull Task<Void> task) {
                if (task.isSuccessful()) {
                    Toast.makeText(getActivity(), "Your profile is
deleted:( Create a account now!", Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
    mFragmentManager.beginTransaction(LoginSignUpFragment.newInstance());
    hideProgressDialog();
} else {
    Toast.makeText(getActivity(), "Failed to delete your
account!", Toast.LENGTH_SHORT).show();
    hideProgressDialog();
}
}

//This method is used to update the current user password
private void changePasswordAction(String oldEmailId) {
    showProgressDialog("Change Account Password", "Loading..");
    if (user != null) {
        //You can send a password reset email to a user with the
        sendPasswordResetEmail method.
        FirebaseAuth.sendPasswordResetEmail(oldEmailId)
        .addOnCompleteListener(new OnCompleteListener<Void>() {
            @Override
            public void onComplete(@NonNull Task<Void> task) {
                if (task.isSuccessful()) {
                    Toast.makeText(getActivity(), "Reset
password email is sent!", Toast.LENGTH_SHORT).show();
                    hideProgressDialog();
                } else {
                    Toast.makeText(getActivity(), "Failed to
send reset email!", Toast.LENGTH_SHORT).show();
                    hideProgressDialog();
                }
            }
        });
        //or You can set a user's password with the updatePassword
method
        /*user.updatePassword(newPassword)
        .addOnCompleteListener(new OnCompleteListener<Void>() {
            @Override
            public void onComplete(@NonNull Task<Void> task) {
                if (task.isSuccessful()) {
                    Log.d(TAG, "User password updated.");
                }
            }
        });
    }
}

```

```

        }
    }
}

} else {
    hideProgressDialog();
}

}

//This method is used to update the current user email id
private void changeEmailAction(String newEmailId) {
    showProgressDialog("Change Account Email", "Loading..");
    if (user != null) {
        //You can set a user's email address with the updateEmail
method.
        user.updateEmail(newEmailId)
            .addOnCompleteListener(new OnCompleteListener<Void>() {
                @Override
                public void onComplete(@NonNull Task<Void> task) {
                    if (task.isSuccessful()) {
                        Toast.makeText(getActivity(), "Email address
is updated. Please sign in with new email id!", Toast.LENGTH_LONG).show();
                        signOutUser(); //signing out the user
                        hideProgressDialog();
                    } else {
                        Toast.makeText(getActivity(), "Failed to
update email!", Toast.LENGTH_LONG).show();
                        hideProgressDialog();
                    }
                }
            });
    } else {
        hideProgressDialog();
    }
}

//Used to capture new email id or password or new to-do item from user.
public void showEditTextDialog(final int forWhich) {
    AlertDialog.Builder dialogBuilder = new
AlertDialog.Builder(getActivity());
    LayoutInflater inflater = getActivity().getLayoutInflater();
    final View dialogView =
inflater.inflate(R.layout.dialog_change_user_credential, null);
    dialogBuilder.setView(dialogView);

    final EditText edt = (EditText) dialogView.findViewById(R.id.edit1);
    if (forWhich == 0) {
        dialogBuilder.setTitle("Changing Password");
        dialogBuilder.setMessage("Enter current email id");
    } else if (forWhich == 1) {
        dialogBuilder.setTitle("Changing Email Id");
        dialogBuilder.setMessage("Enter new email id");
    } else {
        dialogBuilder.setTitle("New ToDo Item");
        dialogBuilder.setMessage("Enter new todo list item");
    }

    dialogBuilder.setPositiveButton("Done", new
DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int whichButton) {

```

```

        //do something with edt.getText().toString();
        if (forWhich == 1)
            changeEmailAction(edt.getText().toString());
        else if (forWhich == 0)
            changePasswordAction(edt.getText().toString());
        else
            addNewTodoItem(new Items(user.getId(), user.getEmail(),
edt.getText().toString(), false));
    }
});
dialogBuilder.setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int whichButton) {
        //pass
    }
});
AlertDialog b = dialogBuilder.create();
b.show();
}

//Request for auth on start
@Override
public void onStart() {
    super.onStart();
    fAuth.addAuthStateListener(fAuthListener);
}

//Request for stop auth on stop
@Override
public void onStop() {
    super.onStop();
    if (fAuthListener != null) {
        fAuth.removeAuthStateListener(fAuthListener);
    }
}

//Request to attach listener for database on resume
@Override
public void onResume() {
    super.onResume();
    if (user != null && mDatabase != null)
mDatabase.child("users").child(user.getId()).addChildEventListener(childEvent
ntListener);
}

//Request to detach listener for database on pause
@Override
public void onPause() {
    super.onPause();
    if (mDatabase != null)
        mDatabase.removeEventListener(childEventListener);
}

```

}

- ✓ Tiếp đến ta sẽ xây dựng trang Login và Signup, tạo class mới đặt tên là LoginSignUpFragment.java. class này sẽ thực hiện các chức năng đăng ký đăng nhập. Code như sau

```
package edu.poly.androidtodoapp;

import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentTransaction;
import android.support.v7.app.AlertDialog;
import android.text.TextUtils;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.EditText;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;

