Firebase as BaaS for College Android Application

Divya Sharma
Dept. of Information Technology
Mulund College of Commerce, Mumbai, India

Hiren Dand, PhD
Dept. of Information Technology
Mulund College of Commerce, Mumbai, India

ABSTRACT

In the present times, words like 'mobile application' and 'cloud' are making a significant mix. Every day there is some new application in the market with the point of conveying the best execution to their clients. Picking the correct BaaS (Backend as a Service) is vital and it comes down to what your project requires. A sturdy infrastructure is built with mobile backend services such as with Firebase which aids developers by handling the backend of these applications. In this paper, the Android application MCCApp utilizes the features provided by Firebase to show its usage as the best Mobile Backend-as-a-service.

Keywords

Firebase, BaaS, Mobile Application, Android, Real-time database, Back-end

1. INTRODUCTION

Android applications use various databases for their back-end. The popular ones used commercially have been SQLite, Realm DB, ORMLite, Berkeley DB and Couchbase Lite. [1] Apart from these, the common ones heard of or implemented in projects are MySQL and Microsoft SQL Server. This method requires the application to be deployed on a server and in order to connect these, PHP files are required. Thus, a lot of coding knowledge is required. NoSQL databases gained popularity in the recent years with examples such as Wasp DB, Couchbase Lite, Forest DB and Firebase Database. [2] These gained popularity due to JSON and Key-Value pairings that gave power to the applications in terms of features such as Shared Preferences.

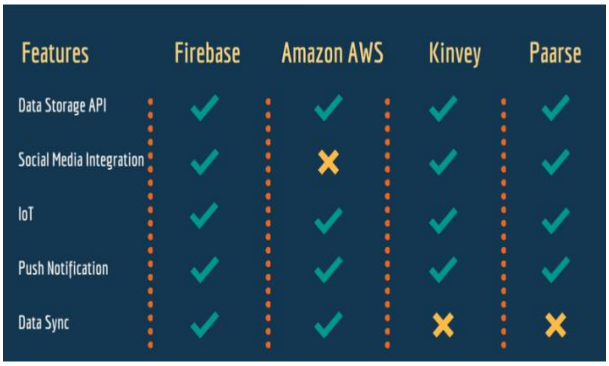
BaaS (Back end as a Service) emerged as the common backend used in today's date. BaaS gives backend to mobile applications, a programming interface, and instruments for different programming dialects to coordinate with the application backend. It likewise gives extra administrations, for example, cloud-based capacity, message pop-ups, push notifications, user account handling and file management. Web and portable applications request practically equivalent to set of highlights on the backend. For instance, email warning, informal community mixes, message pop-ups, distributed storage and so forth. Every one of these administrations has its own API that must be independently consolidated into an application. This is a tedious practice, a procedure that can be tedious and muddled for applications and can be automatized with a BaaS. The most ideal approach to comprehend the idea driving the BaaS is to picture an extension interfacing the backend to the front end of an application. Some of the best BaaS platforms today are as below:

• Amazon web service Mobile- Free, many features, backed by Amazon and utilized by major apps such as Netflix.

• Firebase- Free to begin with, backed by Google and features mainly Realtime Database.

• Parse Platform- Open source, backed by Facebook and a self-hosted backend server which is the SDK for many platforms such as Android.

• Back4App- Free to begin with, cloud hosted parse server and many features.



Features	Firebase	Amazon AWS	Kinvey	Paarse
Data Storage API	✓	✓	✓	✓
Social Media Integration	✓	✗	✓	✓
IoT	✓	✓	✓	✓
Push Notification	✓	✓	✓	✓
Data Sync	✓	✓	✗	✗

Figure 1: Comparison of Baas platforms [3] material.

2. FEATURES OF FIREBASE [4] [10]

The services offered by Firebase are divided into 3 main goal-aiding categories:

2.1 Build better apps –Develop Products [6] [8] [9]

Firebase gives you a chance to construct all the more dominant, secure and versatile applications, utilizing world-class foundation. The services used in MCCApp include:

1. Authentication - Firebase Auth offers various strategies for authentication of the users and provides easy guidelines to code the working.
2. Realtime Database – Uses a cloud-hosted, NoSQL database to offer a great user experience in terms of data accessibility and connectivity.
3. Cloud Storage – Store and share files effectively sorted on the backend.

