



Decentralized Isolated Educational Content Aggregation System

Roshan Shibu¹, Preveen Raj², Yakoob Youseph Cyrus², Dr. Vargheese S. Chooralil⁴

¹Rajagiri School Of Engineering And Technology, India, roshanshibu97@gmail.com

²Rajagiri School Of Engineering And Technology, India, 97preveenraj@gmail.com

³Rajagiri School Of Engineering And Technology, India, yakoobcyrus@gmail.com

⁴Rajagiri School Of Engineering And Technology, India, varghesesc@rajagiritech.edu.in

ABSTRACT

In this age of digital transformation, where every industry equips itself to carry out its operations online, it is often the case, especially in developing nations, that many primary and secondary educational institutions lack the means to realize a quality implementation of such transformation. In this paper, we discuss an educational technology platform that enables academic institutions to share, organize and maintain their educational content along with hosting functions including (but not limited to) parent-teacher communication, event notifications and test taking. The platform is presented as an Android app and uses the Google Firebase for data storage allowing seamless scalability and cost-efficiency.

Key words: educational technology platform, digital transformation, parent-teacher communication, school app

1. INTRODUCTION

In a survey, conducted in Southern India, it is found that 85% of schools did not have an app or website to organize or share educational content. One reason for low adoption rates of such technology is that enterprise solutions are not financially feasible for majority of the schools. On the other hand, institutions that develop in-house solutions are left with lower quality products.

Drawbacks of current implementations

One reason for the economic infeasibility of the enterprise solutions is that the lack of flexible payment options considering the hugely heterogeneous spending powers of different educational institutions. It is also found that such enterprise solutions encompasses a multitude of functionalities that the client, viz teachers or parents, are unaware of or do not make ample use of due to poor awareness of the capabilities of such systems[1].

In the case of in-house solutions, the end product is one that is hugely specific to the needs of the institution. However it is quiet common that such implementations lack regular maintenance and updates unless specifically requested by the institution.

Scope and Objectives

The project revolves around the creation of an android app at both server and client sides. There are a number of isolated cliques of such institutions (servers) and parents (clients). An educational institution environment broadly involves three types of actors namely teacher, student and the environment. Interaction between these three actors generates voluminous data that can systematically be clustered to mine invaluable information. Hence, Educational Data Mining can be made into use to provide an effective student analysis based on their data and enhance the overall performance of the institution.

One of the essential objectives of any educational framework is to equip students with the learning and aptitudes expected to progress into successful professions inside a predetermined period. How successfully worldwide educational frameworks meet this objective is a noteworthy determinant of both monetary and social advancement.

The production release of the DIECAS platform is called Slate. Slate is developed as a "template" app. The deployment model of Slate is as follows; 1) an institute purchases a subscription to the Slate platform, 2) the institute is provided with a base variant of the Slate parent and teacher apps, 3) the institute can then, optionally, make changes themselves or purchase more features from the Slate features arsenal. At the end, the institute is left with two client apps that best fit their needs and financial status. The app can be deployed under the name of the institute as a standalone app powered by Slate.

2. DESIGN

Parent App Design

The parent side app consists of five views; the dashboard, scholastic, communication, progress++ and the settings view

Dashboard

The dashboard tab gives a bird's eye view over the curricular and co-curricular activities regarding the student as shown in figure 1.

This includes:

a: The homework widget - This widget displays the list of active homework that have to be done by the student. Each homework when clicked expands to show more detail like additional help in the form of images, videos or URLs that are added by the teacher at the time of creating the homework from his/her client app.

b: The events widget - This widget displays the upcoming

events at the institution, important details like day and date of the event is displayed directly while more information regarding the same is obtained by expanding each event.
 c: The results widget - This widget lists the results of the student in various examinations in reverse chronological order.

