WWW.IJRPR.COM

**International Journal of Research Publication and Reviews**

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

# Android and Windows Work-System using Firebase

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

[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

**A B S T R A C T**

The websites as well as web applications have been highly dependent upon the huge amount of databases and non-organized information such as videos, text, images, files, audio and other data-file types. Handling the non- structured data is difficult while using the Relational Database Management System (RDBMS). Firebase is in comparison a relatively newer technology for handling large amount of unstructured data better than RDBMS. Firebase is also faster than RDBMS, since it uses JavaScript parser & JSON (JavaScript Object Notation). This paper covers the implementation on utilization of various Firebase services with Windows and Android platforms and focuses to understand the related concepts along with their pros and cons. The further discussion goes on covering some of the Firebase services and functions by developing a work-system of Android app and Windows app connected using a common Firebase instance. This paper covers up the working information about firebase authentication and firebase real-time database. The authentication service provides different methods of authentication of a valid user such as email-password, Gmail and third party identity platforms like twitter, Facebook, etc. The Firebase Real-time database is Firebase's original database with low latency which updates the content in real-time and is hence useful for various applications such as game development, message App, etc.

Keywords: Android, Windows, C#, .net, Firebase, firebasedatabase.net API

## 1. INTRODUCTION

The server which is generally used for developing Android apps is Microsoft SQL Server, Oracle SQL or MySQL. The android app is connected to the server using PHP files. The Firebase ecosystem stores data in JSON format which is "JavaScript Object Notation". The other servers use a table-grid format having multiple rows and columns which stores the data within them.

The Firebase provides NoSQL based database service. There are very less similar cloud-based services like the firebase, such as:

**AWS Mobile Hub-** the AWS Mobile Hub provides a console, pre-integrated within the system, which helps during the development of the applications.

**CloudKit-** It is a framework which is made by Apple and is similar to IOS only, it allows to save data and store assets.

**Parse Server-** To replicate functionality of open source **Parse** server, it was released by Facebook (now named as **META**). Later, Facebook closed this project.

### 1.1 Firebase

Firebase is Google's application development platform which help the developers during building process of high-quality apps. The data is stored in the form of "JavaScript Object Notation" (**JSON**) and query is not required for CRUD i.e. update, delete, insert, etc. operations.

It is the backend of a system that also provides a database service for storage data.

The other available services are:

#### 1.1.1 Firebase Analytics(Google Analytics)

It provides details about app usage statistics of the users. It is used as an app measurement solution that also provides free, unlimited reporting on up to 500 distinct events user engagement. It gives useful insights to the application developer allowing them to understand about how the users are using the application. The Software Development Kit consists of the option of properties and events registering on upon itself and also allows fetching custom user data according to the requirements.

#### 1.1.2 Firebase Cloud Messaging (FCM)

Firebase Cloud Messaging (FCM) provides an efficient connection between the devices and server which allows delivering and receiving messages and notifications on apps and websites at no extra charges.

#### 1.1.3 Firebase Auth

Firebase Authentication provides backend service, ready-made libraries and easy-to-use SDKs for user authentication in the app. It supports authentication using contact numbers, passwords, popular identity platforms like Facebook, Google and Twitter and many more.

#### 1.1.4 Real-time Database

Firebase Real-time database is firebase's original database. For synchronization of app-data, an API is provided and stored on Firebase's cloud storage service. Easy integration with IOS, Android and other types of applications can be achieved using client libraries.

#### 1.1.5 Firebase Firestore

Likewise, Firebase Real-time Database, it keeps the data synchronized across the client apps through data change listeners and offers the offline support for mobile and web, so that the responsive apps can be built to work with or without Internet connectivity of a device.

#### 1.1.6 Firebase Storage

It provides network speed independent file transfer service for the Firebase application. Google Cloud Storage is cost- effective data storage service and it supports Firebase Storage. The developer can use it to store various types of files and formats.