2.2 Improve app quality – Quality Products

Firebase gives you bits of knowledge into application execution and dependability, so you can channel your assets adequately. The services used in MCCApp include:

1. Crashlytics – Metrics of the app usage and crash statistics.
2. Performance Monitoring – Examine various parts of the application from time to time.
3. Test Lab – Discover bugs and run customized tests.

2.3 Grow your business – Grow Products

Firebase encourages you develop to a large number of clients, improving client commitment and maintenance. MCCApp can use the services below:

1. Remote Config – Custom content to users may be delivered.
2. Dynamic Links – Could be utilized for social media growth and promotions.
3. A/B Testing – Refers to Alpha/Beta Testing, wherein the new features are tested within a small group before being rolled out into the market for all users.

Firebase encourages you develop to a large number of clients, improving client commitment and maintenance. It allows for blending and matching Firebase items to settle basic application improvement challenges. [5] [6]

3. EXISTING SYSTEM

This is the scenario in most colleges that is to be overcome with the help of the MCCApp Android Application:

• A notice is printed on a letter head and signed by officials.

• The printed notice is taken to all the classes via a peon, which is then announced by the Professors to the students.

• The announcement is made to the departments intended for during the different lecture timings. This leads to double work for the person as he has to make sure all class signatures are obtained on the back of the notice.

• Upon managing to reach all classes, the notice is put up on the notice board.

This app is an exciting release for this institution. It enables the college MCC to be up to date with the current trends in terms of educational content and social community. The backend already contains the data of the MCCians. Thus, user is first verified. If not, the only way to access the app is via Guest Profile. Verified users may register with basic info, providing an Email and Password for login purpose. The user Department is automatically detected via the backend. Thus, in My Department Activity the user can see their department name and content based on it. User cannot see another departments posts. If any post can be viewed by all, it will be posted on the overall Notice Board. The administrator may post to the desired area together with attachments. Every post is editable to the admin. Admin may review existing users also. For the case of MCCians not yet updated in the database, admin can unlock their account. Even as it includes so much of information, it is not difficult to use at all. Its interface has been designed to navigate as simply as possible and make it look attractive and interesting. The Confessions Corner was added as an exciting feature for the application.

This application does not have complex requirements to run on a system. It can be downloaded via Play Store for Android phones. It does not require a lot of memory either as the app is compressed for a resourceful release.

4. WORKING ALGORITHM

Firebase has become increasingly popular over the past couple of years. Major application companies have adopted it due its perfect fit with agile development. Colleges usually have their own applications that are created and maintained by a third party organization. In this case, this Android application has the capability of being handled by the creator due to Firebase being the backend that provides all the features in a go. With this, even small institutions can adopt this model and run this application to connect all the members.

The working modules are described as in the figure below:

4.1 Welcome Screen

This is the first module launched upon the start of the application. It checks whether the user is logged in to the app or not. If not, it navigates to the Select Profile Intent, else it navigates to the Dashboard:

```
if (!isLoggedIn) {

Intent i = new Intent(getApplicationContext(), ChooseProfileActivity.class);

startActivity(i);}

else {

Intent toNavigationActivity = newIntent(getApplicationContext(), DashboardActivity.class);

startActivity(toNavigationActivity);}
```

4.2 Profile

This activity allows the user to select a profile from the given that are:

• Student - A MCCian will provide the unique PIN number

• Faculty - Provides unique ID

• Alumni - Was a MCCian so will provide previously owned unique PIN

• Guest - No verification necessary

The verification process involves a SMS sent to the registered phone number for the unique PIN containing a One Time Code. The process relies on Firebase’s Phone Number

