УДК 621.3.049.77–048.24:537.2

ИССЛЕДОВАНИЕ ПРЕИМУЩЕСТВ И НЕДОСТАТКОВ ИСПОЛЬЗОВАНИЯ FIREBASE В ПРИЛОЖЕНИЯХ ПОД ОПЕРАЦИОННУЮ СИСТЕМУ ANDROID

Карачун М.Д.

Белорусский государственный университет информатики и радиоэлектроники, г. Минск, Республика Беларусь

Научный руководитель: Проходский Д.В. – магистр техники и технологии, ассистент кафедры ПИКС

Аннотация. Экспериментально исследована платформа Firebase путем подключения ее к программному средству в среде разработки Android Studio, проверены такие сценарии использования, как база данных реального времени, аутентификация пользователей и облачное хранилище. Выявлены основные преимущества и недостатки хранения данных в реальном времени, работы на платформе Google и архитектурного решения без использования серверов.

Ключевые слова: Firebase, база данных реального времени, облачное хранилище, бессерверная платформа, NoSQL

Введение. Хранение данных в разрабатываемых проектах под операционную систему *Android* обеспечивают базы данных (БД). Выделяют два основных типа БД: реляционные и нереляционные. *Firebase* является комплексным набором инструментов и услуг, предлагаемых в качестве платформы *Backend-as-a-Service* (*BaaS*). Он содержит два облачных решения для баз данных, доступных для клиентов, которые поддерживают синхронизацию данных в реальном времени: *Cloud Firestore* и *Firebase Realtime Database* [1].

База данных реального времени – это база данных, размещенная в облаке. Данные хранятся в формате *JSON* и синхронизируются в реальном времени для каждого подключенного клиента. Это значит, что каждое изменение будет автоматически обновлять подключенных клиентов.

Основная часть. Данные в режиме реального времени, что означает, что каждое изменение будет автоматически обновлять подключенных клиентов. *Firebase* использует документно-ориентированную модель данных *NoSQL*. Это означает, что преимущества нереляционных БД также присущи данной базе данных реального времени, а именно: *NoSQL* может обеспечить модель данных, лучше удовлетворяющую потребности приложения, упростив тем самым это взаимодействие и уменьшив количество кода, который необходимо написать, отладить и развить, а также они лучше вписываются в сценарии обработки больших объемов данных, поскольку многие разработаны специально для кластеров [2].

Одним из преимуществ является наличие бесплатного начального плана, который будет особенно полезен для студентов в процессе обучения и для ознакомления с базами данных данного типа. Однако он ограничен 50 подключениями и 1 ГБ хранилища. При увеличении требований к разработке можно выбрать платный тариф, преимуществом которого является учитывание бесплатных лимитов.

Еще одним положительным моментом является то, что *Firebase* предоставляет разработчикам весь спектр продуктов, которые могут понадобиться им в процессе разработки. Помимо двух вариантов баз данных, он позволяет легко и просто выполнять облачное хранилище мультимедиа и обеспечивает разработку приложений без использования сервера с помощью интегрированных облачных решений.

Для исследования было создано приложение, которое взаимодействует с *Firebase*. В созданном приложений необходимо знать личность пользователя, для того чтобы безопасно сохранять пользовательские данные в облаке и обеспечивать одинаковый персонализированный опыт на всех устройствах пользователя.

Для этой цели в *Firebase* есть система *Firebase Authentication*, которая предоставляет серверные службы, простые в использовании *SDK* и готовые библиотеки пользовательского интерфейса для аутентификации пользователей в приложении. Для аутентификации пользователя используется метод *createUserWithEmailAndPassword* (рисунок 1).

```
mAuth = FirebaseAuth.getInstance();
registerBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        final String firstnameTxt = firstname.getText().toString();
        final String emailTxt = email.getText().toString();
        final String passwordTxt = password.getText().toString();
        final String confirmPasswordTxt = confirmPassword.getText().toString();

        if (firstnameTxt.isEmpty() || emailTxt.isEmpty() || passwordTxt.isEmpty() || confirmPasswordTxt.isEmpty()) {
            Toast.makeText( context: Register.this,  text: "Пожалуйста, заполните все поля", Toast.LENGTH_SHORT).show();
        }

        else if (!passwordTxt.equals(confirmPasswordTxt)) {
            Toast.makeText( context: Register.this,  text: "Пароли не совпадают", Toast.LENGTH_SHORT).show();
              confirmPassword.requestFocus();
        } else {

            mAuth.createUserWithEmailAndPassword(emailTxt, passwordTxt).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if (task.isSuccessful()){
                        Toast.makeText( context: Register.this,  text: "User registered successfully", Toast.LENGTH_SHORT).show();
                        finish();
                    }else{
                        Toast.makeText( context: Register.this,  text: "Registration Error: " + task.getException().getMessage(), Toast.LENGTH_SHORT).show();
                    }
                }
```

Рисунок 1 – Пример кода для создания пользователя в Android Studio

Сервис *Firebase Authentication* поддерживает аутентификацию с использованием паролей, телефонных номеров, популярных поставщиков федеративных удостоверений, таких как *Google*, *Facebook* и *Twitter* [3]. Пользователи *Firebase* имеют фиксированный набор основных свойств – уникальный идентификатор, основной адрес электронной почты, имя и *URL*-адрес фотографии – которые хранятся в пользовательской базе данных проекта и могут быть обновлены пользователем (рисунок 2).

