**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №4

по дисциплине: **«Разработка приложений для мобильных устройств»**

на тему: **Интерфейс пользователя в *Android***

Выполнил: студент гр. ИТП-31

Дашкевич Д.А.

Принял: ассистент

Белявский Е. В.

Гомель 2019

**Цель**: изучить разметки и использование элементов управления (*Button*, *TextView*, *EditText*, *RecyclerView*) в *Android*.

**Ход работы**

**Вариант 2**

# **Задание:** используя встроенные возможности GMS библиотеки Google, создать пользовательский интерфейс для работы с Goggle аккаунтом. Вывести общедоступную информацию этого аккаунта для пользователя.

**Результат работы приложения:**

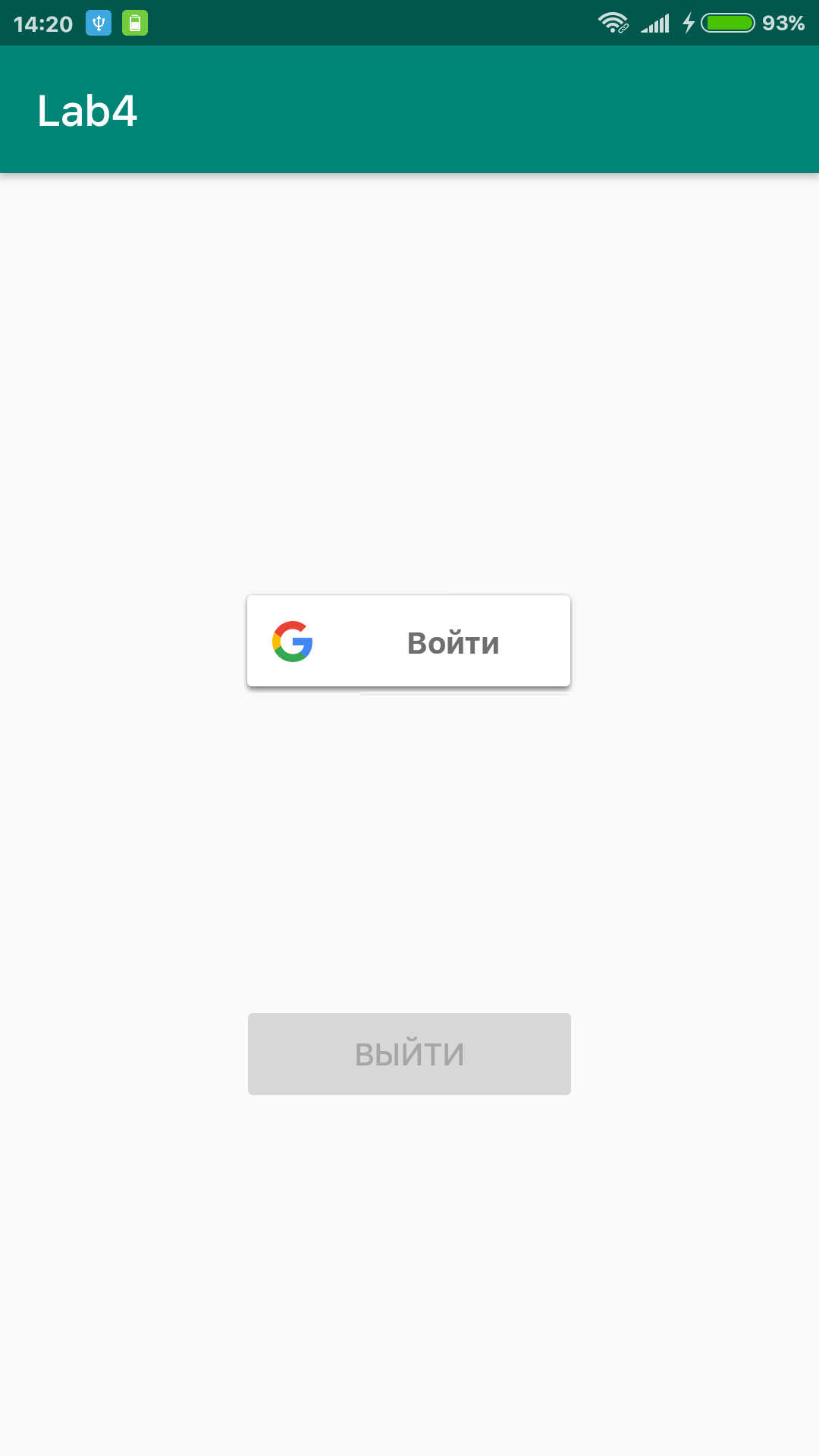


Рисунок 1 – Главное окно приложения с активной кнопкой “Войти” и неактивной кнопкой “Выйти”