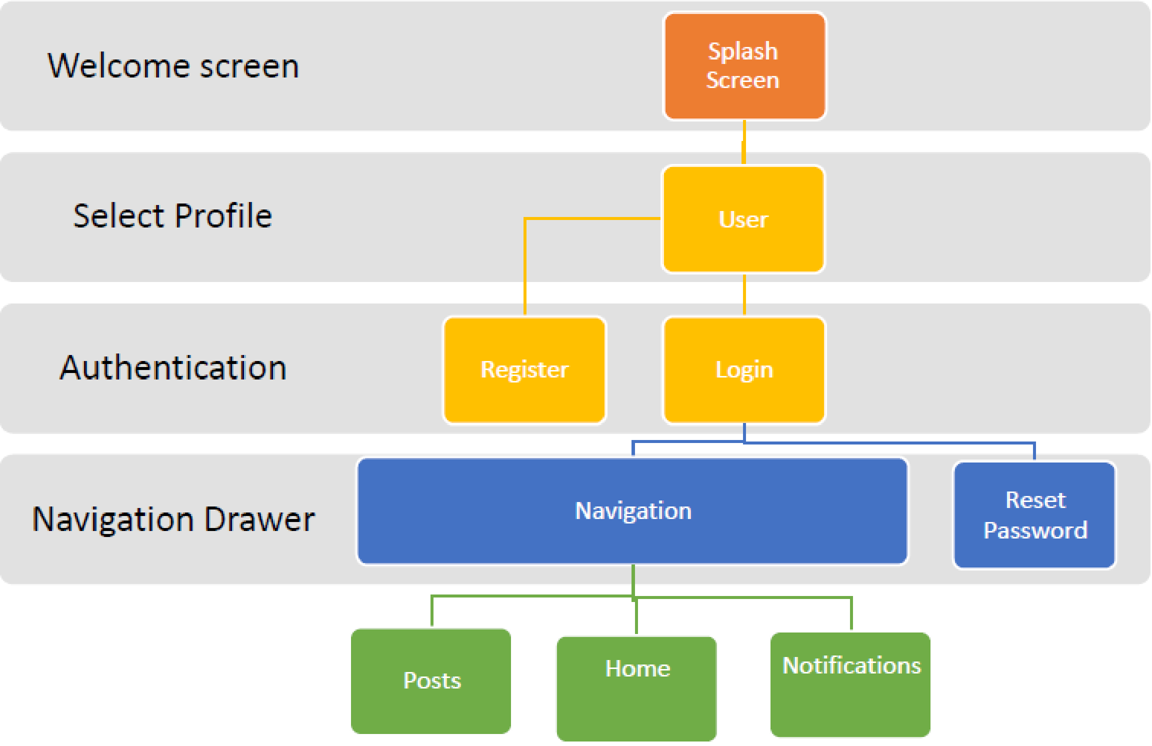


Fig 2: MCCApp Navigation

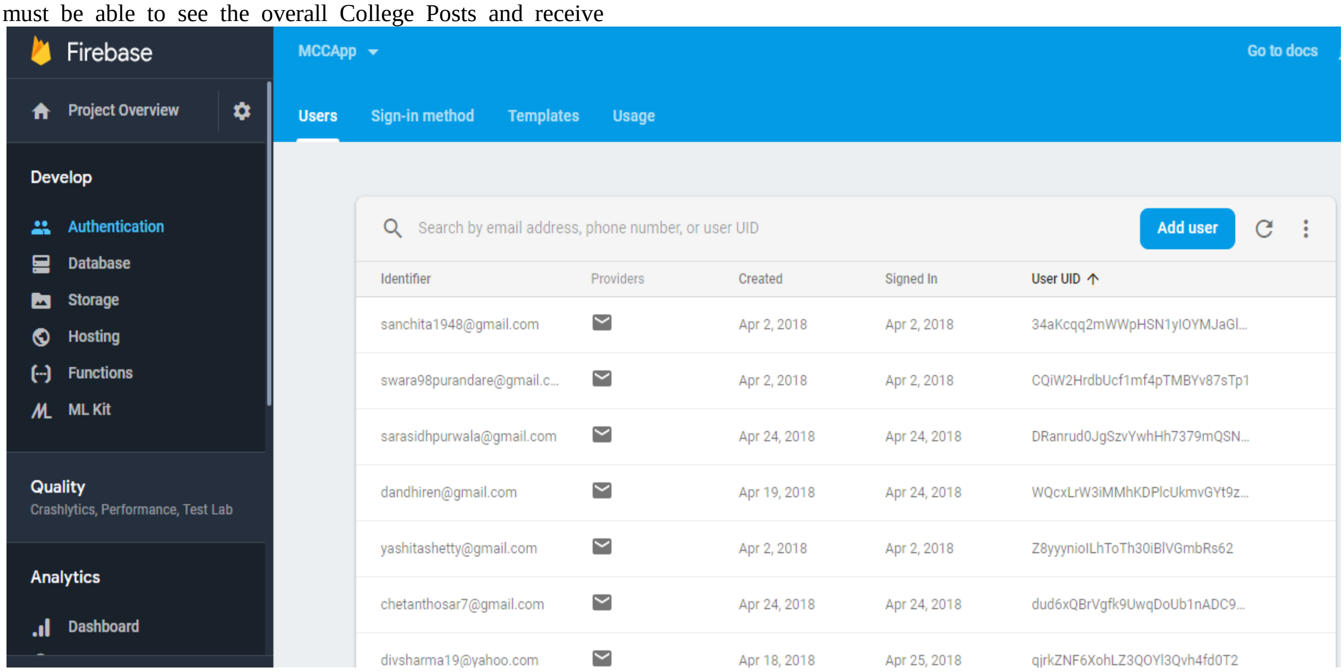
Authentication, as well as server-side scripting. [5][7][8] Upon verification, the user can now move on to the Authentication activity.

