

Воробьев Илья Романович

студент 4 курса

ПГУТИ

Россия, г. Самара

Морозов Дмитрий Денисович

студент 4 курса

ПГУТИ

Россия, г. Самара

Лиманова Наталия Игоревна

к.т.н., зав. каф. «Информационные системы и технологии»

ПГУТИ

Россия, г. Самара

АВТОРИЗАЦИЯ НА МОБИЛЬНЫХ ПРИЛОЖЕНИЯХ С ИСПОЛЬЗОВАНИЕМ БАЗЫ ДАННЫХ

Аннотация: В статье подробно рассмотрен процесс реализации авторизации в мобильном приложении. Он заключается во взаимодействии с самой актуальной на данный момент базой данных *Firestore*. Сам процесс реализации выполнен на языке *Java*, так как этот язык доминирует в сфере создания мобильных приложений и является самым удобным. Данная статья предназначена для пользователей, которые имеют базовые знания в создании мобильных приложений.

Ключевые слова: Авторизация, мобильное приложение, база данных, активити.

Annotation: The article describes in detail the process of implementing authorization in a mobile application. It consists in interacting with the *Firestore*

database, as it is the most up-to-date at the moment. The implementation process itself will be in Java, since this language dominates in this area and is the most convenient. We will use the simplest way to create authorization, using 2 classes, in one to receive data, and in the other to send. This article is intended for users who have basic knowledge in creating mobile applications.

Keywords: *Development, element, list, file, activity.*

Введение

Мобильная разработка – это актуальное направление в настоящее время, так как все сферы человеческой деятельности так или иначе связаны с мобильными приложениями. Можно привести в пример корпорации Yandex, Google, Ozon, и других разработчиков, которые имеют свои мобильные приложения.

В статье рассматривается способ подключения базы данных Firebase к созданному проекту на Android Studio, в котором рассматривается способ создания кода для авторизации в приложении. На рисунке 1 приведена форма регистрации в мобильном приложении.

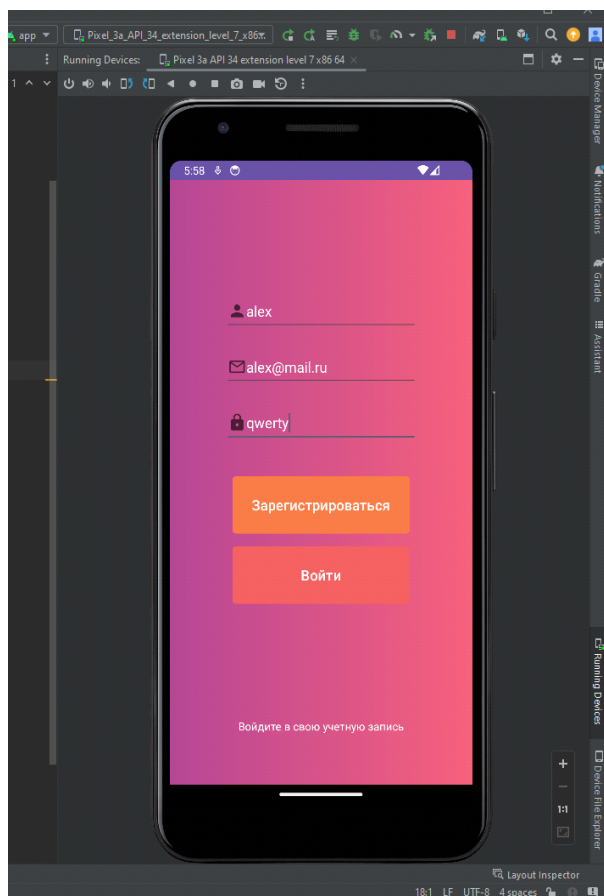


Рис. 1 – Первое активити «Регистрация»

Как видно из рисунка 2, в базу данных добавилась запись после нажатия кнопки “Зарегистрироваться”.

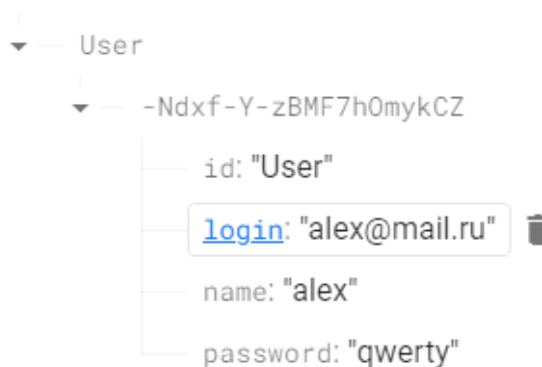


Рис. 2 – База данных Firebase

Рассмотрим способ, при котором создаются три приватных переменных типа «EditText». Приватная переменная типа «DatabaseReference» создается, чтобы получить доступ к базе данных. Эта

переменная вмещает в себя ссылку на базу данных Firebase. При этом создается группа под названием «User», чтобы в базе данных заполнить значения. Метод onCreate нужен для создания первого активити. Функция setContentView нужна для отображения layout (ресурса) на активити. Далее рассматривается вызов метода init. Для переменных edName, edLogin, edPassword присваивается id, чтобы работать с ними. Передаем инстанцию нашей базы данных в переменную mDataBase. Через функцию getReference указывается группа, в которую будут передаваться наши данные.

