



INTEGRATED ADAPTABLE MOBILE BASED PLATFORM FOR UNIVERSITY/COLLEGE MANAGEMENT USING FIREBASE

G Dhanush

Department of Computer Science & Engineering
Presidency University
Bengaluru, Karnataka, India

Dr KG Mohan

Department of Computer Science & Engineering
Presidency University
Bengaluru, Karnataka, India

Abstract— Bagpack is an Mobile Application built in Android which provides a platform for both faculties and college students to possess a much better interaction. This app has eccentric and helpful features like material uploading portal, Smart Attendance management system, group chatting interface etc. The major idea behind Bagpack is to provide a generalized mobile platform for college management with enhanced usability standards and best reliable data managing backend. This paper describes how effectively the college management process could be advanced by integrating conventional materialistic user interface with Firebase through Bagpack.

Keywords— Android, Firebase, backend, material uploading portal, Smart Attendance management system, group chatting interface.

I. INTRODUCTION

In this era of advancement, everything is simply as away as a click of a button through software. The usage of smart mobile phones has increased drastically. These mobile applications are highly sophisticated and simplified to use.

“Bagpack” is that the proposed software system which mainly associates the students and faculty with one another using android application. The proposed work has following major modules: 1. Classroom Creation 2. Share-access environment for files 3. Attendance management portal 4. Discussion forum through group chat. And additional features like resetting forgot password, email notifications to students. Bagpack uses Firebase infrastructure for the mobile backend to maintain the data that is being generated. The major tools involved are Firebase real time database, Firebase Authentication, Firebase storage bucket. Firebase cloud messaging.

Classroom Creation module, facilitates faculty in creating a classroom with a unique perceptible name, As soon as the faculty creates a particular course a unique 9-digit proctor code will be generated which can then be shared to all students.

Share-access environment module is a feature that facilitates the faculty to share all subject related documents or announcements. This mechanism involves the access of storage bucket, in which all the documents can be accessed with an URL token that has been auto-generated by the Firebase storage. That URL is stored in firebase real time database under the node of that classroom created by the faculty with additional descriptions and timestamps pertaining to the data file stored. On the other hand students, once joined or registered to a particular classroom will then be eligible to access the files that were uploaded by faculty. Now those files are in-app access, where any student is allowed to view that file in the mobile app through his/her account itself, and are not facilitated to download the file because the furnished goal is to promote remote data access.

Attendance Management Portal is an environment where faculties can take attendance for the classrooms that they have created. The major backend involved here is real time database. Now the faculty is supposed to load the student list of a particular class and then mark their attendance accordingly. The instance of marking their attendance is the instances of the lecture that will be attached as unique nodes in real time database. On the other hand students can view their own attendance percentage for every class that they have registered.

Discussion forum is enabled by having conventional interface in the form of group chat. Here all the students and faculty of that class can have a discussion session through chatting.

Apart from these modules, Bagpack has basic modules like Login, Logout, session tracking, reset forgot password and dashboard.

II. LITERATURE REVIEW

To manage academic related activities for any university/college there are diversified applications that were made available. Each application was built with unique features of its own, by analysing their own requirements of that institute. But each of these applications was developed to achieve single academic purpose. [4] Bagpack addresses major college activities using mobile phones.



Now-a-days, 'mobile application' and 'cloud' are making a significant mix in terms of application development. Recognizing this combination many applications are being launched in the market giving the best experience to the end users. Backend plays an important and predominant role in providing this user experience and it relates to what the project demands [2].

Abhishek Tyagi et al.(2018) in [1] presented about, integration of multiple use cases into a single problem and developing solution to it using android. They also worked on enhanced communication between different actors.

Divya Sharma et al.(2019) in [2] built a solid infrastructure with mobile backend as Firebase, which aids developers to handle the backend of these applications more effectively. In this paper, the Android application was built that utilizes firebase as backend to showcase the features provided by Firebase and its usage as the best Mobile Backend-as-a-service. The major focus of work is to show case the abilities, advantages and limitation that were bounded with the usage of firebase. They have also worked on comparison of BaaS platforms, and the work depicting in development of better apps using firebase as backend.

Deeksha P Shetty et al.(2017) in [3] told that the major objective is to implement an android based application for attendance management so as to digitalize the process and used firebase as backend.

In [5], it geared towards developing a mobile application on college academic management through mobile phones that is of important to either an institution or a college. This Application can be used as a knowledge/information academic management system for the college/university.

