WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Simple Android Application with Firebase Integration

[1]Vivekanand Thakare, [2]Sarveshkumar Nasare, [3]Saloni Meshram, [4]Dipali Wadatkar, [5]Usha Mohadikar*

[1]Asst. Professor, [2,3,4,5]UG Students
[1,2,3,4,5]Department of Computer Science & Engineering
[1,2,3,4,5]Govindrao Wanjari College of Engineering and Technology Nagpur, India
nasaresk@gmail.com

ABSTRACT

The websites are heavily dependent on the large amount of database and unorganized data such as videos, images, audio, text, files and other arbitrary file types. Relational Database Management System (RDBMS) finds it hard to handle the unstructured data. On the other hand, Firebase is a newer technology for handling such a large amount of unorganized data as compared to RDBMS. It is also faster than RDBMS, since it uses JavaScript parser in background. This paper works around the application of Firebase with Android platform and aims to understand the related concepts, advantages and respective limitations. The paper demonstrates some of the features of Firebase by developing an android application connected using same Firebase instance.

Keywords: *Android, Firebase, java*

1. Introduction

Oracle SQL, Microsoft SQL Server and MySQL are used for android apps which are connected to the server using PHP files. The Firebase ecosystem stores data in JSON format. The other server stores data in the form of rows and columns of a table. The Firebase is NoSQL based. There are other similar services to firebase, such as:

- AWS Mobile Hub
- CloudKit
- Parse Server

1.1 Firebase

Firebase is Google’s app development platform. It helps developers to build high-quality apps and also stores the data in JSON(JavaScript Object Notation) format which doesn’t use query for read or write operation performed on it. It is the backend of a system that also provides a database for storing data.

The services covered in this paper are:

Firebase Auth: Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made libraries for user-interfaces to authenticate users to the app. It supports authentication using phone numbers, passwords, popular federated identity providers like Google, Facebook and Twitter, and more.

Real-time Database: Firebase provides services like backend and real-time database. An API is provided to the application developer which allows synchronization of application data across clients and stored on Firebase's cloud. The client libraries are provided by the company which enables integration with Android, IOS, and other types of applications.

1.2 Android App

Android is a smart-phone operating system [12],[13] which is modified Linux kernel, designed primarily for touch screen mobile devices like Tablets and Smartphone. Applications ("apps”), which extend the functionality of devices, are written using the Android software development kit (SDK) and, often, XML Java/Kotlin programming language. The SDK includes a comprehensive set of development tools, including a debugger, software libraries, a handset emulator based on QEMU(Quick EMUlator), sample code, documentation and tutorials. Other development tools are also available which includes a native development kit (NDK) for the apps. The above listed development tools can be used with an Android Studio IDE to make an Android app.

2. Illustrations

Walter Kriha [1] in his article has mention the overview of NoSQL database and common concepts, techniques and patterns as well as several classes of NoSQL databases (key/value-stores, document databases, columnoriented databases) and individual products. Various advantages and limitations of using NoSQL database has been discussed. Supriya S. Pore, Swalaya B. Pawari [2] conducted a comparative study between SQL and NoSQL. The study focuses on databases like SQL and NoSQL. The study concludes that ACID property is not used in the NoSQL databases due to data inconsistency. Vatika Sharma, Meenu Dave [3] overviewed NoSQL databases focusing on how it has lowered the dominance of SQL with its background and characteristics. Daniel Pan [4] has demonstrated connecting firebase with an Android app and basics of designing the structure of real-time database of Firebase. Landon Cox [5] has shown the comparative study between SQLite and firebase. It also focuses on organizing data as a JSON tree in order to store in Firebase.