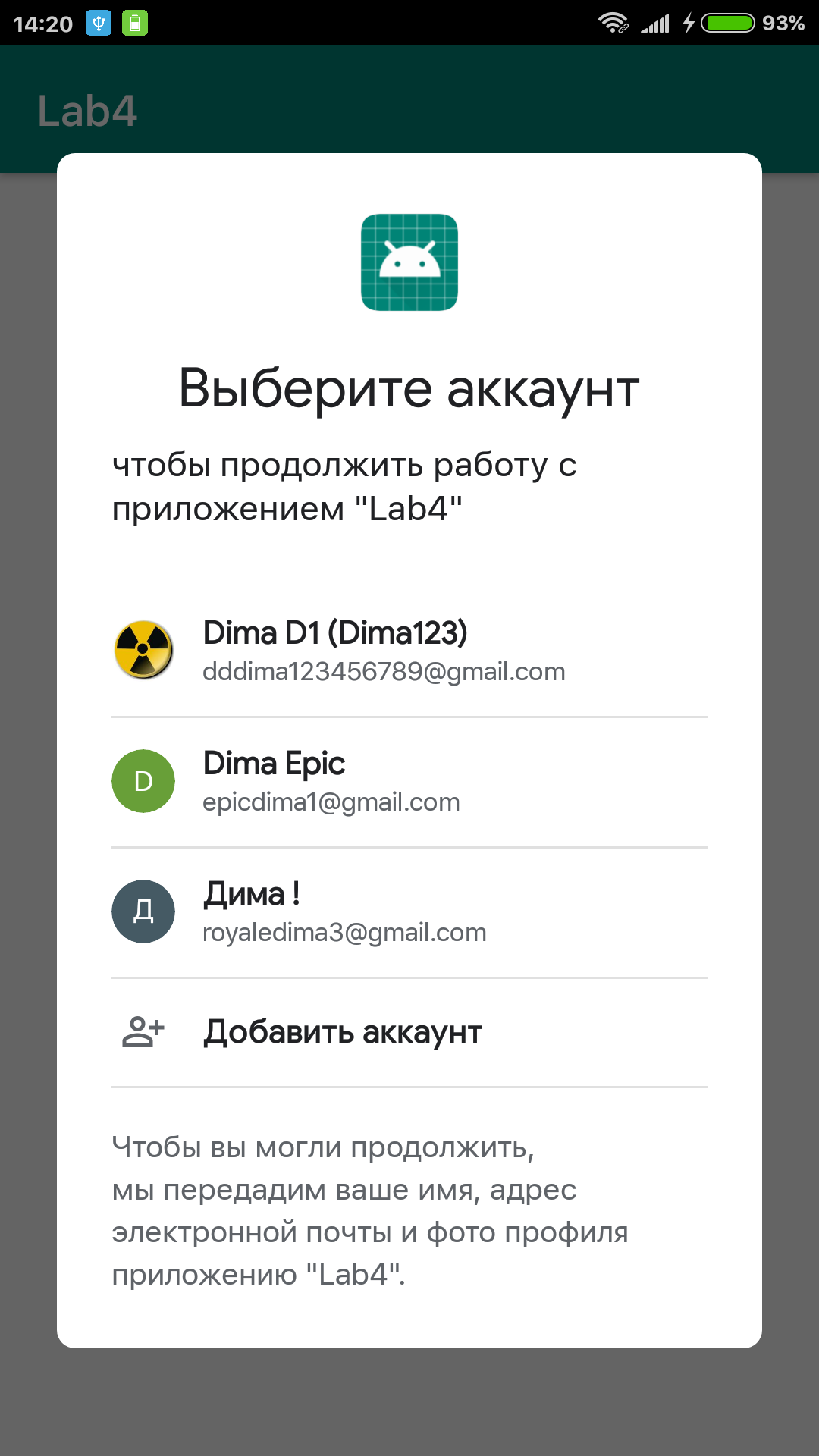


Рисунок 2 – Диалоговое окно выбора Google аккаунта из списка доступных на телефоне