Sai Spandhana Reddy Emmadi et al.(2019) in [6] developed an android application where the primary goal is to create a system which can create chat room according to the users and store the related data at a single place. To implement the real time chatting application, that allows the user to make instant group messages by implementing Firebase real time database and Firebase cloud messaging.

In [8], college/university academic management system focuses on staff's, lecturers, and students of that institution. The software product is focused to help the faculty for the progression and the academic development of the institution i.e. by minimizing the conventional manual work, checking the performance factors of the student(s).

Ritika Dhiman et al. (2019) in [9] made a mobile application named Breeze, which provides a usual platform, which is easy to use. The major workflow involves the college academicians to develop a better interaction with colleagues and

management. They also maintain feature like raising queries, where all students are allowed to post queries and everyone can answer those queries that were raised. They have also used online software backend system firebase. The app also has features where user can see regular updates of his/her attendance, syllabus, and timetable.

Chatterjee et al. (2018) in [11] made a mobile app keeping in mind that how communication plays vital role in data exchange. Real time access of data is the key point of this paper. They have used firebase for the real time communication of the data processing from UI to cloud and vice-versa.

Major concentration of research is purely on the android applications as android has taken the highest reach of about 87 percent market share by 2019. Design of views [12] that facilitates user in terms of usability in HCI and chatting interface are the triggering points of Backpack. Selection of relevant backend would really help the view components that are bound to be built using android. Research carried on various views' designs like recycler view, card view, view pager, fragmental design approach, and grid layout etc.

III. SYSTEM ARCHITECTURE

The system architecture that Backpack follows is 3-tier architecture as depicted in Fig. 1.

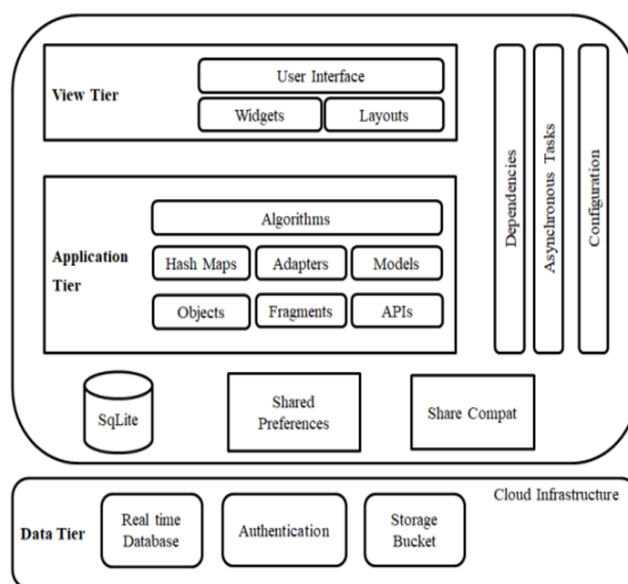


Fig. 1. System Architecture

A. View Tier:

In this tier, the actual discussion is purely on the user interface design and quality attributes that were used as part of designing the UI. This tier is the only gateway where any user can communicate with the product model and gets his/her jobs



done, so the UI for which the users are expected to use with should be simple and minimalist.

The layouts that we typically used in designing the screens are Linear Layouts, Relative Layouts, Frame Layouts, and Grid Layouts. All these layouts have their own significance in designing the UI.

The widgets that we have used in the model are minimal and accurate to the requirements and these widgets are inflated into layouts based on the presentation logic that was written to render the view components on screens.

B. Application Tier:

In this tier, we discuss majorly on the algorithms, different collection frameworks that were used while handling data from cloud. Application tier framework is as follows,

Client request from UI → Data processing in Application Tier → Cloud Sync → request hit to cloud → Query processing in cloud → response hit back to Application tier → Data pulled from cloud to java objects → response shown to user in view tier.

So, in the above shown process of pulling and pushing of data from/to cloud infrastructure, we need accurate java collection framework that can actually process the data and render in view modules.

We have used Hash Maps as one of the Collection framework tool to pull data from firebase. Hash Map really helped a lot because of its key-value pair feature and typically the same is required for JSON in firebase. Apart from Hash Maps, we have used List objects, Adapter classes to render the data into view tier.

We have used Fragments for the best of the design to render the different screens in the same activity, and we have used APIs as dependencies in our android project and those dependencies will intern act as a trigger to use remote libraries in our project

C. Data Tier:

In this tier, the entire discussion point is all about Cloud infrastructure. We have lot of features in firebase to access with and majorly we have used three things.

- Firebase Real time Database → Our Database
- Firebase Authentication → All users will get authenticated at the time of login
- Firebase Storage bucket → All files, images, media will be stored here [7].