/**
 * Created by Han on 02/01/2017.
 */

public class LoginSignUpFragment extends Fragment {
    private EditText edtEmail, edtPassword;
    private FirebaseAuth fAuth;
    private ProgressDialog mProgressDialog;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        return inflater.inflate(R.layout.login_signup, container, false);
    }

    public static LoginSignUpFragment newInstance() {
        return new LoginSignUpFragment();
    }

    private void mFragmentTransaction(Fragment mFragment) {
        FragmentManager fm = getFragmentManager();
        Fragment oldFragment =
            fm.findFragmentByTag(MainActivity.FRAGMENT_TAG);
        FragmentTransaction ft = fm.beginTransaction();
```

```

        if (oldFragment != null)
            ft.remove(oldFragment);
        ft.replace(R.id.content, mFragment, MainActivity.FRAGMENT_TAG);
        ft.addToBackStack("replace");
        ft.commit();
    }

    private void showProgressDialog(String title, String message) {
        if (mProgressDialog != null && mProgressDialog.isShowing())
            mProgressDialog.setMessage(message);
        else
            mProgressDialog = ProgressDialog.show(getActivity(), title,
message, true, false);
    }

    private void hideProgressDialog() {
        if (mProgressDialog != null && mProgressDialog.isShowing()) {
            mProgressDialog.dismiss();
        }
    }

    @Override
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);

        //get firebase auth instance
        FirebaseAuth fAuth = FirebaseAuth.getInstance();

        //To get the currently signed-in user by calling getCurrentUser
        if (fAuth.getCurrentUser() != null) {
            mFragmentManager.beginTransaction().add(R.id.content, HomeFragment.newInstance());
        }

        edtEmail = (EditText) getView().findViewById(R.id.email_edt);
        edtPassword = (EditText) getView().findViewById(R.id.pwd_edt);

        //To reset forgotten password with recovery email id
        getView().findViewById(R.id.pwd_forgot_btn).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                showRecoveryForgottenPasswordDialog();
            }
        });

        getView().findViewById(R.id.signup_btn).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {

                String email = edtEmail.getText().toString();
                String password = edtPassword.getText().toString();

                if (TextUtils.isEmpty(email)) {
                    Toast.makeText(getActivity(), "Enter email address!",
Toast.LENGTH_SHORT).show();
                    return;
                }
            }
        });
    }

```

FPT POLYTECHNIC

```

        if (TextUtils.isEmpty(password)) {
            Toast.makeText(getActivity(), "Enter password!",
Toast.LENGTH_SHORT).show();
            return;
        }

        showProgressDialog("Creating user account", "Loading..");
        //create user with email and password
        fAuth.createUserWithEmailAndPassword(email, password)
            .addOnCompleteListener(getActivity(), new
OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult>
task) {
                Toast.makeText(getActivity(),
"createUserWithEmail:onComplete:" + task.isSuccessful(),
Toast.LENGTH_SHORT).show();

                hideProgressDialog();
                // If sign in fails, display a message to
the user. If sign in succeeds
                // the auth state listener will be notified
and logic to handle the
                // signed in user can be handled in the
listener.

                if (!task.isSuccessful()) {
                    Toast.makeText(getActivity(), "SignUp
failed:: " + task.getException(),
                                Toast.LENGTH_SHORT).show();
                } else {

mFragmentTransaction(HomeFragment.newInstance());
                }
            }
        });

        getView().findViewById(R.id.login_btn).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String email = edtEmail.getText().toString();
                String password = edtPassword.getText().toString();

                if (TextUtils.isEmpty(email)) {
                    Toast.makeText(getActivity(), "Enter email address!",
Toast.LENGTH_SHORT).show();
                    return;
                }

                if (TextUtils.isEmpty(password)) {
                    Toast.makeText(getActivity(), "Enter password!",
Toast.LENGTH_SHORT).show();
                    return;
                }

                showProgressDialog("Signing user account", "Loading..");

```

```

        //authenticate user with email and password
        fAuth.signInWithEmailAndPassword(email, password)
            .addOnCompleteListener(getActivity(), new
OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult>
task) {
                    // If sign in fails, display a message to
the user. If sign in succeeds
                    // the auth state listener will be notified
and logic to handle the
                    // signed in user can be handled in the
listener.

                    hideProgressDialog();
                    if (task.isSuccessful()) {

mFragmentManager.beginTransaction().newInstance());
                    } else {
                        Toast.makeText(getActivity(), "Login
failed:: " + task.getException(), Toast.LENGTH_LONG).show();
                    }
                }
            });
    }

    //This method used to recovery fo

    public void showRecoveryForgottenPasswordDialog() {
        AlertDialog.Builder dialogBuilder = new
AlertDialog.Builder(getActivity());
        LayoutInflater inflater = getActivity().getLayoutInflater();
        final View dialogView =
inflater.inflate(R.layout.dialog_change_user_credential, null);
        dialogBuilder.setView(dialogView);

        final EditText edt = (EditText) dialogView.findViewById(R.id.edit1);
        dialogBuilder.setTitle("Recovery Forgotten Password");
        dialogBuilder.setMessage("Enter your registered email id");

        dialogBuilder.setPositiveButton("Done", new
DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int whichButton) {
                //passing the email id to sendPasswordResetEmail method
                recoveryForgottenPassword(edt.getText().toString());
            }
        });
        dialogBuilder.setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int whichButton) {
                //cancel dialog
            }
        });
        AlertDialog b = dialogBuilder.create();
        b.show();
    }
}

```



