

## Создание авторизации в AndroidStudio

*Семченко Регина Викторовна*

*Приамурский государственный университет имени Шолом-Алейхема*

*студент*

*Еровлев Павел Андреевич*

*Приамурский государственный университет имени Шолом-Алейхема*

*студент*

### **Аннотация**

Данная статья посвящена разработке авторизации в приложении Android-смартфоны в среде разработки «Android Studio» . Программирование на языке Java. Практическим результатом исследования является рабочее мобильное приложение с функцией авторизации на смартфоны с операционной системой Android.

**Ключевые слова:** информационные технологии, приложения, игры, компьютерная графика

## Create login in AndroidStudio

*Semchenko Regina Viktorovna*

*Sholom-Aleichem Priamursky State University*

*student*

*Erovlev Pavel Andreevich*

*Sholom-Aleichem Priamursky State University*

*student*

### **Abstract**

This article is devoted to login development in the Android-smartphones in the development in "Android Studio". Java programming. The practical result of the study is a working mobile application with the login function for the Android mobile platform.

**Keywords:** information technologies, applications, games, computer graphics

Любое приложение, имеющее какой-нибудь полезный функционал имеет функцию авторизации. В этом есть несколько плюсов, один из них это возможность не потерять клиента, т.к. при регистрации пользователь указывает свою почту и у разработчика будет возможность создать рассылку новостей и оповестить об этом клиента.

Цель данной статьи создать рабочее мобильное приложение с возможностью регистрации, входа и выхода из профиля на языке программирования JavaScript и среде разработки AndroidStudio.

Исследованиями в области разработки мобильных приложений занимались многие российские и зарубежные исследователи. А.С. Винокуров, Р.И. Баженов [1] рассмотрели разработку приложений для мобильных устройств. С.К. Заманова, Г.Е. Сейдахметова, Г.Г. Масимова, А.Е. Манатова[2]. Они изучили современные подходы к разработке мобильных приложений. Также они рассмотрели разработку приложения в среде Rad Studio XE7. Е.А. Зотова, М.И. Притчина [3] рассмотрели развитие программных платформ iOS и Android. И.С. Полотнянчиков, Л.А.Залогова описали выбор инструментов, провели анализ предметной области и продемонстрировали реализацию собственного мобильного приложения. Е.Н.Амиргалиев и др. разработал свою собственную модель отправки информационных сообщений для мобильных операционных систем.[4]

Приложение будет иметь 3 экрана: экран входа, экран регистрации и главный экран, после авторизации.

Для начала откроем Android Studio и создадим новый проект и заполним все необходимые данные (рис.1).