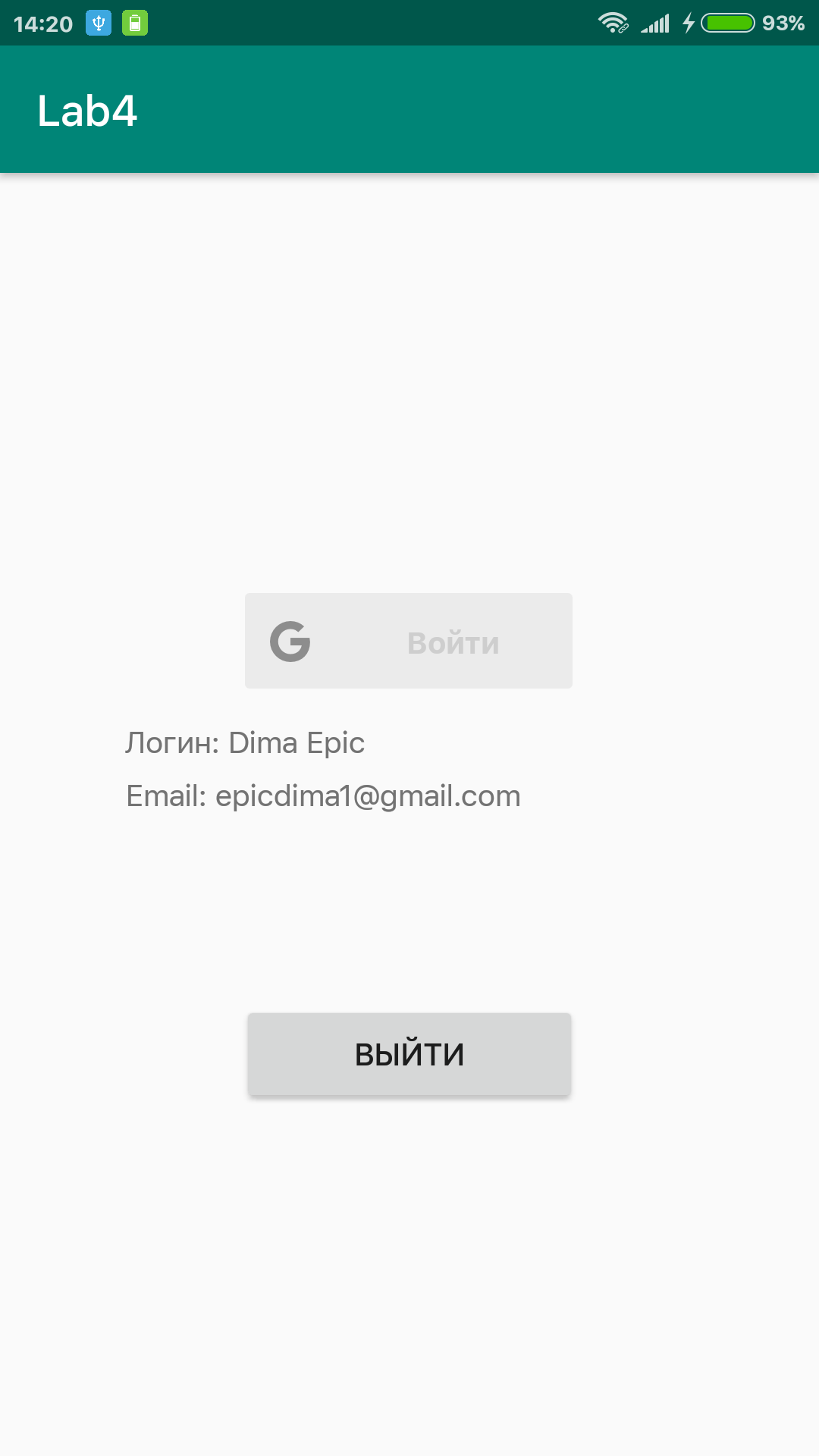


Рисунок 3 – Главное окно приложения с выводом логина и email Google аккаунта, активная кнопка “Выйти” и неактивная кнопка “Войти”

**Исходный код приложения**

**MainActivity.java:**

**public class** MainActivity **extends** AppCompatActivity **implements** GoogleApiClient.ConnectionCallbacks, GoogleApiClient.OnConnectionFailedListener,  
 View.OnClickListener {  
  
 **private static final int *SIGN\_IN\_REQUEST\_CODE*** = 0;  
  
 **private** GoogleApiClient **mGoogleApiClient**;  
  
 **private** SignInButton **mSignInButton**;  
 **private** Button **mSignOutButton**;  
 **private** TextView **mNameTextView**;  
 **private** TextView **mEmailTextView**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 **mSignInButton** = findViewById(R.id.***sign\_in\_button***);  
 **mSignOutButton** = findViewById(R.id.***sign\_out\_button***);  
 **mNameTextView** = findViewById(R.id.***displayname\_textview***);  
 **mEmailTextView** = findViewById(R.id.***email\_textview***);  
  
 **mSignInButton**.setOnClickListener(**this**);  
 **mSignOutButton**.setOnClickListener(**this**);  
  
 **mGoogleApiClient** = buildGoogleApiClient();  
 }  
  