```

        //This method used to recovery forgotten password by sending reset link
        to registered email id.
        private void recoveryForgottenPassword(String currentEmailId) {

            if (TextUtils.isEmpty(currentEmailId)) {
                Toast.makeText(getActivity(), "Enter your registered email id",
                Toast.LENGTH_SHORT).show();
                return;
            }

            showProgressDialog("Resetting password", "Loading..");
            //You can send a password reset email to a user with the
            sendPasswordResetEmail method.
            FirebaseAuth.sendPasswordResetEmail(currentEmailId)
                .addOnCompleteListener(new OnCompleteListener<Void>() {
                    @Override
                    public void onComplete(@NonNull Task<Void> task) {
                        if (task.isSuccessful()) {
                            Toast.makeText(getActivity(), "We have sent you
                            instructions to reset your password!", Toast.LENGTH_SHORT).show();
                        } else {
                            Toast.makeText(getActivity(), "Failed to send
                            reset email!", Toast.LENGTH_SHORT).show();
                        }
                        hideProgressDialog();
                    }
                });
        }
    }
}

```

✓ Tiếp tục tạo 2 class mới xử lý tính năng message của firebase

✓ Tạo class mới đặt tên là MyFirebaseInstanceIdService.java và code như sau

```

package edu.poly.androidtodoapp;

import android.util.Log;

import com.google.firebase.iid.FirebaseInstanceId;
import com.google.firebase.iid.FirebaseInstanceIdService;

/**
 * Created by chiru on 10-08-2016.
 */
public class MyFirebaseInstanceIdService extends FirebaseInstanceIdService {

    private static final String TAG = "MyFirebaseIIDService";

    @Override
    public void onTokenRefresh() {

        //Getting registration token
    }
}

```

```
String refreshedToken = FirebaseInstanceId.getInstance().getToken();

//Displaying token in logcat
Log.e(TAG, "Refreshed token: " + refreshedToken);

// If you want to send messages to this application instance or
// manage this apps subscriptions on the server side, send the
// Instance ID token to your app server.
sendRegistrationToServer(refreshedToken);
}

private void sendRegistrationToServer(String token) {
    //You can implement this method to store the token on your server
    //Not required for current project
}
}
```

✓ Tạo tiếp class mới đặt tên là MyFirebaseMessagingService.java và code như sau

```
package edu.poly.androidtodoapp;

import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.media.RingtoneManager;
import android.net.Uri;
import android.support.v4.app.NotificationCompat;
import android.util.Log;

import com.google.firebase.messaging.FirebaseMessagingService;
import com.google.firebase.messaging.RemoteMessage;

/**
 * Created by Han on 02/01/2017.
 */

public class MyFirebaseMessagingService extends FirebaseMessagingService {
    private static final String TAG = "StartingAndroid";

    @Override
    public void onMessageReceived(RemoteMessage remoteMessage) {

        //It is optional
        Log.e(TAG, "From: " + remoteMessage.getFrom());
        Log.e(TAG, "Notification Message Body: " +
remoteMessage.getNotification().getBody());

        //Calling method to generate notification

sendNotification(remoteMessage.getNotification().getTitle(), remoteMessage.ge
```

FPT POLYTECHNIC

```
tNotification().getBody());
    }

    //This method is only generating push notification
    private void sendNotification(String title, String messageBody) {
        Intent intent = new Intent(this, MainActivity.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        PendingIntent pendingIntent = PendingIntent.getActivity(this, 0,
            intent,
                PendingIntent.FLAG_ONE_SHOT);

        Uri defaultSoundUri=
            RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
        NotificationCompat.Builder notificationBuilder = new
            NotificationCompat.Builder(this)
                .setSmallIcon(R.mipmap.ic_launcher)
                .setContentTitle(title)
                .setContentText(messageBody)
                .setAutoCancel(true)
                .setSound(defaultSoundUri)
                .setContentIntent(pendingIntent);

        NotificationManager notificationManager=
            (NotificationManager)
            getSystemService(Context.NOTIFICATION_SERVICE);

        notificationManager.notify(0, notificationBuilder.build());
    }
}
```

✓ Mở file AndroidManifest.xml và đăng ký 2 service sau

```
<service
    android:name=".services.MyFirebaseMessagingService">
    <intent-filter>
        <action android:name="com.google.firebase.MESSAGING_EVENT"/>
    </intent-filter>
</service>

<service
    android:name=".services.MyFirebaseInstanceIdService">
    <intent-filter>
        <action android:name="com.google.firebase.INSTANCE_ID_EVENT"/>
    </intent-filter>
</service>
```

✓ Cuối cùng ta mở file MainActivity.java và tiến hành thực thi các code sau.