So, in firebase we have the processing power of real time data, it means that data that we send to cloud will depict the change immediately in no time, probably the sync time to cloud is very less that is less than two seconds. The upload speed of PDF files, images etc. into storage bucket are purely dependent on Internet connectivity. Dependencies, Asynchronous Tasks and configurations are defined to use diversified features and Share Compat is basic intent builder that trigger the action into other sharable applications like

whatsapp, facebook, shareIt etc. SQLite is used to store user specific data which will intern act as a shared preference and used to get the unique id of that user until the session expires.

IV. WORK FLOW

The major workflow involved for pushing and pulling data in the application is described in the following sections:

A. Pushing Data to cloud

In this work flow the major content deals with the data write access with the cloud storage. Here, the application will get its context and getData() method will be invoked so that it will fetch the data that is mostly likely to be pushed to cloud. After that, all the data objects are ready, application will make a cloud sync request and cloud returns a response object as shown in Fig. 2.

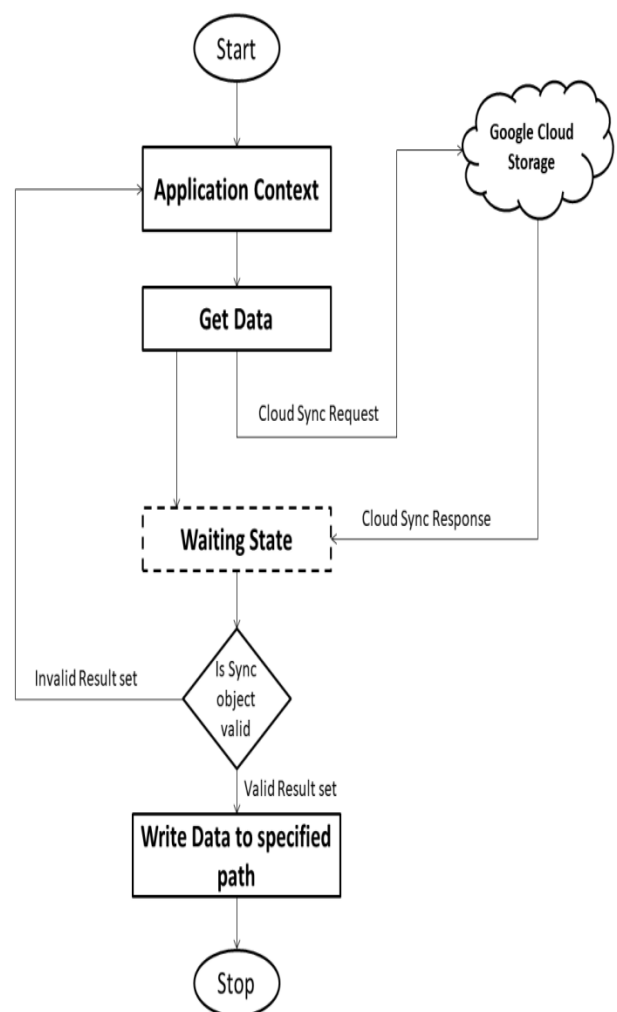


Fig. 2. Workflow for pushing data to cloud

On validity of the response object that data will either be written/pushed to cloud or the data will be rolled back.