#### 1.1.7 Firebase Test Lab for Android

It is used for Android app testing. Developers can carry various testing processes on their apps as well as run the app on a wide range of virtual devices and their configurations with one single operation. The test results for performed operations are available in the Firebase console page. The Test Lab can automatically analyze the app and look for the bugs and the crashes.

#### 1.1.8 Firebase Crash Reporting (Firebase Crashlytics)

Crash reporting is a real-time service in Firebase which allows quickly troubleshooting and bug fixing in the app by collecting and grouping the app crashes based on their location in the application's code.

1.1.9 Firebase Notifications

It is a freely available service which provides to show targeted user notifications for mobile app developers.

### 1.2 Android App

**Android** is an operating system [12],[13] made for mobile phones which is modified version of the Linux kernel and other open-source software, designed primarily for mobile devices with touch screen interfaces such as 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 consists of useful set of development tools, such as a debugger, software libraries, a mobile handset emulator based on QEMU(Quick EMUlator), sample code, documentation and tutorials. Other development tools are also available such as native development kit (NDK) for the applications.

Using the Android studio and the above listed tools, an Android app is made. The app will also be called as "Third-party app".

### 1.3 Windows App Using C# and .NET framework

Windows apps are made in a variety of languages; each language used is selected according to the requirements for the end-product. For instance, if your app is time critical, like a game, you should be developed in C or C++, or if something that involves AI then the app should use Python with its hundreds of libraries that allows doing anything with it. Overall, what language to be used is really up to the requirements, as long as it has a compiler that supports having Windows targets.

**C# Programming language:** C# is a **modern**, **innovative**, **open-source**, **cross-platform** object-oriented programming language. C# is garbage-collected. It supports different programming paradigms, including structured (particularly procedural), functional programming andobject-oriented programming.

There are numerous libraries made in C# language. In this work, we have used following libraries:

**FirebaseDatabase.net:** this library is a simple wrapper which is based on Firebase Realtime Database REST API. It also supports streaming API which can be used for real-time notifications.

**WinUI3:** WinUI 3 is the native UI platform component that comes with the Windows App SDK. The Windows App SDK provides a complete set of tools and APIs that can be used to create production ready desktop apps targeting Windows 10 and later versions.

## 2. LITERATURE REVIEW

In [1], Walter Kriha has mentioned and overviewed the common concepts, methodologies and patterns in NoSQL databases, along with several classes of NoSQL databases (key/value-stores, document databases, column-oriented databases) and individual products. Supriya S. Pore, Swalaya B. Pawari [2] in their work, has conducted a comparative analysis between SQL and NoSQL databases. The study focuses on SQL and NoSQL databases. The study

concludes that the data consistency is the reason on why ACID properties are not utilized in the NoSQL databases. In [3], Vatika Sharma, Meenu Dave have overviewed the NoSQL databases, their work further explains about how NoSQL databases has lowered the dominance of SQL based databases. In [4], Daniel Pan has demonstrated about connecting a firebase instance with the Android app. It also covers the simple structure designing of database in Firebase. In [5], Landon Cox has done a comparative study of SQLite database which is default database for android development with the firebase. It further covers the method data organization as a JSON tree for storing it into the firebase instance.

## 3. STEPS TO ADD FIREBASE TO ANDROID

Firebase requires minimum of Android version 2.3 (Gingerbread) and at least of Google Play services 9.6.1 version is required. The following are the further requires steps:

1. Firstly, create a project in firebase console page. Later on, enter the project name and choose the location of database. Project name under firebase console might vary from the actual application's given name.
2. Click on "Add Firebase to the Android app" and follow the mentioned steps mentioned in the documentation.
3. The user will be prompted for the package name and optional SHA-1 certificate for encryption, whose data-credentials are available in android studio.
4. At last, a file named "google-services.json" file will be downloaded. The downloaded file has to be pasted into the project's "*app.gradle"* or "*module"* folder. Firebase has been successfully integrated into the project.