```
package edu.poly.androidtodoapp;

import android.app.ProgressDialog;
import android.os.PersistableBundle;
import android.support.v4.app.Fragment;
```

FPT POLYTECHNIC

```

import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentTransaction;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.Toolbar;
import android.widget.ListView;
import android.widget.TextView;

import com.google.firebase.analytics.FirebaseAnalytics;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.messaging.FirebaseMessaging;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

public class MainActivity extends AppCompatActivity {

    private FirebaseAuth fAuth;
    private Fragment mContent;
    public final static String FRAGMENT_TAG = "MyFragment";
    private FirebaseAnalytics mFirebaseAnalytics;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //setSupportActionBar((Toolbar) findViewById(R.id.toolbar));
        //get firebase auth instance
        fAuth = FirebaseAuth.getInstance();
        //firebase messaging instance
        FirebaseMessaging fMsg = FirebaseMessaging.getInstance();
        //firebase instance to subscribe topics
        fMsg.subscribeToTopic("todos");
        //firebase analytics instance
        mFirebaseAnalytics = FirebaseAnalytics.getInstance(this);

        //Sets whether analytics collection is enabled for this app on this
device.
        mFirebaseAnalytics.setAnalyticsCollectionEnabled(true);

        //Sets the minimum engagement time required before starting a
session. The default value is 10000 (10 seconds). Let's make it 20 seconds
just for the fun
        mFirebaseAnalytics.setMinimumSessionDuration(20000);

        //Sets the duration of inactivity that terminates the current
session. The default value is 1800000 (30 minutes).
        mFirebaseAnalytics.setSessionTimeoutDuration(600000);
        if (savedInstanceState == null) {
            // To get the currently signed-in user by calling getCurrentUser
            // If no currently user signed in, we'll redirect the user to
login screen else home screen
            Bundle bundle = new Bundle();

```

```

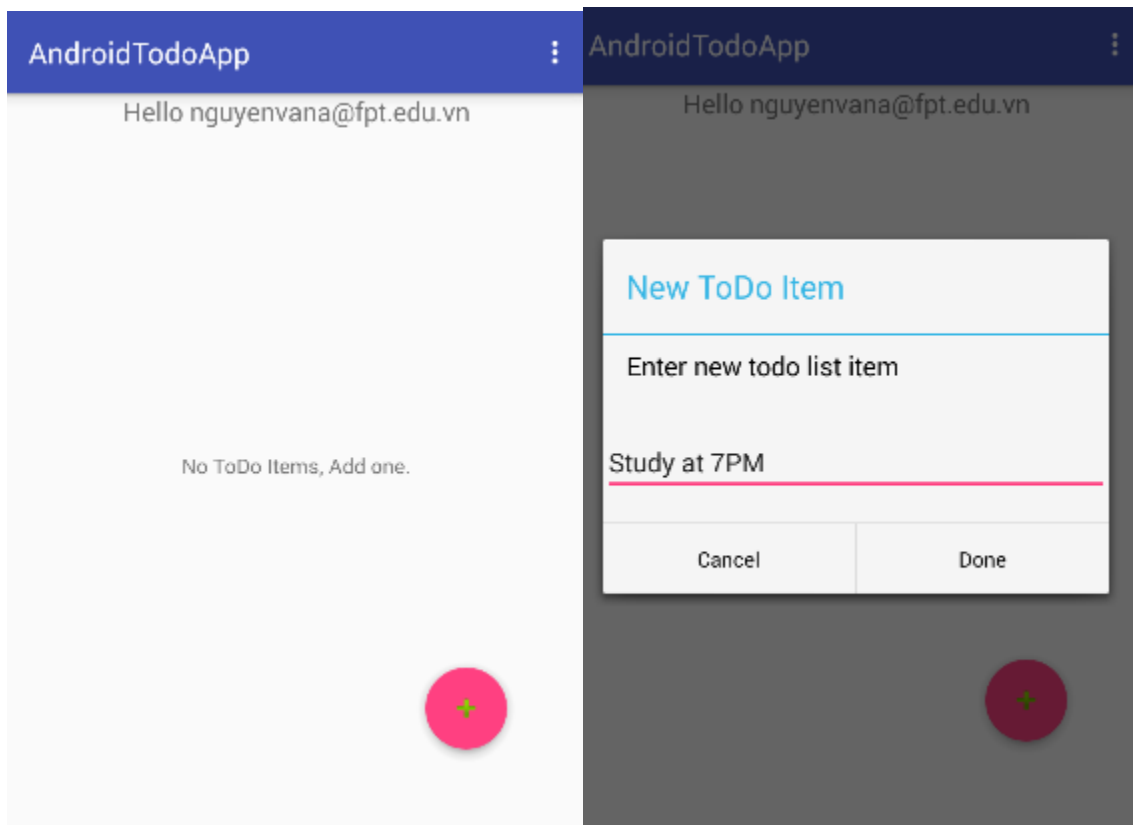
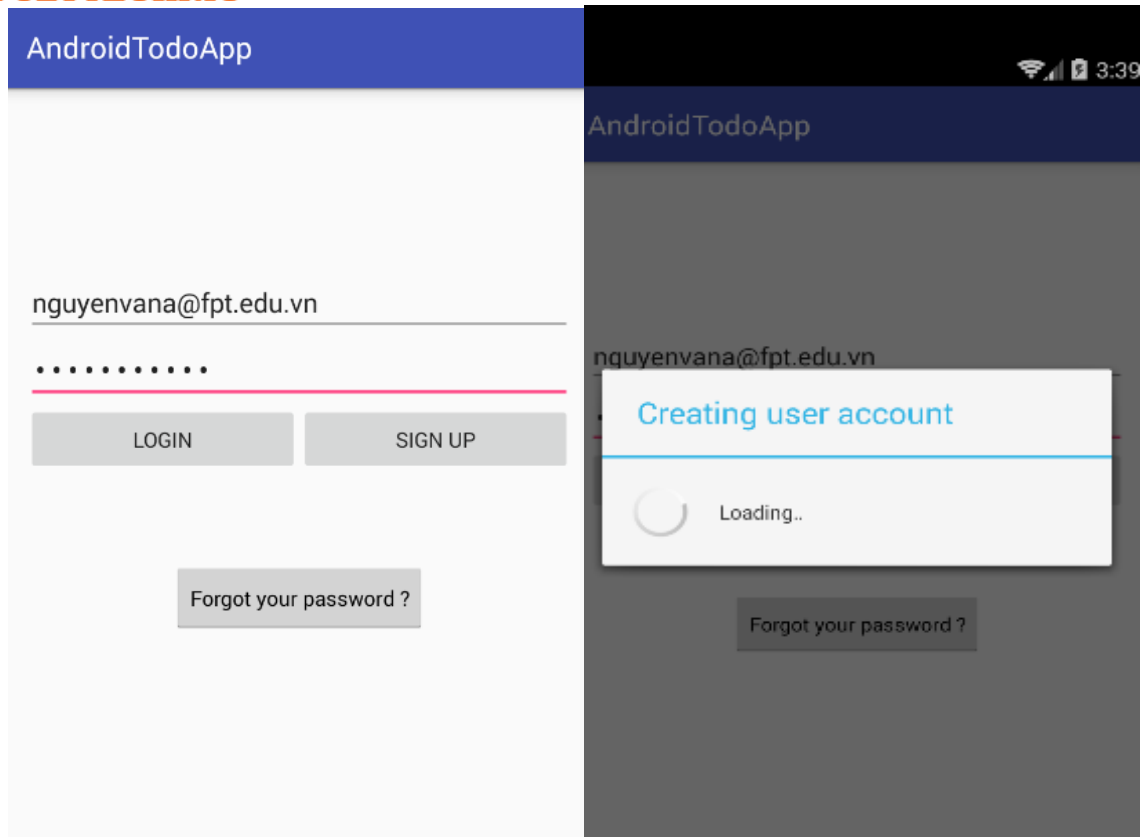
        if (fAuth.getCurrentUser() != null) {
            mContent = HomeFragment.newInstance();
            bundle.putString(FirebaseAnalytics.Param.DESTINATION, "Home
Fragment");
        } else {
            mContent = LoginSignUpFragment.newInstance();
            bundle.putString(FirebaseAnalytics.Param.DESTINATION, "Login
Fragment");
        }
        FragmentManager fm = getSupportFragmentManager();
        // This Log event logs whether the user has logged in before or
not
        bundle.putString(FirebaseAnalytics.Param.ORIGIN, "Main
Activity");
        mFirebaseAnalytics.logEvent(FirebaseAnalytics.Event.APP_OPEN,
bundle);