B. Pulling Data from cloud

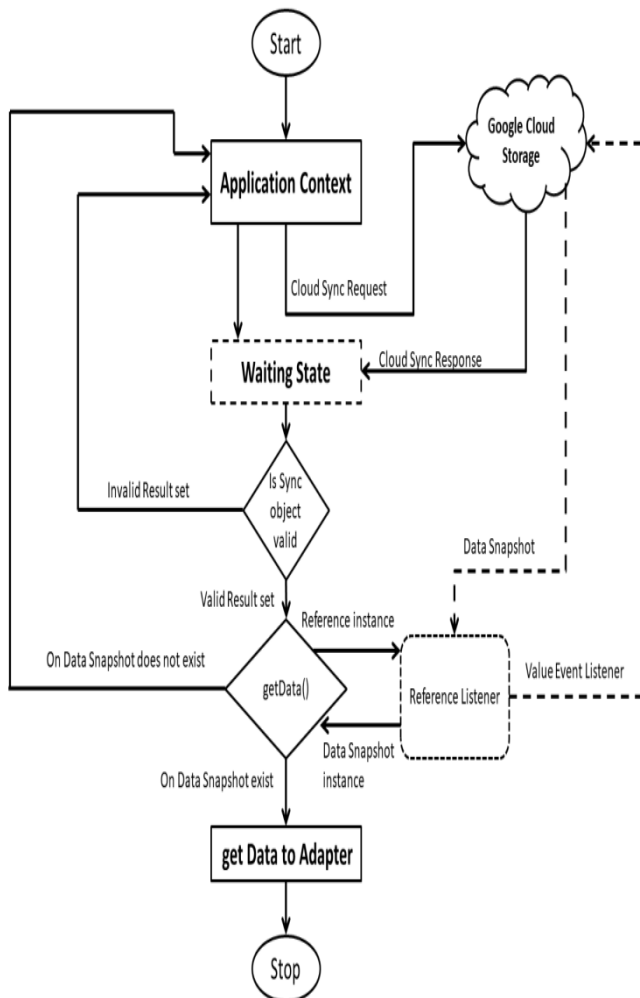


Fig. 3. Workflow for pulling data from cloud

In this work flow the major content deals with the data read access with the cloud storage. Here, application will make a cloud sync request and cloud returns a response object as shown in Fig. 3.

Cloud sync request was made to cloud for connection purpose the cloud will then generate a cloud sync response. On validating the sync object we can justify whether the cloud service/ server is available or not.

The request object is bounded with an event listener on which we call the API with path where the data needed to be fetched. Now the cloud will return a data snapshot in which the entire data will be present as specified by the path. If data snapshot object is not null then we can read the data, else it has no data at the specified path.

V. RESULTS AND DISCUSSION

A. Responsive traits

The final product built has satisfied all the requirements that were elicited.

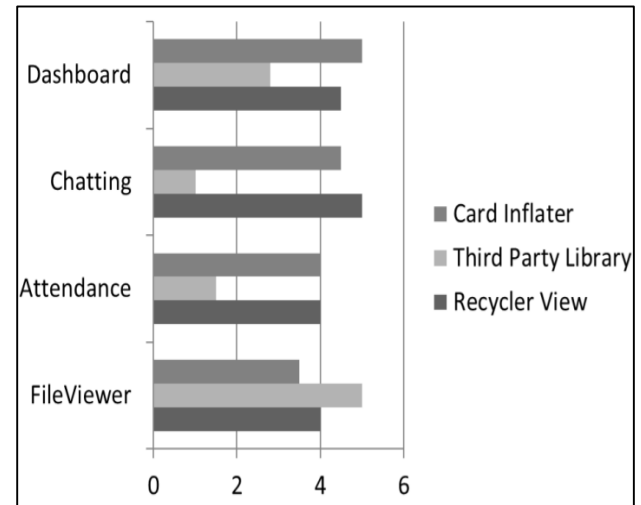


Fig. 4. Comparison of view factors used in few Modules

The graph in Fig.4 shows the usage of few responsive view factors that help in rendering the layout of the modules as desired in different screen resolutions.

B. Performance Factors

The app was analysed for its performance in real time, for which the android profiler tool was used. The android profiler tool provides real-time data to help understand how the app uses CPU, memory, network, and battery resources when deployed to physical handset. [10]

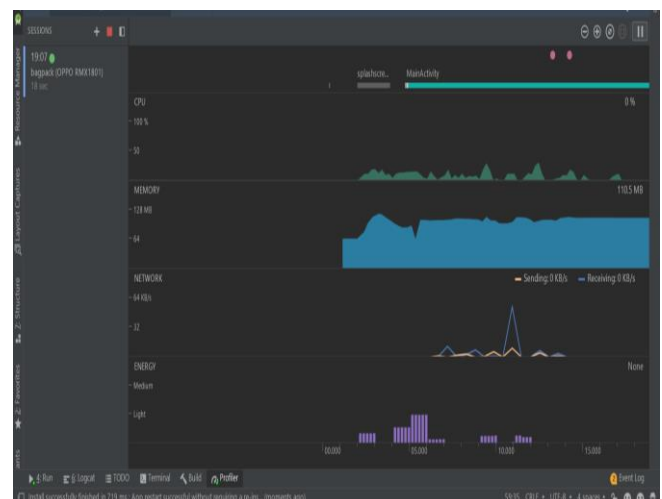


Fig. 5. Profiler interface to monitor performance of the app

In Fig. 5. The app was connected to the physical handset through USB cable, and then in android studio the profiler



option was enabled, immediately the interface would have started depicting the performance in the form of visualization factors. The app continuously was monitored through the profiler, which was capturing each and every action(s) that was performed on the app. Even the small entry of data to edit text was also monitored and was reflected on the screen. The code written for a use case is thoroughly monitored using the profiler that tracks the performance of an algorithm used for that particular use case.