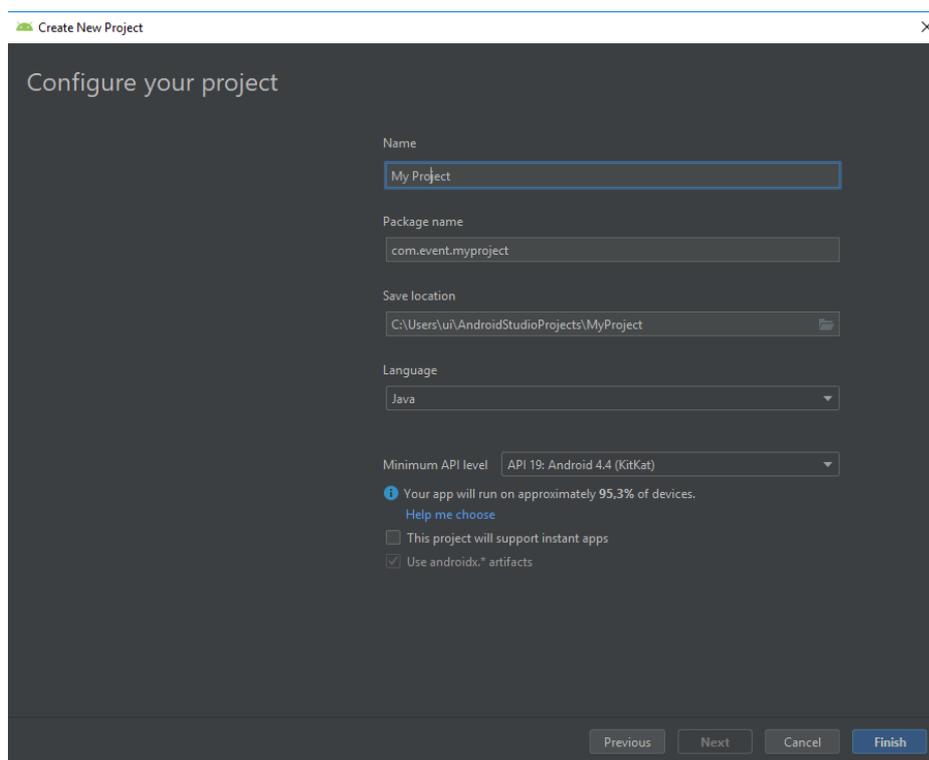


Рисунок 1 – Создание проекта

Далее создадим в папке «src» три директории: app, activity, helper.

Следующим шагом будет добавление библиотеки «volley». Откроем «build.gradle» и добавим « compile 'com.mcxiaoke.volley: library-aar: 1.0.0'»(рис.2).

```
11     versionName "1.0"
12     testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
13 }
14 buildTypes {
15     release {
16         minifyEnabled false
17         proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
18     }
19 }
20 }
21 }
22 dependencies {
23     implementation fileTree(dir: 'libs', include: ['*.jar'])
24     implementation 'androidx.appcompat:appcompat:1.1.0'
25     testImplementation 'junit:junit:4.12'
26     androidTestImplementation 'androidx.test.ext:junit:1.1.1'
27     androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
28     compile 'com.android.support:appcompat-v7:23.1.1'
29     compile 'com.android.support:design:23.1.1'
30     compile 'com.mcxiaoke.volley:library-aar:1.0.0'
31 }
32 }
```

Рисунок 2 – Добавление библиотеки

Так же необходимо добавить переменные, которые будут использоваться в приложении. Откроем «res -> values -> strings.xml» и впишем значения (рис.3).

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3
4     <string name="app_name">My Project</string>
5     <string name="hint_email">Email</string>
6     <string name="hint_password">Password</string>
7     <string name="hint_name">Fullname</string>
8     <string name="btn_login">LOGIN</string>
9     <string name="btn_register">REGISTER</string>
10    <string name="btn_link_to_register">Not a member? Sign up now.</string>
11    <string name="btn_link_to_login">Already registered! Login Me.</string>
12    <string name="welcome">Welcome</string>
13    <string name="btn_logout">LOGOUT</string>
14    <string name="name">Fullname</string>
15
16 </resources>
17
18
```

Рисунок 3 – strings.xml

Добавим значения цвета, перейдем в «res -> values -> colors.xml» (рис.4).

После внесения переменных и данных цвета создадим в директории «app» папку с названием «AppConfig.java». Впишем код реализующий соединение с сервером хранящие данные о зарегистрировавшихся и регистрируемых (рис.5).

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3
4 <color name="bg_login">#26ae90</color>
5 <color name="bg_register">#2e3237</color>
6 <color name="bg_main">#428bca</color>
7 <color name="white">#ffffff</color>
8 <color name="input_login">#222222</color>
9 <color name="input_login_hint">#999999</color>
10 <color name="input_register">#888888</color>
11 <color name="input_register_bg">#3b4148</color>
12 <color name="input_register_hint">#5e6266</color>
13 <color name="btn_login">#26ae90</color>
14 <color name="btn_login_bg">#eceeef</color>
15 <color name="lbl_name">#333333</color>
16 <color name="btn_logout_bg">#ff6861</color>
17
18 </resources>
19
20
21
```

Рисунок 4 – Colors.xml

```
1 package info.androidhive.loginandregistration.app;
2
3 public class AppConfig {
4     // Server user login url
5     public static String URL_LOGIN = "http://192.168.0.102/android_login_api/login.php";
6
7     // Server user register url
8     public static String URL_REGISTER = "http://192.168.0.102/android_login_api/register.php";
9 }
10
11
```