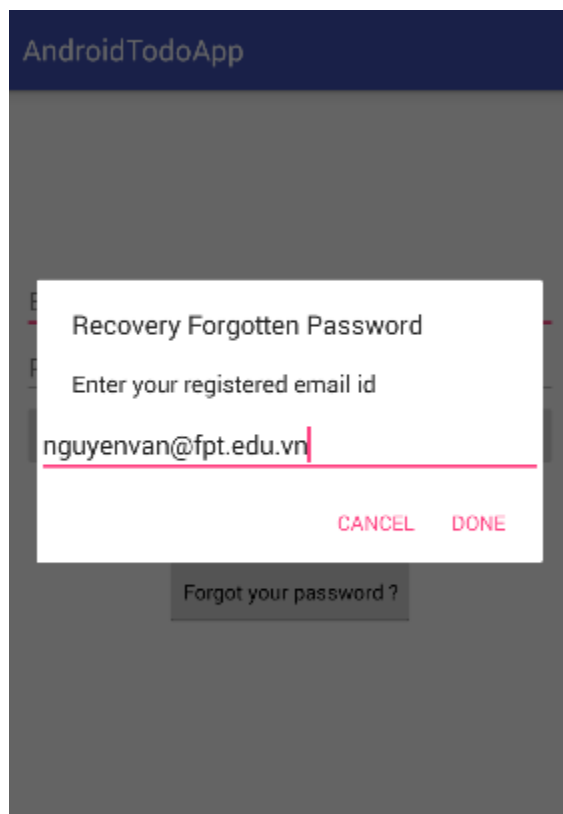
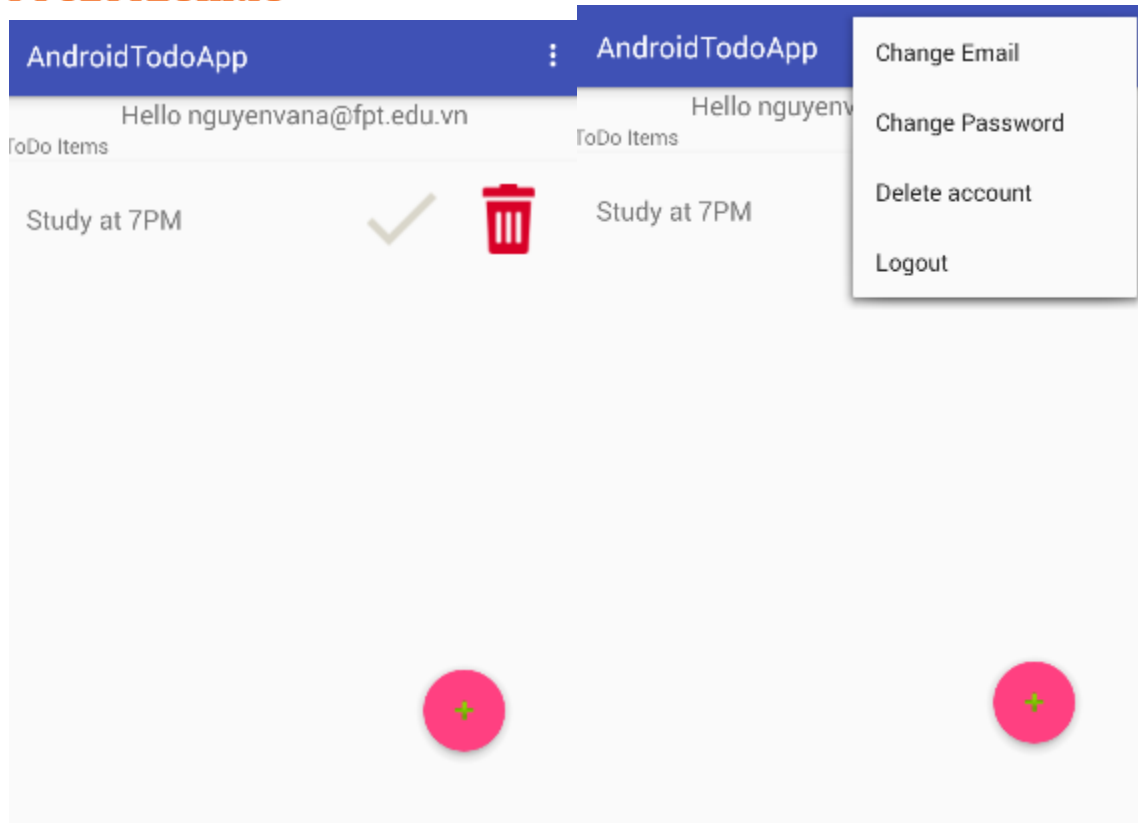
        FragmentTransaction ft = fm.beginTransaction();
        ft.add(R.id.content, mContent, FRAGMENT_TAG);
        ft.commit();
    } else {
        mContent = getSupportFragmentManager().getFragment(
            savedInstanceState, FRAGMENT_TAG);
    }
}