The major observations that were inferred from the profiler was overall analysis about CPU utilization, Memory required, Network used, Consumption of battery.

The below mentioned table(s) are the generalized chart of the performance of Backpack in few modules.

TABLE I. ANALYSIS OF DASHBOARD & CHATTING MODULES

Performance Measure	Dashboard		Chatting	
	Min Observed	Max Observed	Min Observed	Max Observed
CPU Utilization	10%	50%	15%	59%
Memory Requirement(RAM Used)	68MB	118.9 MB	79MB	200MB
Network Used	10 KB/S	279 KB/S	5.7 KB/S	20 KB/S
Battery Consumption	Light	Medium	Medium	Medium

TABLE II. ANALYSIS OF ANNOUNCEMENT MODULE

Performance Measure	File Upload		File Retrieve	
	Min Observed	Max Observed	Min Observed	Max Observed
CPU Utilization	8%	45%	10%	59%
Memory Requirement(RAM Used)	64.7MB	134.1 MB	65MB	124MB
Network Used	As per file size	As per file size	As per file size	As per file size
Battery Consumption	Medium	Medium	Medium	Medium

TABLE III. ANALYSIS OF ATTENDANCE MODULE(S)

Performance Measure	Attendance upload		Attendance view	
	Min Observed	Max Observed	Min Observed	Max Observed
CPU Utilization	7%	34%	11%	69%
Memory Requirement(RAM Used)	55MB	97 MB	65 MB	130 MB
Network Used	100 KB/S	256 KB/S	65 KB/S	170 KB/S
Battery Consumption	Light	Light	Light	Medium

VI. FUTURE WORK AND CONCLUSION

Currently the app is limited with number of files to be shared at once, group notification on receiving a message in classroom. So, as part of future scope of the app, these features would be added as updates.

Backpack has been envisioned for the purpose of reducing the efforts that occur in the traditional management system. The aim is to automate and make a system that is useful for educational institutions. This method is secure enough, reliable and available for use.

VII. REFERENCES

- [1] Abhishek Tyagi, Shubham Godiyal, Shivani Sharma,(2018) "A literature survey on digitizing the college management ecosystem by using android mobile application", International Journal of Advance Research, Ideas and Innovations in Technology, Vol. 4 Issue 2, 2018
- [2] Sharma, Divya & Dand, Hiren. (2019). Firebase as BaaS for College Android Application. International Journal of Computer Applications. 178. 1-6. 10.5120/ijca2019918977.
- [3] Deeksha P Shetty, Gowri N Murthy, Gaurav Rungta, Kundan Kumar(2017),"Android Attendance Management System", IJARIE-ISSN(O)-2395-4396, Vol-3 Issue-2, 2017
- [4] Ms. Niharika Dedhia, Dr. V. C. Kotak, (2017) "ANDROID BASED CAMPUS SOLUTION FOR COLLEGE MANAGEMENT SYSTEM", International Journal of Computer Science and Mobile Computing, Vol.6 Issue.11, November- 2017, pg. 12-17
- [5] Pooja Naik, Kavita Kattimani, Suvarnamala Divate,(2015) "Android Application On College Management", International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE) ISSN: 0976-1353 Volume 14 Issue 2 –APRIL 2015.
- [6] Emmadi, S.S.R. & Potluri, Sirisha. (2019). Android based instant messaging application using firebase. International Journal of Recent Technology and Engineering. 7. 352-355.
- [7] <https://firebase.google.com/docs>
- [8] K. Vivek Thangam, T. Shyam Kumar,V.Yogesh kumar and S.Prabhu,(2017) "Android Application For College Management System (M-Insproplus)", International Journal of Modern Trends in Engineering and Research (IJMTER),Volume 04, Issue 2, [February– 2017] ISSN (Online):2349–9745; ISSN (Print):2393-8161.
- [9] Ritika Dhiman, Ayush Basral and Dr. Jaswanti, (2019)"A New Android Application (Breeze) for CollegeManagement System"2019 3rd International



conference on computing methodologies and communication (ICCMC). IEEE, 2019.

- [10] <https://developer.android.com>
- [11] Chatterjee, Nilanjan & Chakraborty, Souvik & Decosta, Aakash & Nath, Asoke. (2018). Real-time Communication Application Based on Android Using Google Firebase. IJARCSMS. 6.
- [12] <https://material.io/develop/android/docs/building-from-source/>