Рисунок 5 – AppConfig.java

Для теста будет написан локальный IP-адрес компьютера. В той же папке «app» создадим файл «AppController.java» и впишем код который будет запускаться при запуске приложения (рис.6).

```
1 package info.androidhive.loginandregistration.app;
2
3 import android.app.Application;
4 import android.text.TextUtils;
5
6 import com.android.volley.Request;
7 import com.android.volley.RequestQueue;
8 import com.android.volley.toolbox.Volley;
9
10 public class AppController extends Application {
11
12     public static final String TAG = AppController.class.getSimpleName();
13
14     private RequestQueue mRequestQueue;
15
16     private static AppController mInstance;
17
18     @Override
19     public void onCreate() {
20         super.onCreate();
21         mInstance = this;
22     }
23
24     public static synchronized AppController getInstance() {
25         return mInstance;
26     }
27
28     public RequestQueue getRequestQueue() {
29         if (mRequestQueue == null) {
30             mRequestQueue = Volley.newRequestQueue(getApplicationContext());
31         }
32
33         return mRequestQueue;
34     }
35
36     public <T> void addToRequestQueue(Request<T> req, String tag) {
37         req.setTag(TextUtils.isEmpty(tag) ? TAG : tag);
38         getRequestQueue().add(req);
39     }
40
41     public <T> void addToRequestQueue(Request<T> req) {
42         req.setTag(TAG);
43         getRequestQueue().add(req);
44     }
45
46     public void cancelPendingRequests(Object tag) {
47         if (mRequestQueue != null) {
48             mRequestQueue.cancelAll(tag);
49         }
50     }
51 }
```

Рисунок 6 - AppController.java

Приложение не будет работать без интернета, поэтому необходимо добавить разрешение на использование интернета в приложении. Переходим в «src -> res -> AndroidManifest.xml» и впишем код разрешения на использования, а так же добавим заранее экраны, которые создадим попозже, чтобы не возвращаться больше (рис.7)

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     package="info.androidhive.loginandregistration"
5     android:versionCode="1"
6     android:versionName="1.0" >
7
8     <uses-sdk
9         android:minSdkVersion="9"
10        android:targetSdkVersion="21" />
11
12    <uses-permission android:name="android.permission.INTERNET" />
13
14    <application
15        android:name="info.androidhive.loginandregistration.app.AppController"
16        android:allowBackup="true"
17        android:icon="@drawable/ic_launcher"
18        android:label="@string/app_name"
19        android:theme="@style/AppTheme"
20        tools:ignore="GoogleAppIndexingWarning">
21        <activity
22            android:name=".LoginActivity"
23            android:label="@string/app_name"
24            android:launchMode="singleTop"
25            android:windowSoftInputMode="adjustPan" >
26            <intent-filter>
27                <action android:name="android.intent.action.MAIN" />
28
29                <category android:name="android.intent.category.LAUNCHER" />
30            </intent-filter>
31        </activity>
32        <activity
33            android:name=".RegisterActivity"
34            android:label="@string/app_name"
35            android:launchMode="singleTop"
36            android:windowSoftInputMode="adjustPan" />
37        <activity
38            android:name=".MainActivity"
39            android:label="@string/app_name"
40            android:launchMode="singleTop" />
41    </application>
42
43 </manifest>
44
```

Рисунок 7 - AndroidManifest.xml

Следующая очередь папки «Helper», в нем создадим файл «SessionManager.java», в этом файле код будет поддерживать данные в приложении, а так же с помощью логического процесса «isLoggedIn» будет проверка состояния входа в систему (рис.8)

```
1 package info.androidhive.loginandregistration.helper;
2
3 import android.content.Context;
4 import android.content.SharedPreferences;
5 import android.content.SharedPreferences.Editor;
6 import android.util.Log;
7
8 public class SessionManager {
9     // LogCat tag
10    private static String TAG = SessionManager.class.getSimpleName();
11
12    // Shared Preferences
13    SharedPreferences pref;
14
15    Editor editor;
16    Context _context;
17
18    // Shared pref mode
19    int PRIVATE_MODE = 0;
20
21    // Shared preferences file name
22    private static final String PREF_NAME = "AndroidHiveLogin";
23
24    private static final String KEY_IS_LOGGEDIN = "isLoggedIn";
25
26    public SessionManager(Context context) {
27        this._context = context;
28        pref = _context.getSharedPreferences(PREF_NAME, PRIVATE_MODE);
29        editor = pref.edit();
30    }
31
32    public void setLogin(boolean isLoggedIn) {
33
34        editor.putBoolean(KEY_IS_LOGGEDIN, isLoggedIn);
35
36        // commit changes
37        editor.commit();
38
39        Log.d(TAG, "User login session modified!");
40    }
41
42    public boolean isLoggedIn(){
43        return pref.getBoolean(KEY_IS_LOGGEDIN, false);
44    }
45 }
```

Рисунок 8 - SessionManager.java

Так же в этой папке создадим файл «SQLiteHandler.java» и напишем код, который будет заботиться о хранении данных пользователя в базе данных SQLite (рис.9). Так же, каждый раз, когда необходимо будет получить зарегистрированную пользовательскую информацию, то будет процесс извлечения данных из SQLitea, а не отправка запроса и ожидание ответа.