@Override
public void onSaveInstanceState(Bundle outState, PersistableBundle
outPersistentState) {
    super.onSaveInstanceState(outState, outPersistentState);
    mContent =
getSupportFragmentManager().findFragmentByTag(FRAGMENT_TAG);
    if (mContent != null)
        getSupportFragmentManager().putFragment(outState, FRAGMENT_TAG,
mContent);
}
}

```

✓ Chạy ứng dụng và xem kết quả như bên dưới



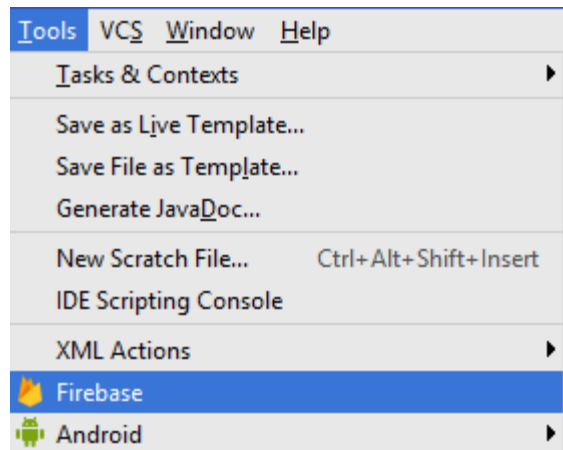


FPT POLYTECHNIC**BÀI 3 (2 ĐIỂM)**


Viết ứng dụng android đơn giản upload hình từ máy điện thoại lên project android trên Firebase. Sử dụng tính năng storage.


HƯỚNG DẪN:

- ✓ Mở Android Studio và tạo project mới, sau khi tạo tiến hành import firebase vào project theo cách sau Tools – Firebase (phiên bản Android studio 2.2 trở lên)










- ✓ Chọn tiếp tính năng storage như bên dưới

Assistant 

 **Firebase**

Firebase gives you the tools and infrastructure from Google to help you develop, grow and earn money from your app. [Learn more](#)

sign-in, Facebook and other login providers, [more info](#)

- ▶  **Realtime Database**
Store and sync data in realtime across all connected clients. [More info](#)
- ▼  **Storage**
Store and retrieve large files like images, audio, and video without writing server-side code. [More info](#)
 [Upload and download a file with Firebase Storage](#)
- ▶  **Remote Config**
Customize and experiment with app behavior using cloud-based configuration parameters. [More info](#)
- ▶  **Test Lab**
Test your apps against a wide range of physical devices hosted in Google's cloud. [More info](#)
- ▶  **Crash Reporting**
Get actionable insights and reports on app crashes, ANRs or other errors. [More info](#)
- ▶  **Notifications**
Send targeted notifications to engage the right users at the right time. [More info](#)

- ✓ Click vào đường link Upload and download a file with Firebase Storage. Tiến hành connect với Firebase để tạo project và import thư viện cần thiết vào

Upload and download a file with Firebase Storage

Firebase Storage provides secure file uploads and downloads for your Firebase apps, regardless of network quality. You can use it to store images, audio, video, or other user-generated content.

[Launch in browser](#)

1 Connect your app to Firebase

 Connected

2 Add Firebase Storage to your app

 Dependencies set up correctly

3 Create a reference

Declare a [StorageReference](#) and initialize it in the `onCreate` method.

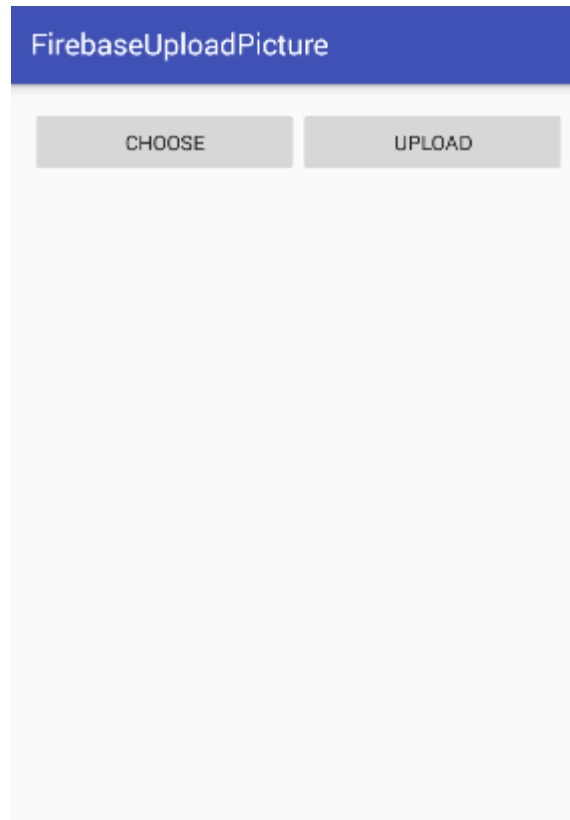
```
private StorageReference mStorageRef;
```