3. STEPS TO ADD FIREBASE TO ANDROID

Firebase can be added with minimum of Android 2.3 (Gingerbread) and at least have Google Play services version 9.6.1. The steps below are to be followed up next: A project needs to be created in firebase console. Enter project name and the location. Project name under firebase console may be anything and also may different from the application name given. Click on “Add Firebase to the Android app” and follow steps mentioned along there. The user will be asked for the package name and optional SHA-1 certificate of android studio project. At the end “google-services.json” file will be downloaded after the successful completion of above steps. The file can be downloaded again at any time from the Firebase console. The downloaded file has to be placed into the project’s app or module folder. Firebase is added to the project.

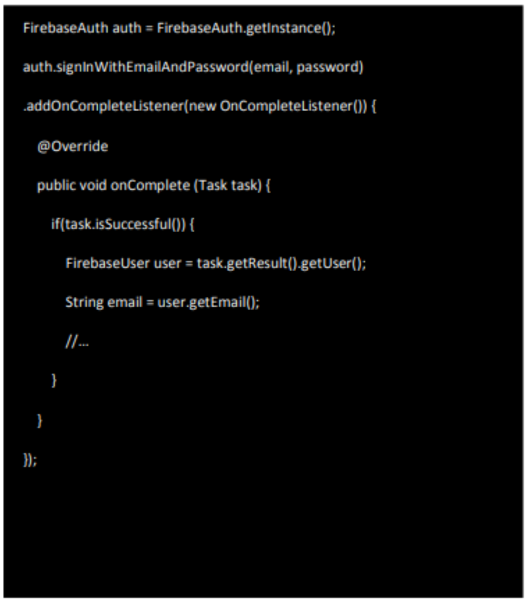
4. USING FIREBASE FEATURES IN ANDROID APPLICATION

All authors are required to complete the Procedia exclusive license transfer agreement before the article can be published, which they can do online. This transfer agreement enables Elsevier to protect the copyrighted material for the authors, but does not relinquish the authors’ proprietary rights. The copyright transfer covers the exclusive rights to reproduce and distribute the article, including reprints, photographic reproductions, microfilm or any other reproductions of similar nature and translations. Authors are responsible for obtaining from the copyright holder, the permission to reproduce any figures for which copyright exists.

Firebase can be integrated with the android applications with a few lines of code. The features like authentication and realtime database have been discussed in section. The methodology to use some features are as:

4.1 Authentication:

After adding firebase and authentication dependency to the Android application, the login id of the user can be created by the following Java code [6]:



```
FirebaseAuth auth = FirebaseAuth.getInstance();

auth.signInWithEmailAndPassword(email, password)

.addOnCompleteListener(new OnCompleteListener() {

  @Override

  public void onComplete (Task task) {

    if(task.isSuccessful()) {

      FirebaseUser user = task.getResult().getUser();

      String email = user.getEmail();

      //...

    }

  }

});
```

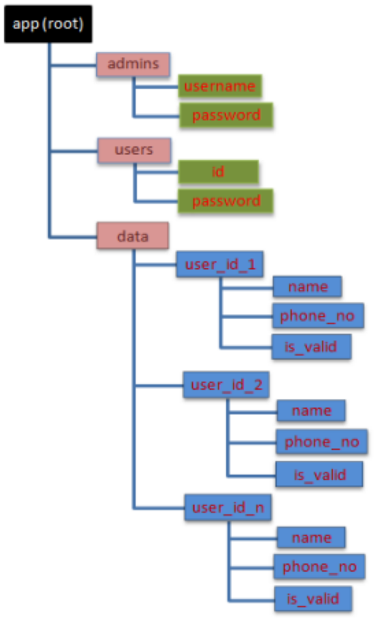
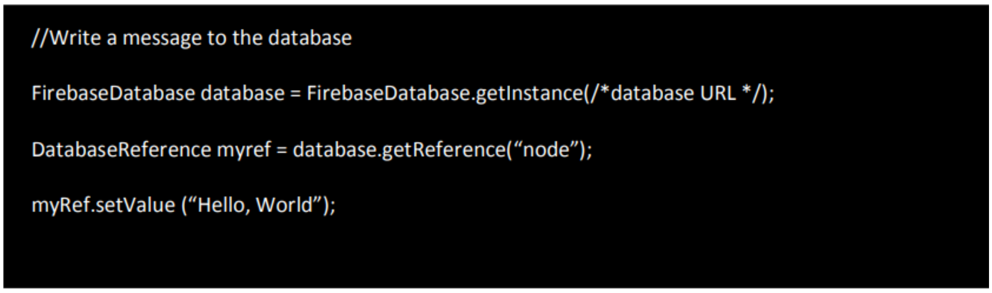