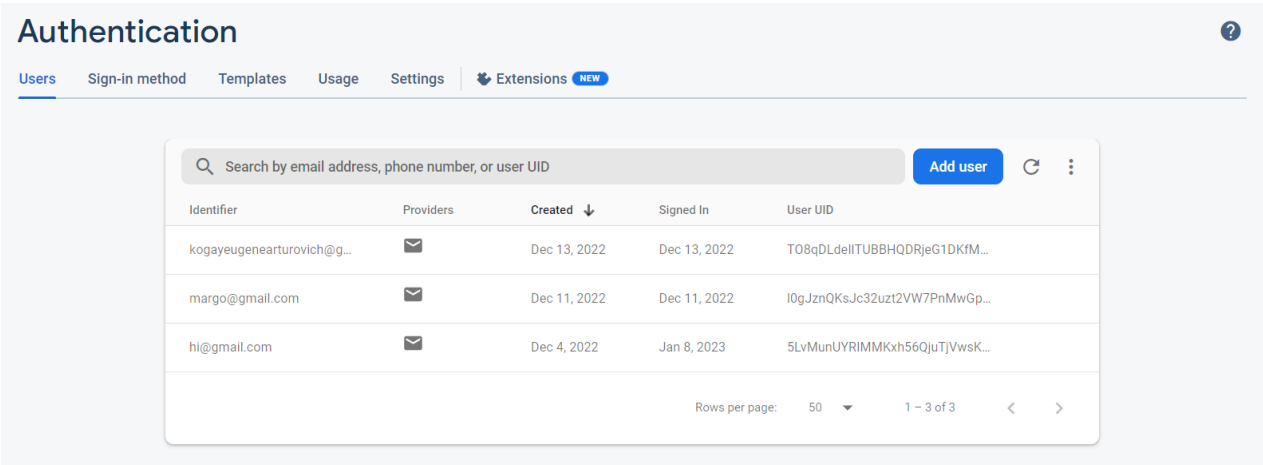

Рисунок 2 – Пример созданных пользователей сервиса Firebase Authentication

Firebase предоставляет услугу облачного хранения, которая позволяет разработчикам хранить и извлекать двоичные файлы, такие как изображения и видео, что делает его идеальным для создания приложений, требующих хранения файлов.

Масштабирование кластера баз данных является очень сложной задачей, а оптимизация производительности для обеспечения бесперебойной работы при огромных рабочих нагрузках требует наличие опытных инженеров.

Firebase решает эту проблему, поскольку поставляется с архитектурой, в которой нет серверов и в которой оплата производится на основе запросов, в которой нет необходимости управлять инфраструктурой серверов и даже беспокоиться о ней.

Ещё одним достоинством является оптимальная безопасность и доступность данных за счет осуществления регулярного резервного копирования. Приложения защищены от любой возможности потери данных благодаря использованию функции автоматического резервного копирования, которая есть на этой платформе.

Помимо достоинств в *Firebase* присутствуют и существенные недостатки. Эта платформа работает на поддомене *Google* и ее официальный сайт- и он заблокирован во многих странах. Оба варианта базы данных *Firebase*, *Firestore* и *Firebase Realtime Database*, предлагают *NoSQL*, и нет никакой возможности использовать реляционную базу данных. Несмотря на то, что *Firebase Realtime Database* поддерживает транзакции, разработчикам приходится реализовывать свой собственный код, в отличие от использования стандартной реляционной базы данных, что приведет к усложнению системы. Отсутствие выделенных серверов обеспечивает меньшую гибкость, в отличии от полностью выделенной кластерной структуры.

Заключение. Исследованы основные преимущества и недостатки Firebase платформы. Изучены особенности нереляционных баз данных и их применение в *Firebase Database*. Создано *Android* приложение с возможностью добавить пользователя в базу данных с использованием *Firebase Authentication*. Изучена услуга облачного хранения и способ обеспечения безопасности. Установлено, что отсутствие выделенных серверов является одновременно и преимуществом, и недостатком.

Список литературы

1. *Laurence Moroney. The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform / Laurence Moroney. – Seattle, Washington, USA, 2017. – Pp. 1–73. – ISBN-13 (pbk): 978-1-4842-2942-0032*

2 Садаладж, Фаулер: NoSQL. Методология разработки нереляционных баз данных/ *Садаладж Прамодкумар Дж., Фаулер Мартин // Диалектика. – 2020. – Vol. 1, N 14. – Pp. 14–33.*

3. *Firebase Authentication // Firebase. – [Электронный ресурс]. – Режим доступа: https://firebase.google.com/docs/auth. – Дата доступа: 20.03.2023.*

UDC 621.3.049.77–048.24:537.2

CO RESEARCH OF ADVANTAGES AND DISADVANTAGES OF USING FIREBASE IN APPLICATIONS FOR THE ANDROID OPERATING SYSTEM

Karachun M.D.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Prakhodski D.V. – master of engineering and technology, assistant of the Department of ICSD

Annotation. The Firebase platform was experimentally investigated by connecting it to a software tool in the Android Studio development environment, such usage scenarios as a real-time database, user authentication and cloud storage were tested. The main advantages and disadvantages of real-time data storage, working on the Google platform and an architectural solution without using servers are revealed.

Keywords: Firebase, real-time database, cloud storage, serverless platform, NoSQL