```
1 package info.androidhive.loginandregistration.helper;
2
3 import android.content.ContentValues;
4 import android.content.Context;
5 import android.database.Cursor;
6 import android.database.sqlite.SQLiteDatabase;
7 import android.database.sqlite.SQLiteOpenHelper;
8 import android.util.Log;
9
10 import java.util.HashMap;
11
12 public class SQLiteHandler extends SQLiteOpenHelper {
13
14     private static final String TAG = SQLiteHandler.class.getSimpleName();
15
16
17     private static final int DATABASE_VERSION = 1;
18
19
20     private static final String DATABASE_NAME = "android_api";
21
22
23     private static final String TABLE_USER = "user";
24
25
26     private static final String KEY_ID = "id";
27     private static final String KEY_NAME = "name";
28     private static final String KEY_EMAIL = "email";
29     private static final String KEY_UID = "uid";
30     private static final String KEY_CREATED_AT = "created_at";
31
32     public SQLiteHandler(Context context) {
33         super(context, DATABASE_NAME, null, DATABASE_VERSION);
34     }
35
36
37     @Override
38     public void onCreate(SQLiteDatabase db) {
39         String CREATE_LOGIN_TABLE = "CREATE TABLE " + TABLE_USER + "("
40             + KEY_ID + " INTEGER PRIMARY KEY," + KEY_NAME + " TEXT,"
41             + KEY_EMAIL + " TEXT UNIQUE," + KEY_UID + " TEXT,"
42             + KEY_CREATED_AT + " TEXT" + ")";
43         db.execSQL(CREATE_LOGIN_TABLE);
44
45         Log.d(TAG, "Database tables created");
46     }
```

Рисунок 9 - SQLiteHandler.java

Следующий шаг – это добавление экрана входа. Создадим модуль входа в систему и добавим код для запроса входа на сервере.

В папке «res -> layout» создадим файл «activity\_login.xml» Это будет экран входа в систему. Размеры и цвет выбираем по вкусу (рис.10)

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="fill_parent"
4     android:layout_height="fill_parent"
5     android:background="@color/bg_login"
6     android:gravity="center"
7     android:orientation="vertical"
8     android:padding="10dp" >
9
10 <LinearLayout
11     android:layout_width="fill_parent"
12     android:layout_height="wrap_content"
13     android:layout_gravity="center"
14     android:orientation="vertical"
15     android:paddingLeft="20dp"
16     android:paddingRight="20dp" >
17
18 <EditText
19     android:id="@+id/email"
20     android:layout_width="fill_parent"
21     android:layout_height="wrap_content"
22     android:layout_marginBottom="10dp"
23     android:background="@color/white"
24     android:hint="@string/hint_email"
25     android:inputType="textEmailAddress"
26     android:padding="10dp"
27     android:singleLine="true"
28     android:textColor="@color/input_login"
29     android:textColorHint="@color/input_login_hint" />
30
31 <EditText
32     android:id="@+id/password"
33     android:layout_width="fill_parent"
34     android:layout_height="wrap_content"
35     android:layout_marginBottom="10dp"
36     android:background="@color/white"
37     android:hint="@string/hint_password"
38     android:inputType="textPassword"
39     android:padding="10dp"
40     android:singleLine="true"
41     android:textColor="@color/input_login"
42     android:textColorHint="@color/input_login_hint" />
43
```