```
mStorageRef = FirebaseStorage.getInstance().getRefere:
```

4 Upload a file

The simplest way to upload to your storage bucket is by uploading a

- ✓ Thêm INTERNET permission cho ứng dụng
- ✓ Bây giờ mở file `activity_main.xml` và xây dựng giao diện đơn giản như bên dưới



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="edu.poly.firebaseuploadpicture.MainActivity">

    <LinearLayout
        android:id="@+id/linearLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <Button
            android:id="@+id/buttonChoose"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Choose" />

        <Button
            android:id="@+id/buttonUpload"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
```

```

        android:layout_weight="1"
        android:text="Upload" />

</LinearLayout>

<ImageView
    android:id="@+id/imageView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@id/linearLayout" />
</RelativeLayout>

```

✓ Mở MainActivity.java và code chức năng upload như bên dưới.

```

package edu.poly.firebaseuploadpicture;

import android.app.ProgressDialog;
import android.content.Intent;
import android.graphics.Bitmap;
import android.net.Uri;
import android.provider.MediaStore;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.OnProgressListener;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;

import java.io.IOException;

public class MainActivity extends AppCompatActivity implements
View.OnClickListener {

    //a constant to track the file chooser intent
    private static final int PICK_IMAGE_REQUEST = 234;

    //Buttons
    private Button buttonChoose;
    private Button buttonUpload;

    //ImageView
    private ImageView imageView;

    //a Uri object to store file path
    private Uri filePath;

    private StorageReference mStorageRef;

    @Override

```

FPT POLYTECHNIC

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    //getting views from layout
    buttonChoose = (Button) findViewById(R.id.buttonChoose);
    buttonUpload = (Button) findViewById(R.id.buttonUpload);

    imageView = (ImageView) findViewById(R.id.imageView);
    mStorageRef = FirebaseStorage.getInstance().getReference();
    //attaching listener
    buttonChoose.setOnClickListener(this);
    buttonUpload.setOnClickListener(this);
}

@Override
public void onClick(View view) {
    //if the clicked button is choose
    if (view == buttonChoose) {
        showFileChooser();
    }
    //if the clicked button is upload
    else if (view == buttonUpload) {
        uploadFile();
    }
}

//method to show file chooser
private void showFileChooser() {
    Intent intent = new Intent();
    intent.setType("image/*");
    intent.setAction(Intent.ACTION_GET_CONTENT);
    startActivityForResult(Intent.createChooser(intent, "Select
Picture"), PICK_IMAGE_REQUEST);
}

//this method will upload the file
private void uploadFile() {
    //if there is a file to upload
    if (filePath != null) {
        //displaying a progress dialog while upload is going on
        final ProgressDialog progressDialog = new ProgressDialog(this);
        progressDialog.setTitle("Uploading");
        progressDialog.show();

        StorageReference riversRef =
mStorageRef.child("images/pic.jpg");
        riversRef.putFile(filePath)
            .addOnSuccessListener(new
OnSuccessListener<UploadTask.TaskSnapshot>() {
            @Override
            public void onSuccess(UploadTask.TaskSnapshot
taskSnapshot) {

                //if the upload is successfull
                //hiding the progress dialog
                progressDialog.dismiss();