С помощью метода Registrasia записываются данные в базу данных по нажатию кнопки «Зарегистрироваться». Id получаем от базы данных с помощью функции getKey, далее создаем три переменных, в которые запишем текст, полученный из переменных edName, edLogin, edPassword с помощью функции getText(), и присвоим данным тип string.

На рисунке 3 приведено второе окно приложения.

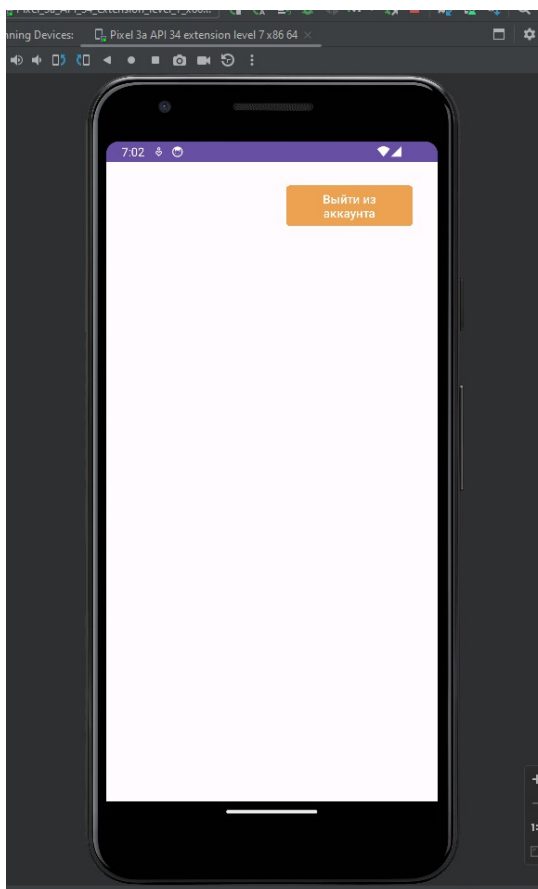


Рис. 3 – Второе активити

Создается класс User, который будет хранить данные из EditText, которые мы вводим в форме регистрации. На следующем шаге создается объект класса new User() в классе MainActivity, в который передаются все элементы из формы регистрации, далее этот объект передает данные в класс User. Класс User будет записываться в созданную базу данных, поэтому передаем данные в него.

Создаем объект класса intent, который позволяет перейти из одного окна в другой. Внутри объекта указываем окно из которого будем переходить (функция «this») в нужное нам окно, в нашем случае MainActivity2.

Метод startActivity(intent) - запускает жизненный цикл активити, в него передается переменная, в которой хранится переход между окнами.

На рисунке 4 приведен код реализации первого окна.

```
public class MainActivity extends AppCompatActivity {  
  
    2 usages  
    private EditText edName,edLogin,edPassword;  
    3 usages  
    private DatabaseReference mDataBase;  
    1 usage  
    private String USER_KEY = "User";  
  
    @Override  
    3 protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        init();  
    }  
  
    1 usage  
    private void init(){  
        edName = findViewById(R.id.edname);  
        edLogin = findViewById(R.id.edlogin);  
        edPassword = findViewById(R.id.edpassword);  
        mDataBase = FirebaseDatabase.getInstance().getReference(USER_KEY);  
    }  
  
    1 usage  
    public void Registrasia(View view){  
        String id = mDataBase.getKey();  
        String name = edName.getText().toString();  
        String login = edLogin.getText().toString();  
        String password = edPassword.getText().toString();  
        User newUser = new User(id,name,login,password);  
        mDataBase.push().setValue(newUser);  
        Intent intent = new Intent( packageContext: this, MainActivity2.class);  
        startActivity(intent);  
    }  
}
```

Рис. 4 – Код первого активити

На рисунке 5 проиллюстрирован код создания класса User.

```
public class User {
    1 usage
    public String id,name,login,password;

    public User() {
    }

    1 usage
    public User(String id, String name, String login, String password) {
        this.id = id;
        this.name = name;
        this.login = login;
        this.password = password;
    }
}
```

Рис. 5 – Код класса User

В классе User сохраняются данные, для одновременной записи информации в каждую переменную, чтобы записывать их сразу все, а не по одному в базу данных.

Создаем 4 публичных переменных id, name, login, password.

Далее создается конструктор этого класса, для этого внедряем пустой класс, и класс со всеми значениями выше.

Класс User запрашивает значения id, name, login, password из основного класса MainActivity и сохраняет эти значения.

Таким образом выполняется авторизация пользователя в мобильном приложении.

Заключение

Материал был подготовлен для начинающих создателей мобильных приложений, для того чтобы они могли опираться на материал статьи и создавать свои первые проекты. Статья в полном объеме раскрывает

основные этапы разработки авторизации на мобильных приложениях с использованием базы данных.

Рассмотрен наиболее предпочтительный способ подключения базы данных Firebase к созданному проекту на Android Studio. Рассмотрен способ создания кода для авторизации в приложении.

Список источников

1. Федотенко М.А., Разработка мобильных приложений. Первые шаги. -2019.
2. Колисниченко Д.Н., Программирование для Android. -2020.
3. Черников В., Разработка мобильных приложений на Java для iOS и Android.-2020.
4. Хортон Д., Разработка Android-приложений с нуля. 3-е изд.-2023.