Рисунок 10 - activity\_login.xml

Создадим файл с именем «LoginActivity.java» в директории «Activity» . В коде параметр «checkLogin ()» проверяет данные для входа на сервер, выполняя http – запрос (рис.11).

```
29
30 public class LoginActivity extends Activity {
31     private static final String TAG = RegisterActivity.class.getSimpleName();
32     private Button btnLogin;
33     private Button btnLinkToRegister;
34     private EditText inputEmail;
35     private EditText inputPassword;
36     private ProgressDialog pDialog;
37     private SessionManager session;
38     private SQLiteDatabase db;
39
40     @Override
41     public void onCreate(Bundle savedInstanceState) {
42         super.onCreate(savedInstanceState);
43         setContentView(R.layout.activity_login);
44
45         inputEmail = (EditText) findViewById(R.id.email);
46         inputPassword = (EditText) findViewById(R.id.password);
47         btnLogin = (Button) findViewById(R.id.btnLogin);
48         btnLinkToRegister = (Button) findViewById(R.id.btnLinkToRegisterScreen);
49
50
51         pDialog = new ProgressDialog(this);
52         pDialog.setCancelable(false);
53
54
55         db = new SQLiteDatabase(getApplicationContext());
56
57
58         session = new SessionManager(getApplicationContext());
59
60
61         if (session.isLoggedIn()) {
62             Intent intent = new Intent(LoginActivity.this, MainActivity.class);
63             startActivity(intent);
64             finish();
65         }
66     }
67
68     return pref.getBoolean(KEY_IS_LOGGEDIN, false);
69 }
70
71 }
```

Рисунок 11 - LoginActivity.java

Теперь, при запуске приложения, будет виден экран входа в систему. Но вход в систему может не работать, так как нет никакой пользовательской информации в базе данных mysql. Это можно сделать, добавив экран регистрации (рис.12).

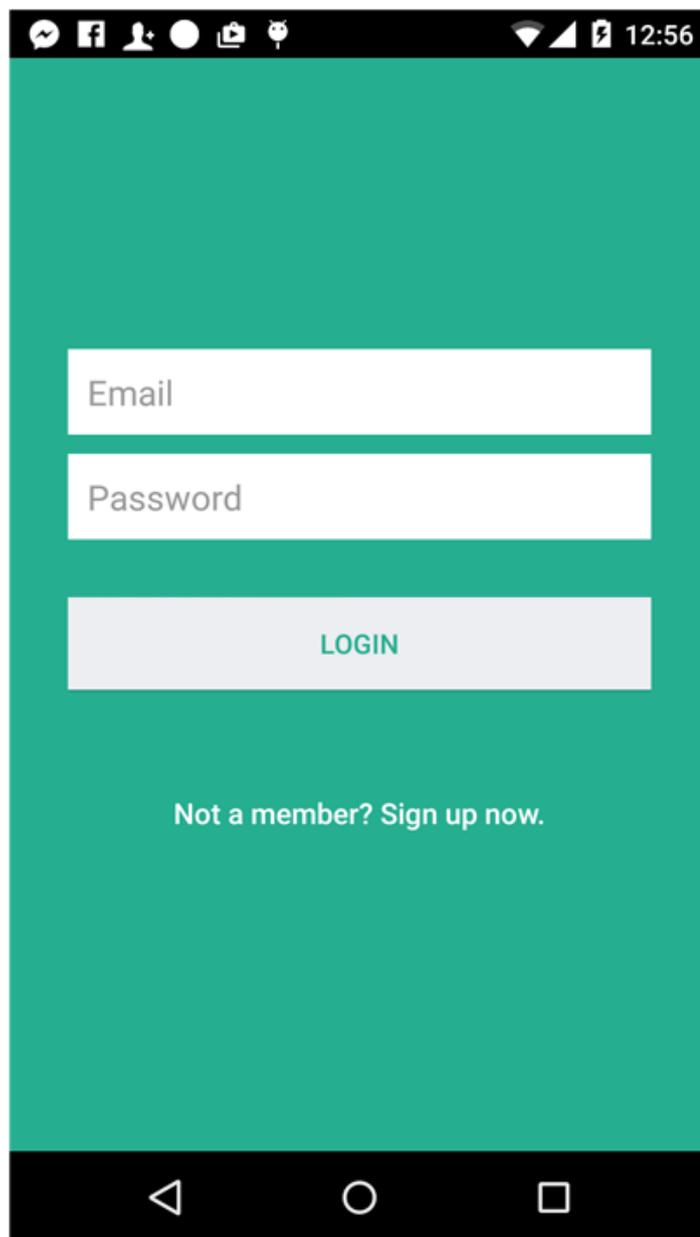


Рисунок 12 – Экран авторизации

Осталось создать два экрана и следующим будет экран регистрации  
Переходим в директорию «layout»и создаем документ «activity\_register.xml» и прописываем код, который будет отображать информацию, текст, кнопки(рис.13).