## 4. USING FIREBASE SERVICES IN ANDROID APPLICATION

The Firebase services can be used in the Android app by using a few lines of code snippet. The Google Firebase guide link covers the furthermore details about the features, this link is listed in the reference section. The method to use some features are as:

### *4.1 Authentication:*

Once the firebase and respective authentication dependencies are added to the Android app project, the user login id can be created by the below given Java code [6]:

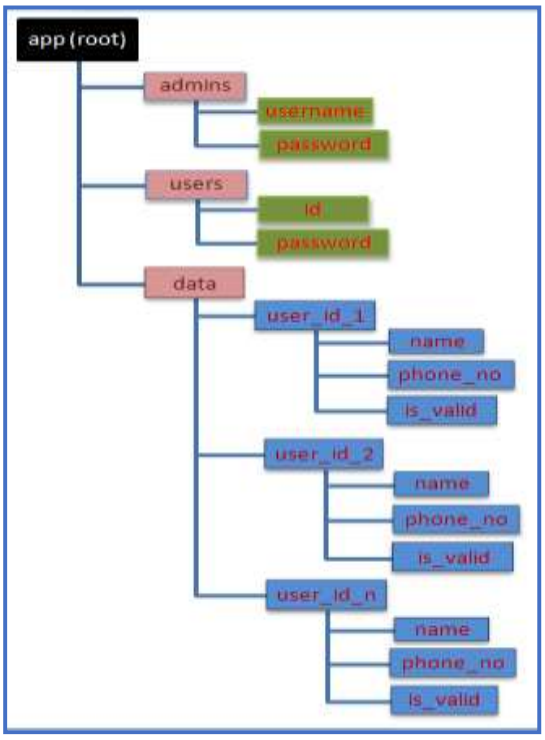


***Figure 1: firebase real-time database data structure***

In Figure 1, the structure of data stored in Firebase Real-Time database is shown where we have the **root** of the app. The three children of the **root node** which are **admins**, **users** & **data**. These nodes also have their respective child nodes as shown in the figure. The **admins** and **users** consist of data about

Admins (Windows app users) and Users (Android app users) of the work-system respectively. The **data** node consists of user id of users following the information (name and phone number) uploaded by the respective users through the android app, the Boolean variable **is_valid** specifies whether the data is checked and validated by the admins through Windows app or not.

Real-time database in Firebase is also very easy to integrate as well as use. Once the required dependencies of Firebase and database are added to the app, non- structured data can be added to database by the below Java code [7]:

```
//Write a message to the database

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

DatabaseReference myref = database.getReference("node");

myRef.setValue ("Hello, World");
```

**Inserting New Data**

A model class instance or a map is utilized to insert new data. After that, navigate the Firebase reference to the inserting position where a child is to be added. The **push**() method is useful before the **setValue**() is called, if a created list does not contains the specific names for individual child:

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

//or

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

***4.2 Updating Data***

Create a child object using model class or map which consists of updated values, use a Firebase reference pointing towards the parent node to update that child.

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

***4.3 Removing Data***

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

Firebase reference pointing towards the parent node is used to remove the child node.

## 5. STEPS TO ADD FIREBASE TO WINDOWS APPLICATION USING PYTHON

In the **section 3**, we set up the Firebase for the Android application. Now, we set up the same for Windows application. The windows app is made using **C#** programming language, the Firebase setup on C# is done using **firebasedatabase.net** API.

To connect to Firebase auth., the following lines of **C#** code are required:

```
var auth = "ABCDE"; // your app secret
var firebaseClient = new FirebaseClient(
  "<FIREBASE URL>",
  new FirebaseOptions
  {
    AuthTokenAsyncFactory = () => Task.FromResult(auth)
  });
```

Firebase is now also set up with **C#**.