4.3 Authentication

The Authentication feature of Firebase allows the verified users to register with the application [10], or log in to an existing account. [7] In case the user forgets the password, Firebase's Reset Password feature kicks in. The user should be verified if a MCCian. Non-MCCians should be able to access the basic information via the Guest profile. All users must be able to see the overall College Posts and receive Notifications. This is done with the help of SharedPreferences where each user type is assigned a ROLE in Constants:

public static String KEY ROLE =”ROLE”;

Using this constant it can be determine who is logged in and what should be displayed accordingly. The Admin must be able to view everything, however a Guest user cannot view any Departmental Information, neither can they upload anything.



Identifier	Providers	Created	Signed In	User UID ↑
sanchita1948@gmail.com	✉	Apr 2, 2018	Apr 2, 2018	34aKcqq2mWWpHSN1yIOYMJaGl...
swara98purandare@gmail.c...	✉	Apr 2, 2018	Apr 2, 2018	CQiW2HrdbUcf1mf4pTMBYv87sTp1
sarasidhpurwala@gmail.com	✉	Apr 24, 2018	Apr 24, 2018	DRanrud0JgSzvYwhHh7379mQSN...
dandhiren@gmail.com	✉	Apr 19, 2018	Apr 24, 2018	WQcxLrW3iMMhKDPlcUkmvGYt9z...
yashitashetty@gmail.com	✉	Apr 2, 2018	Apr 2, 2018	Z8yyyniolLhToTh30iBlVGmbRs62
chetanthosar7@gmail.com	✉	Apr 24, 2018	Apr 24, 2018	dud6xQBrVgfk9UwqDoUb1nADC9...
divsharma19@yahoo.com	✉	Apr 18, 2018	Apr 25, 2018	qjrkZNF6XohLZ3QOYl3Qvh4fd0T2

Figure 3: MCCApp Authentication

4.4 Home

•Depending on the user profile, the Dashboard contents are dynamically applied based on what the user is allowed to see.
•Admin profile sees everything, and can update anything.

•The Navigation Board differs based on the ROLE identified:

```
prefs = getActivity().getSharedPreferences

(Constants.SHARED PREFERENCES FILE NAME, MODE PRIVATE);
```

• There are three main fragments in the Home Activity, namely Home, Posts and Notifications. These are for the overall college posts. The contents include dynamic public posts, as well as static data. Upon updations, the user receives a notification with the timestamp.

• The department is auto-detected from the database and thus its respective posts are displayed.

5. USING FIREBASE DATABASE

The main step initially was to convert the excel data file received from the college office into JSON format so that all the data is imported in order to be utilized.