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="fill_parent"
4     android:layout_height="fill_parent"
5     android:background="@color/bg_register"
6     android:gravity="center"
7     android:orientation="vertical"
8     android:padding="10dp" >
9
10 <LinearLayout
11     android:layout_width="fill_parent"
12     android:layout_height="wrap_content"
13     android:layout_gravity="center"
14     android:orientation="vertical"
15     android:paddingLeft="20dp"
16     android:paddingRight="20dp" >
17
18 <EditText
19     android:id="@+id/name"
20     android:layout_width="fill_parent"
21     android:layout_height="wrap_content"
22     android:layout_marginBottom="10dp"
23     android:background="@color/input_register_bg"
24     android:hint="@string/hint_name"
25     android:padding="10dp"
26     android:singleLine="true"
27     android:inputType="textCapWords"
28     android:textColor="@color/input_register"
29     android:textColorHint="@color/input_register_hint" />
30
31 <EditText
32     android:id="@+id/email"
33     android:layout_width="fill_parent"
34     android:layout_height="wrap_content"
35     android:layout_marginBottom="10dp"
36     android:background="@color/input_register_bg"
37     android:hint="@string/hint_email"
38     android:inputType="textEmailAddress"
39     android:padding="10dp"
40     android:singleLine="true"
41     android:textColor="@color/input_register"
42     android:textColorHint="@color/input_register_hint" />
43
```

Рисунок 13 – activity\_register.xml

Теперь в директории «activity» создадим файл «RegisterActivity.java» с кодом регистрации пользователя на сервере (рис.14). «registerUser ()» будет сохранять пользователя, передавая имя, адрес электронной почты и пароль на сервер. «db.addUser ()» - добавит пользователя в базу данных SQLite после его успешной регистрации.

```
24     import info.androidhive.loginandregistration.R;
25     import info.androidhive.loginandregistration.app.AppConfig;
26     import info.androidhive.loginandregistration.app.AppController;
27     import info.androidhive.loginandregistration.helper.SQLiteHandler;
28     import info.androidhive.loginandregistration.helper.SessionManager;
29
30     public class RegisterActivity extends Activity {
31         private static final String TAG = RegisterActivity.class.getSimpleName();
32         private Button btnRegister;
33         private Button btnLinkToLogin;
34         private EditText inputFullName;
35         private EditText inputEmail;
36         private EditText inputPassword;
37         private ProgressDialog pDialog;
38         private SessionManager session;
39         private SQLiteHandler db;
40
41         @Override
42         public void onCreate(Bundle savedInstanceState) {
43             super.onCreate(savedInstanceState);
44             setContentView(R.layout.activity_register);
45
46             inputFullName = (EditText) findViewById(R.id.name);
47             inputEmail = (EditText) findViewById(R.id.email);
48             inputPassword = (EditText) findViewById(R.id.password);
49             btnRegister = (Button) findViewById(R.id.btnRegister);
50             btnLinkToLogin = (Button) findViewById(R.id.btnLinkToLoginScreen);
51
52             pDialog = new ProgressDialog(context);
53             pDialog.setCancelable(false);
54
55             session = new SessionManager(getApplicationContext());
56
57             db = new SQLiteHandler(getApplicationContext());
58
59             if (session.isLoggedIn()) {
60
61                 Intent intent = new Intent(packageContext, RegisterActivity.this,
62                     MainActivity.class);
63                 startActivity(intent);
64                 finish();
65             }
```

Рисунок 14 - RegisterActivity.java

Экран регистрации готов, остался главный экран куда пользователь будет переходить после входа или регистрации. Сделаем экран регистрации в виде отображения вошедшему пользователю его данные.

В «layout» создадим файл «activity\_main.xml» и пропишем код, это будет отображение экрана (рис.15).