Figure 1: Firebase database structure

User login can also be used from Google, Facebook, etc.

4.2 Database:

Real-time database in Firebase is very easy to integrate with applications. Once the respective dependencies of Firebase and database are added to the app, non-structured data can be added to database by the following Java code [7]:

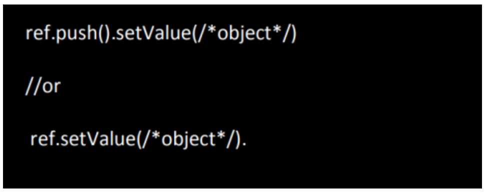


```
//Write a message to the database

FirebaseDatabase database = FirebaseDatabase.getInstance(/*database URL */);

DatabaseReference myref = database.getReference(“node”);

myRef.setValue (“Hello, World”);
```

In Figure 1, the structure of data stored in Firebase real-time database is shown where we have the root node and there are three children of the root that is **admins**, **users** & **data**. The **users**, **admins** & **data** have their child nodes as per the figure. The **admins** and **users** consist of data about Admins and Users of the work-application respectively. The data node contains user_id of users that is **name** and **phone_number** uploaded by the respective users through android app, Boolean value **is_valid** specifies whether the data is validated by the admins or not.

4.3 Inserting New Data:

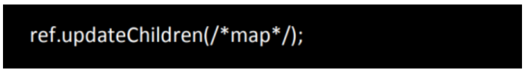
An instance of object class or a map can be used to insert new data. Once the object is created, navigate the Firebase reference to the position where a child should be added. If a created list does not have a specific names for each child, the push() method can be used before the setValue() is called:



```
ref.push().setValue(/*object*/)

//or

 ref.setValue(/*object*/).
```

4.4 Updating Data

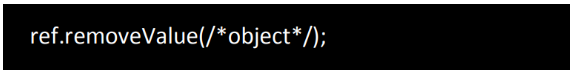
Navigate the Firebase reference to the parent of the item that one wants to update and then a map containing the update values can be created.



```
ref.updateChildren(/*map*/);
```

4.5 Removing Data

Navigate the Firebase reference to the item that one is to be removed.



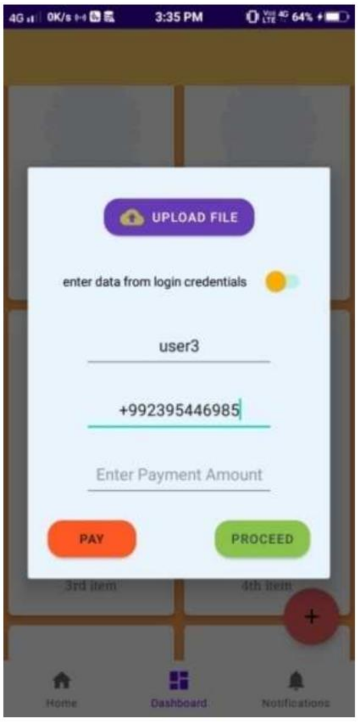
```
ref.removeValue(/*object*/);
```

5. WORKFLOW OF THE SYSTEM

The system is made up of two main modules:

• **Android app**: The app requires sign-up/sign-in for users which demonstrates the Firebase authentication, about which section-4.1 of the paper is discussed with implementation. To demonstrate the CRUD operations on the database, user will enter its name and phone_no which will be stored in data node of the specific user node (refer figure-1).