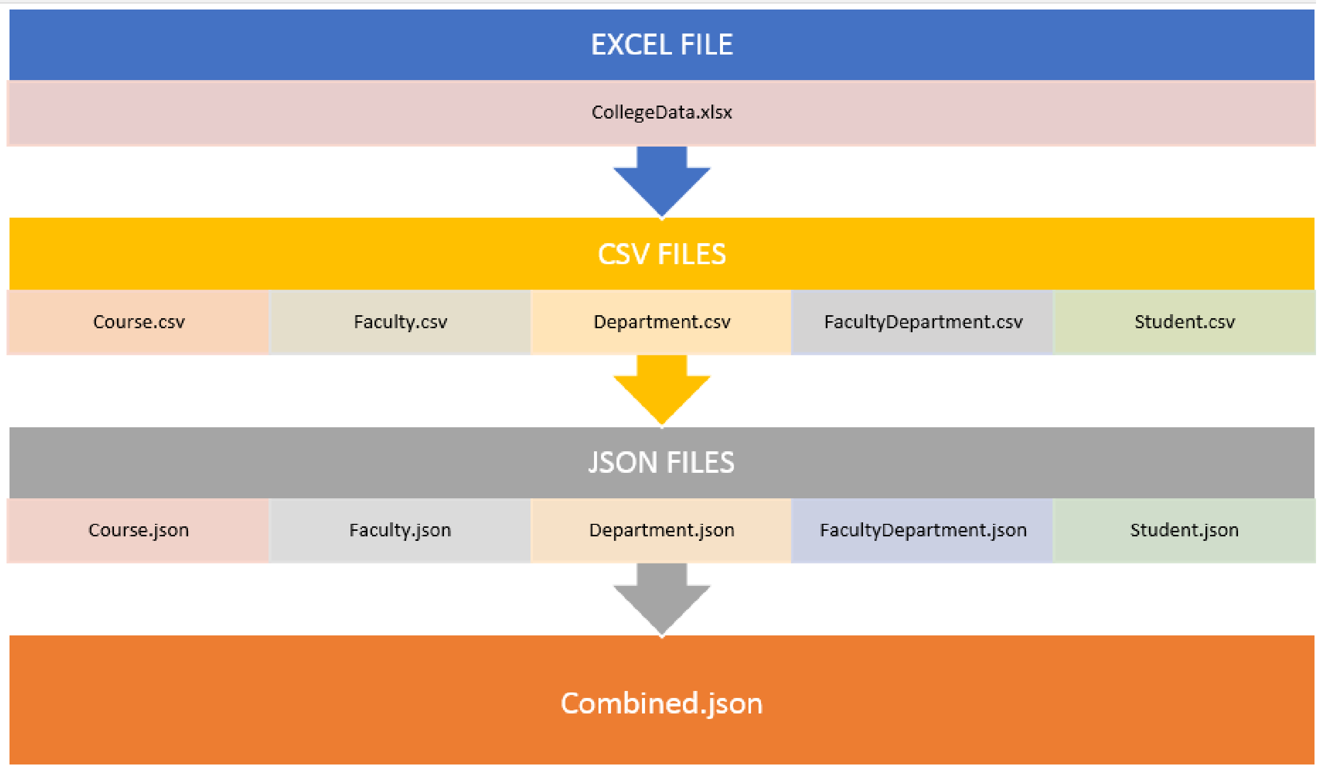


Figure 4: MCCApp Database Files

This file produced is uploaded onto Firebase using the Import feature to load the data onto the database to produce the given keys with their values. Each file here acts like an individual table. The other tables are added via code.