```
1 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   tools:context="{relativePackage}.{activityClass}" >
6
7   <LinearLayout
8     android:layout_width="fill_parent"
9     android:layout_height="wrap_content"
10    android:layout_centerInParent="true"
11    android:layout_marginLeft="20dp"
12    android:layout_marginRight="20dp"
13    android:gravity="center"
14    android:orientation="vertical" >
15
16     <TextView
17       android:layout_width="wrap_content"
18       android:layout_height="wrap_content"
19       android:text="@string/welcome"
20       android:textSize="20dp" />
21
22     <TextView
23       android:id="@+id/name"
24       android:layout_width="wrap_content"
25       android:layout_height="wrap_content"
26       android:padding="10dp"
27       android:textColor="@color/lbl_name"
28       android:textSize="24dp" />
29
30     <TextView
31       android:id="@+id/email"
32       android:layout_width="wrap_content"
33       android:layout_height="wrap_content"
34       android:textSize="13dp" />
35
36     <Button
37       android:id="@+id/btnLogout"
38       android:layout_width="fill_parent"
39       android:layout_height="wrap_content"
40       android:layout_marginTop="40dip"
41       android:background="@color/btn_logout_bg"
42       android:text="@string/btn_logout"
43       android:textAllCaps="false"
44       android:textColor="@color/white"
45       android:textSize="15dp" />
46   </LinearLayout>
47
48 </RelativeLayout>
```

Рисунок 15 - activity\_main.xml

Здесь просто извлекаем информацию о зарегистрированном пользователе из SQLite и отображаем ее на экране. Кнопка выхода из системы позволяет выйти из системы, очистив сеанс и удалив пользователя из таблицы SQLite(рис.16).

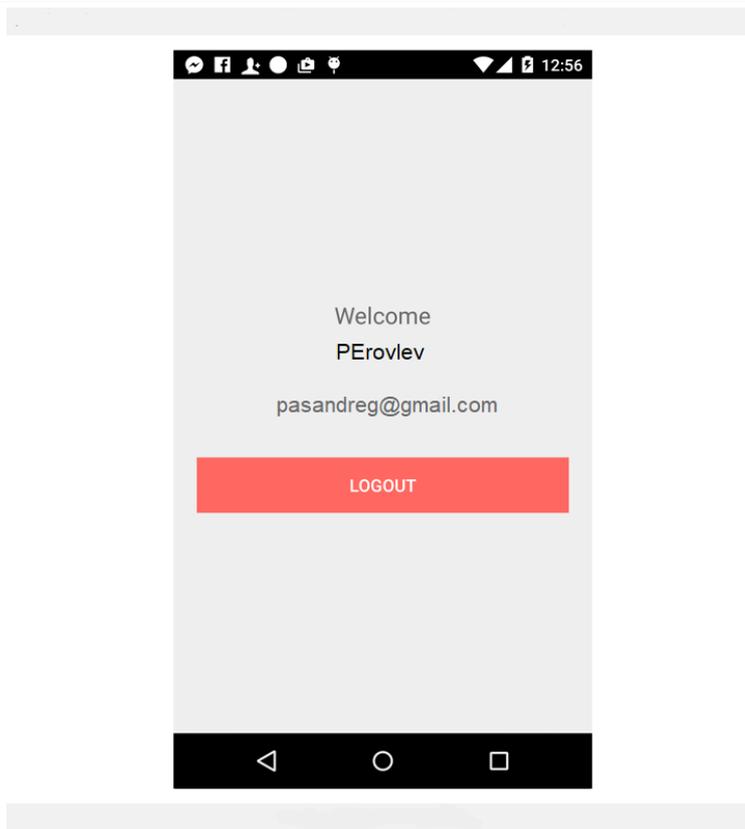


Рисунок 16 – Главный экран

Теперь, если запустить приложение, отобразится экран после успешного входа в систему.

Как видно, реализация авторизации в приложении не так уж и трудно, но дает свои плюсы в то, что разработчик не потеряет клиента.

Подводя итоги можно сказать, что было реализована авторизация в мобильном приложении и успешно протестирована.

### **Библиографический список**

1. Винокуров А.С., Баженов Р.И. Разработка мобильного приложения информационного сайта для абитуриентов и первокурсников университета // Современные научные исследования и инновации. 2015. № 7-2 (51). С. 54-62.
2. Заманова С.К., Сейдахметова Г.Е., Масимова Г.Г., Манатова А.Е. Разработка мобильного приложения в среде Rad Studio XE7 // Труды Международного симпозиума «Надежность и качество». 2015. №1. С. 237-240.
3. Amirgaliyev E.N., Kalizhanova A.U., Kozbakova A.KH. Development of applications to mobile devices in Android platform // Труды Международного симпозиума «Надежность и качество». 2015. №1. С. 240-242.