 **private** GoogleApiClient buildGoogleApiClient() {  
 GoogleSignInOptions gso = **new** GoogleSignInOptions.Builder(GoogleSignInOptions.***DEFAULT\_SIGN\_IN***)  
 .requestEmail()  
 .build();  
  
 **return new** GoogleApiClient.Builder(**this**)  
 .addConnectionCallbacks(**this**)  
 .addOnConnectionFailedListener(**this**)  
 .enableAutoManage(**this**, **this**)  
 .addApi(Auth.***GOOGLE\_SIGN\_IN\_API***, gso)  
 .build();  
 }  
  
 @Override  
 **protected void** onStart() {  
 **super**.onStart();  
 **mGoogleApiClient**.connect();  
 }  
  
 @Override  
 **protected void** onStop() {  
 **mGoogleApiClient**.disconnect();  
 **super**.onStop();  
 }  
  
 @Override  
 **public void** onClick(View v) {  
 **if** (!**mGoogleApiClient**.isConnecting()) {  
 **switch** (v.getId()) {  
 **case** R.id.***sign\_in\_button***:  
 Intent signInIntent = Auth.***GoogleSignInApi***.getSignInIntent(**mGoogleApiClient**);  
 startActivityForResult(signInIntent, ***SIGN\_IN\_REQUEST\_CODE***);  
 **break**;  
 **case** R.id.***sign\_out\_button***:  
 onSignedOut();  
 **break**;  
 }  
 }  
 }  
  