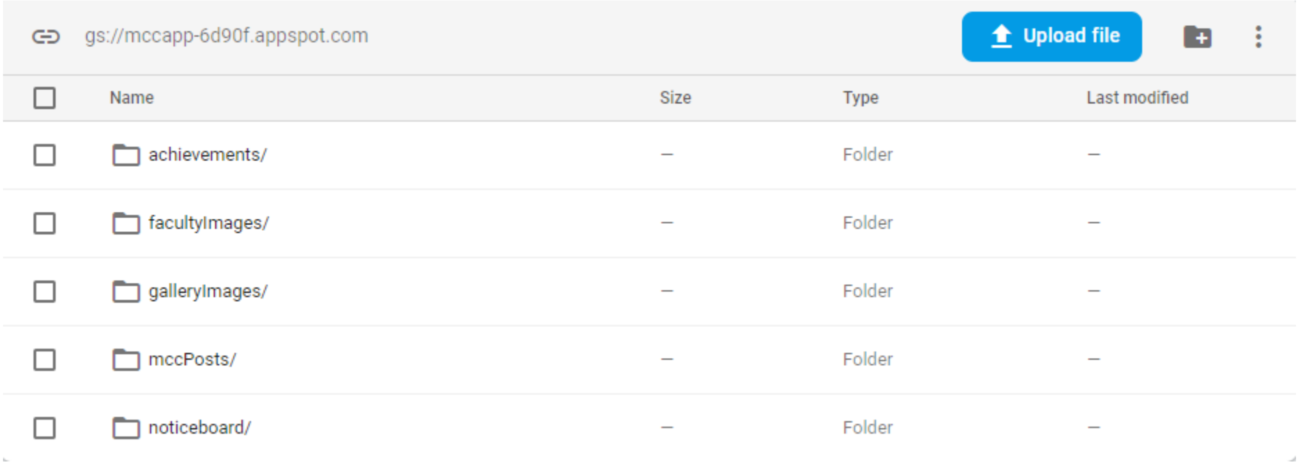
5.1 POJOs

POJOs (Plain Old Java Objects) are the classes which are being used for transferring data. In MCCApp, data is fetched from Firebase and kept in POJO Objects.

```
package com.devdroid.mccapp.pojo;

/** * Created by Divya on 19-Apr-18. */

public class MCCNotification{

private String notificationText; private String
notificationDate;

public String getNotificationText(){

return notificationText;

}

public void setNotificationText(String notificationText) {

this.notificationText = notificationText;

}

public String getNotificationDate() {

 return notificationDate;

}

public void setNotificationDate(String notificationDate) {

this.notificationDate = notificationDate; }}
```

5.2 Storage Bucket

This feature of Firebase allows pictures to be stored in the associated folders. They include the ones that are uploaded by the users [10].



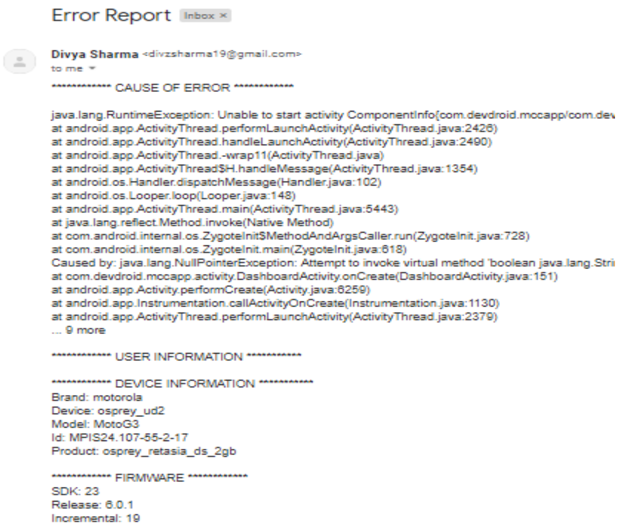
□	Name	Size	Type	Last modified
□	achievements/	–	Folder	–
□	facultyImages/	–	Folder	–
□	galleryImages/	–	Folder	–
□	mccPosts/	–	Folder	–
□	noticeboard/	–	Folder	–

Figure 5: MCCApp Storage Bucket

5.3 Debugging

MCCApp faces crashes due to bugs at endless moments. Thus, being able to debug is essential. The process itself is systematic as you have to understand your code to be able to debug. Firstly, the app should run in debug mode. Breakpoints may be inserted where necessary. This allows you to inspect the values and expressions line by line. Firebase Crash Reports aid in the developer console. Despite its features, the Crash Report was taking at least half an hour to appear on the Console. Thus, it was essential to come up with a faster and efficient way of handling exceptions to avoid abrupt crashes. The ExceptionHandler class created caught any sort of exception causing a crash, and instead opens the Email Intent. Here, the Stack Trace is feeded automatically and the user simply clicks Send for the developer to receive all details. These include:

- User Name
- Brand
- Device
- API



Divya Sharma <divzsharma19@gmail.com>
to me

```
************ CAUSE OF ERROR ************

java.lang.RuntimeException: Unable to start activity ComponentInfo{com.devdroid.mccapp/com.dev
at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:2426)
at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:2490)
at android.app.ActivityThread.-wrap11(ActivityThread.java)
at android.app.ActivityThread$H.handleMessage(ActivityThread.java:1354)
at android.os.Handler.dispatchMessage(Handler.java:102)
at android.os.Looper.loop(Looper.java:148)
at android.app.ActivityThread.main(ActivityThread.java:5443)
at java.lang.reflect.Method.invoke(Native Method)
at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:728)
at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:618)
Caused by: java.lang.NullPointerException: Attempt to invoke virtual method 'boolean java.lang.Stri
at com.devdroid.mccapp.activity.DashboardActivity.onCreate(DashboardActivity.java:151)
at android.app.Activity.performCreate(Activity.java:6259)
at android.app.Instrumentation.callActivityOnCreate(Instrumentation.java:1130)
at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:2379)
... 9 more

************ USER INFORMATION ***********

************ DEVICE INFORMATION ***********
Brand: motorola
Device: osprey_ud2
Model: MotoG3
Id: MPIS24.107-55-2-17
Product: osprey_retasia_ds_2gb

************ FIRMWARE ************
SDK: 23
Release: 6.0.1
Incremental: 19
```

Figure 6: MCCApp Error Report via Email

6. CONCLUSIONS

MCCApp Android Application was made such that Mulund College of Commerce finally gets its own App. Today, applications are so common, making it a requirement for institutes to reach out to the users. The main aim of the project was to enhance user interaction, allowing for basic information to be conveyed, as well as real-time updates. Firebase as the backend allowed for the smooth agile process implemented. The developed system:

1 Allows users to access the app based on the profile.

2 Verifies MCCians.

3 Students and Faculty can see their respective Department.

4 Confessions corner can be unlocked for Students and Faculty.

5 Admin can upload and manage posts.

6 Alumni can register for future event updates.

Thus the MCCApp Android Application is an effective and efficient app fulfilling the needs required by an organization. The posts allow College events to be noted by all users. Data is going to be preserved carefully for longer period hence it gives more reliability. More comfortable for all users because of its simplicity. Users can get the required information via the fragments. At the end, the basic requirements of the application for our college were covered as per the requirements analysis.

7. REFERENCES

[1] Roukounaki, K. (2014, September 10). Five popular databases for mobile. Retrieved from Developer Economics: https://www.developereconomics.com/five-popular-databases-for-mobile

[2] Esplin, C. (2016, October 4). What is Firebase? Retrieved from How To Firebase: https://howtofirebase.com/what-is-firebase-fcb8614ba442

[3] Fatima, H. (2017, December 12). Why is Firebase the best Mobile Backend-as-a-Service. Retrieved from blog.resellerclub: https://blog.resellerclub.com/why-is-firebase-the-best-mobile-backend-as-a-service/

[4] Google. (2018, October 29). Products. Retrieved from Firebase: https://firebase.google.com/products/

[5] Sevilleja, C. (2016, May 23). A Look at the New Firebase: A Powerful Google Platform. Retrieved from Scotch: https://scotch.io/bar-talk/a-look-at-the-new-firebase-a-powerful-google-platform

[6] Khawas, Chunnu & Shah, Pritam. (2018). Application of Firebase in Android App Development-A Study. International Journal of Computer Applications. 179. 49-53. 10.5120/ijca2018917200.

[7] Singh, Navdeep. (2016). Study of Google Firebase API for Android.International Journal of Innovative Research in Computer and Communication Engineering. Vol 4, issue 9, September 2016.

[8] Gupta, Shashank & Kapoor, Bhaskar. (2016). Firebase In App Development.International Research Journal of Engineering and Technology. Vol 2, issue 12, December 2016.

[9] Pratik Lahudkar, Sarita Sawale, Vijay Deshmane & Krishna Bharambe. (2018). NoSQL Database - Google's Firebase: A Review. International Journal of Innovative Research in Science, Engineering and Technology. Vol 7, issue 3, March 2018.

[10] Shashank Kumar Singh. (2017). Firebase API for IOS. International Journal for Scientific Research & Development. Vol 5, issue 4, April 2017.

[11] Andreas Ragil Wiratno1 & Khafiizh Hastuti. (2017). Implementation of Firebase Realtime Database to track BRT Trans Semarang. Scientific Journal of Informatics. Vol 4, issue 2, November2017.