## 6. USING FIREBASE FEATURES IN WINDOWS APPLICATION

The work presented in this paper, deals with basic operations done on the database and user authentication.

### *6.1 Basic Operations*

To perform basic operations, the following code [17] is used to set the reference variable to the **root** of the database:

The following functions are used to

```
using Firebase.Database;
using firebase.Database.Query;
var firebase = new FirebaseClient
      ("<firebaseurl path>");
```

perform operations on the database:

- Querying:

```
using Firebase.Database;
using Firebase.Database.Query;

var firebase = new FirebaseClient("FirebaseURL");
var dinos = await firebase
  .Child("dinosaurs")
  .OrderByKey()
  .StartAt("pterodacty1")
  .LimitToFirst(2)
  .OnceAsync<Dinosaur>();
```

- Insert on specific node and let the client generate new key:

```
var newUser = await firebase
  .Child("/*specific node*/")
```

- Overwrite data on specific node:

```
await firebase
  .Child("/*specific
  node*/")
```

- Delete specific child node:

```
await firebase
  .Child("specific
  parent")
  .Child("specific
```

## 7. WORKFLOW OF THE SYSTEM

The system comprises of three main modules:

- **Android app:** The app is made for **users** and it requires **sign- up/sign-in** for users which demonstrates the Firebase authentication, about which **section-4** of the paper is covered with implementation. To demonstrate the **CRUD** operations on the database, the user will enter its **name** and **phone_no** which will be stored in **data** node of the **user** node (refer **figure-1**).
- **Firebase:** The Firebase works as a database which store the data on the cloud storage. The structure of database is discussed in **section-4** and **figure-1** of the paper. The Firebase connects the **Android app** and **Windows app**.
- **Windows app:** The windows app is made for **admins** and it also requires **sign-in/sign-up** for admins, about which **section-6** is covered with basic operations. The person signed in as **admin** manually checks and validates the **data** of the **user**, which demonstrates the accessibility of same information in the Firebase using both apps.

## 8. CONCLUSION AND FUTURE SCOPE

The overview on the study about application of Firebase, some of its APIs and authentication and real-time database services has been done in this paper. This paper helps in understanding the usage of Firebase services which are Firebase authentication and Firebase Real-time database in the Android application as well as Windows desktop application. Firebase integration makes android and desktop apps efficient and faster, since no third-party medium such as PHP is required for establishing connection with the database. Firebase provides a set of APIs for accessing respective services using JAVA for Android application and C# for Windows application. The online available study material and official documentation is the basis of our work and this

paper. Firebase has been getting updated on the regular basis, many new kind of services has been getting introduced by the Firebase too. Firebase can also be used to connect with cross platform applications such as React-Native. Our work in this paper concludes that the Firebase is relatively costly than other available options but the setup process, Integration and compatibility factor makes it a beginner friendly platform because all the heavy workout gets done by the Firebase itself. The work presented in this paper can be furthermore extended by introducing newer functionalities as per the requirements.

## 9. ACKNOWLEDGEMENT

We thank to all the anonymous reviewers for spending their time, calmly reading the paper. Hope this work is worthwhile to your time. We also thank to Dr. Salim Chavan, Principal, Govindrao Wanjari College of Engineering and Technology, for their constant support and encouragement in preparation of this report and for providing Library and laboratory facilities needed to prepare our project report. We also thank our project guide and co-ordinator **Prof. Vivekanand Thakare**, HoD, Department of Computer Science and Engineering, Govindrao Wanjari College of Engineering and Technology for their valuable guidance and all the encouragement that lead towards completion of our project.

Last but not least, we would like to thank our advisor, friends, parents, teaching and non-teaching staff of GWCET.

## 10. 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/

[14] https://www.csharp.org/

[15] https://firebase.google.com/docs/reference/admin/csharp

[16] https://www.freecodecamp.org/news/how-to-get-started-with- firebase-using-csharp /