                //and displaying a success toast

```

```

        Toast.makeText(getApplicationContext(), "File
Uploaded ", Toast.LENGTH_LONG).show();
    }
})
.addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception exception)
{
        //if the upload is not successfull
        //hiding the progress dialog
        progressDialog.dismiss();

        //and displaying error message
        Toast.makeText(getApplicationContext(),
exception.getMessage(), Toast.LENGTH_LONG).show();
    }
})
.addOnProgressListener(new
OnProgressListener<UploadTask.TaskSnapshot>() {
    @Override
    public void onProgress(UploadTask.TaskSnapshot
taskSnapshot) {
        //calculating progress percentage
        double progress = (100.0 *
taskSnapshot.getBytesTransferred()) / taskSnapshot.getTotalByteCount();

        //displaying percentage in progress dialog
        progressDialog.setMessage("Uploaded " + ((int)
progress) + "%...");
    }
});
}
//if there is not any file
else {
    //you can display an error toast
}

//handling the image chooser activity result
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == PICK_IMAGE_REQUEST && resultCode == RESULT_OK &&
data != null && data.getData() != null) {
        filePath = data.getData();
        try {
            Bitmap bitmap =
MediaStore.Images.Media.getBitmap(getContentResolver(), filePath);
            imageView.setImageBitmap(bitmap);

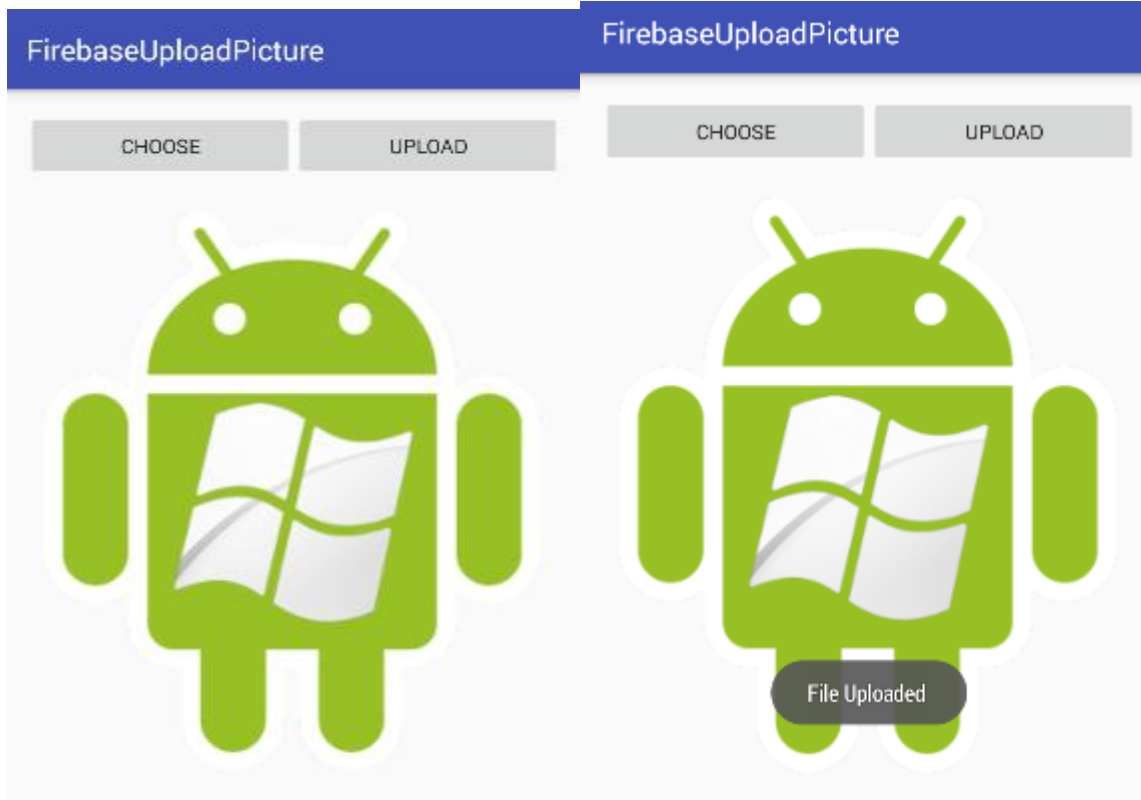
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
}

```

- ✓ Mở project firebase đã tạo trong browser và vào mục storage thay đổi rules như sau

```
1  service firebase.storage {  
2    match /b/fir-uploadpicture.appspot.com/o {  
3      match /{allPaths=**} {  
4        allow read, write: if true;  
5      }  
6    }  
7  }  
8
```

- ✓ Lưu ý chỉ nên sử dụng việc thay đổi này cho phát triển app ở dạng test. Khi sinh viên triển khai ứng dụng thật thì cần đăng nhập nếu không bất cứ ai cũng có thể truy store của app
- ✓ Chạy ứng dụng và xem kết quả như sau




Storage

FILES

RULES

gs://fir-uploadpicture.appspot.com > images UPLOAD FILE + ↺

<input type="checkbox"/>	Name	Size	Type	Last modified
<input type="checkbox"/>	 pic.jpg	22.59 KB	image/png	Jan 2, 2017

BÀI 5 (2 ĐIỂM):

Giảng viên cho thêm