 @Override  
 **public void** onConnected(@Nullable Bundle bundle) {  
 OptionalPendingResult<GoogleSignInResult> opr  
 = Auth.***GoogleSignInApi***.silentSignIn(**mGoogleApiClient**);  
  
 opr.setResultCallback(**new** ResultCallback<GoogleSignInResult>() {  
 @Override  
 **public void** onResult(@NonNull GoogleSignInResult result) {  
 **if** (result.isSuccess()) {  
 **try** {  
 **mSignInButton**.setEnabled(**false**);  
 **mSignOutButton**.setEnabled(**true**);  
 GoogleSignInAccount account = result.getSignInAccount();  
 **if** (account != **null**) {  
 **mNameTextView**.setText(String.*format*(**"Логин: %s"**,  
 account.getDisplayName()));  
 **mEmailTextView**.setText(String.*format*(**"Email: %s"**,  
 account.getEmail()));  
 }  
 **mGoogleApiClient**.disconnect();  
 **mGoogleApiClient**.connect();  
 }  
 **catch**(Exception ex){  
 ex.printStackTrace();  
 }  
 }  
 }  
 });  
 }  
  
 @Override  
 **public void** onConnectionSuspended(**int** i) {  
 **mGoogleApiClient**.connect();  
 }  
  
 @Override  
 **public void** onConnectionFailed(@NonNull ConnectionResult connectionResult) {  
 **try** {  
 connectionResult.startResolutionForResult(**this**, ***SIGN\_IN\_REQUEST\_CODE***);  
 } **catch** (IntentSender.SendIntentException e) {  
 **mGoogleApiClient**.connect();  
 }  
 }  
  
  
 @Override  
 **protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  
 **super**.onActivityResult(requestCode, resultCode, data);  
 **if** (requestCode == ***SIGN\_IN\_REQUEST\_CODE***) {  
 **mGoogleApiClient**.disconnect();  
 **mGoogleApiClient**.connect();  
 }  
 }  
  
 **private void** onSignedOut() {  
 **mSignOutButton**.setEnabled(**false**);  
 Auth.***GoogleSignInApi***.signOut(**mGoogleApiClient**).setResultCallback(  
 **new** ResultCallback<Status>() {  
 @Override  
 **public void** onResult(@NonNull Status status) {  
 **mSignInButton**.setEnabled(**true**);  
 **mNameTextView**.setText(**""**);  
 **mEmailTextView**.setText(**""**);  
 **mGoogleApiClient**.disconnect();  
 **mGoogleApiClient**.connect();  
 }  
 });  
 }  
}

**activity\_main.xml:**

*<?***xml version="1.0" encoding="utf-8"***?>*<**androidx.constraintlayout.widget.ConstraintLayout 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"  
 tools:context=".MainActivity"**>  
  
 <**com.google.android.gms.common.SignInButton  
 android:id="@+id/sign\_in\_button"  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:enabled="true"  
 android:width="150dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.352"**/>  
  
 <**Button  
 android:id="@+id/sign\_out\_button"  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:enabled="false"  
 android:text="Выйти"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.704"** />  
  
 <**TextView  
 android:id="@+id/email\_textview"  
 android:layout\_width="250dp"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.503"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.484"** />  
  
 <**TextView  
 android:id="@+id/displayname\_textview"  
 android:layout\_width="250dp"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.441"** />  
  
</**androidx.constraintlayout.widget.ConstraintLayout**>

**AndroidManifest.xml:**

*<?***xml version="1.0" encoding="utf-8"***?>*<**manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 package="com.dima.lab4"**>  
  
 <**uses-permission android:name="android.permission.INTERNET"** />  
 <**uses-permission android:name="android.permission.GET\_ACCOUNTS"** />  
 <**uses-permission android:name="android.permission.USE\_CREDENTIALS"** />  
  
  
 <**application  
 android:allowBackup="false"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme"  
 tools:ignore="GoogleAppIndexingWarning"**>  
  
 <**meta-data  
 android:name="com.google.android.gms.version"  
 android:value="@integer/google\_play\_services\_version"** />  
  
 <**activity android:name=".MainActivity"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 </**intent-filter**>  
 </**activity**>  
 </**application**>  
  
</**manifest**>

**Вывод:** в результате выполнения лабораторной работы были изучены разметки и элементы управления в *Android*, также написано и запущено приложение с элементами управления, которое может выполнить вход в Google аккаунт, вывести некоторую информацию из этого аккаунта, выйти из аккаунта.