• **Firebase**: The Firebase works as an online database which stores the data on the cloud storage. The Firebase connects the Android app users to the admins.



enter data from login credentials

user3

+992395446985

Figure 2: Android app

6. RESULT

The Android app presented in this work is tested on a mobile device (RAM: 2GB/ROM: 16GB) powered by Android 8.1.0 and created in Android Studio bumblebee (2021.1.1) using Java & XML. The Windows app presented is tested on computer powered by Windows-10 Pro (64-bit) with processor of 3.1 GHz Intel core i3-2100 processor with 4GB DDR3 RAM and 500 GB ROM.

6.1 Android App

The app contains several different fragments bound to the single activity whereas figure-2 shows relevant User Interface responsible to deal with Firebase operations. Here, the EditText with text-hint ‘Enter Your Name ’ and ‘Enter Phone Number’ accepts the specified input which is later on sent to be stored on Firebase by a click on PROCEED button. The View instances shown in the figure-2 other than these three (2-EditTexts and 1-Button) are not related to the work presented in this paper.

6.2 Firebase Real-time Database

The data received from android app is stored in real-time database, along with the unique user-id of the user logged in on the android app. The admin can validate the data which was uploaded by the Android app user to the Firebase, which will in-turn toggle the value of respective is_valid(Boolean data) for independent data as shown in figure-1 to denote that the data is validated by the admin.

7. CONCLUSION AND FUTURE SCOPE

This paper presents the study about Firebase API and its features. This paper helps in studying how to make simple Firebase Android application. This also helps in making apps faster and efficient as no PHP is required as a third party language to communicate with the database. It provides a secure channel to communicate with the database directly from JAVA for Android application. The study material is based on the online blogs and referring to the examples given. Google has been updating Firebase on regular basis, It can not only be used in Android but also to connect cross platform. The work can be further extended by adding new features and exploring new possibilities in Android and Windows applications.

Acknowledgements

It is our proud duty and privilege to acknowledge the kind of help and guidance received from several people in preparation of this report. It would not have been possible to prepare this project in this form without their valuable help, co-operation and guidance. First and foremost, we thank our project guide Mr. Vivekanand Thakare, HOD, Department of Computer Science and Engineering, Govindrao Wanjari College of Engineering and Technology for his valuable suggestions and guidance throughout the period of this project. We also wish to record our sincere gratitude of Dr. Salim Chavan, Principal, Govindrao Wanjari College of Engineering and Technology for his constant support and encouragement in preparation of this report and for providing Library and laboratory facilities needed to prepare our project report. Last but not least, we would like to thank our advisor, friends, parents, teaching and non-teaching staff of GWCET.

References

[1] Kriha Walter, 2009 NoSQL Databases Hochschule der Medien. Stuttgart Media University. Stuttgart.

[2] Pore Supriya S, Pawar Swalaya B, 2015. Comparative Study of SQL & NoSQL Databases. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET). Volume 4 Issue 5, May 2015

[3] Sharma Vatika, Dave Meenu. 2012. SQL and NoSQL Databases. International Journal of Advanced Research in Computer Science and Software Engineering. Volume 2, Issue 8, August 2012.

[4] Daniel Pan. 2016. Firebase Tutorial. October, 2016.

[5] Cox Landon. 2017. SQLite in Android. March 2017.

[6] Kalsov, 2012. Developer Meet Firebase dated 18/3/17.

[7] "Firebase Realtime Database". Firebase, Inc dated 18/3/17.

[8] Bill Stonehem, Google Android Firebase: Learning the Basics Paperback, 2016 dated 18/3/17.

[9] Isuru Madusanka, Busy programmer‟s guide to Firebase with Android, 2013.

[10] G. Harisson, “10 things you should know about NoSQL databases” dated 18/3/17.

[11] https://www.techrepublic.com/article/10-things-you-should-know-about-nosql-databases/

[12] https://www.android.com

[13] https://developer.android.com/