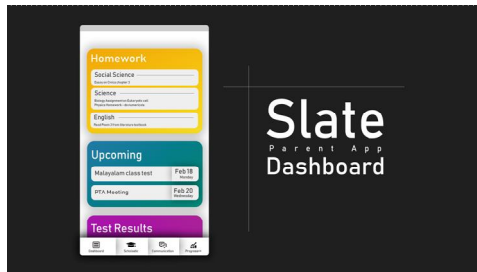


Figure 1: Mockup of the Dashboard Tab

Scholastic

The scholastic tab shown in figure 2 is the heart of the decentralized isolated educational content aggregation system[2]. Here the user is presented with all the educational content which includes images, videos and documents that are uploaded into the Firebase storage by the respective teachers. The data is well organized into different subjects and chapters within each subject. Google Firebase ensures high availability of the data and has flexible payment plans that allow more storage space on demand. On expanding each subject, a recycle rview is displayed which provides a feed-like view of all the educational content. Images and videos are displayed directly while URLs are opened in a default web browser app. YouTube player widget is embedded so that YouTube videos can be played within the app. Third party libraries are used for caching the images so that they need not be fetched repeatedly. The educational content of each School is isolated in the Firebase storage by placing them in different containers and restricting their access using Firebase rules. Hence the data of one Institution can inherently be never viewed by anyone who does not have its credentials. Additionally, the scholastic tab also contains quick launches to the various active homework that are to be done by the student.

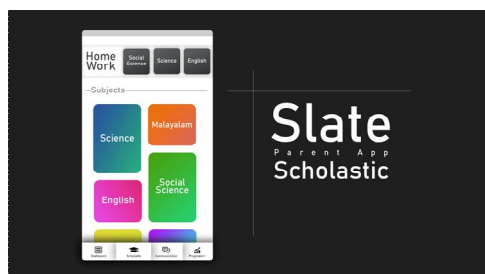


Figure 2: Mockup of the Scholastic Tab

Communication

One important feature of our platform is parent-teacher

Communication as shown in figure 3. A chat module is available that parents and teachers can both make use of to communicate with each other on matters regarding the student. This can be considered as a Digital Diary of sorts. The chat module is fairly modern and has all the essential features of a conventional messaging app like read receipts and time stamps. Both parties are notified of new chats as they arrive in real time. The parent can see the list of all teachers who teach the student in a scroll view above the chat module along with a thumbnail and details like name and the subject taught.



Figure 3: Mockup of the Dashboard Tab

Progress++

This is the part of DIECAS that leverages educational data mining[3] to provide parents with detailed statistics on their ward's performance in the various examinations. Progress++ uses various learning methods and calculates performance variations in different exams. This can point out some useful information like what areas of study the student should concentrate more on and how any new learning techniques have affected their results. Students can view their rankings as compared to their class and can engage in some healthy competition with themselves. The results of various examinations can also be used to determine the aptitude of the student and necessary steps may be taken to improve the areas that are lacking and provide aid to better the areas that the student excels at.

Settings

The settings tab allows the user to adjust various configuration options within the app. This tab also allows the user to sign out of an already logged in profile and log into another student profile. Special effort has gone into making the app readable to a very heterogeneous user group. The settings tab includes additional options to increase font size and change color schemes if the user wishes in order to make it more readable. Data saving options are available that reduces data bandwidth consumption by disabling media auto download. The cache size can be specified by the user so as to better fit his/her storage availability. Notification control options are also made available to the user.

Teacher App Design

The design of the teacher app is almost similar to that of the parent app with five tabs that facilitate the same functions.

Dashboard

In the teacher app, the dashboard tab contains almost the same widgets but the data is as seen from the teacher's perspective. The homework widget displays the list of active homework that the teacher has created. This widget also houses a button that lets the teacher create a homework directly as shown in figure 4.

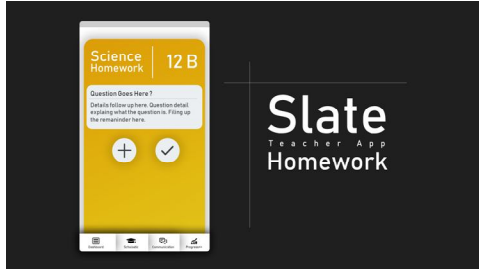


Figure 4: Mockup of the Teacher-side homework entry screen

The homework file consists of a brief Title, and a detailed description both of which are saved as text as shown in figure 5. The teacher may append additional rich data like images URLs, or multiple choice responses.



Figure 5: Teacher-side homework entry has rich text options along with multimedia upload

The Homework file is saved in the Firebase real-time database in an XML format as shown in figure 6. Hence, the text of the homework can be quickly fetched from the database and XML parsing may be done locally on the client app.

```
public static String parseXML(String source, String tag){
    int flag=0;
    int startIndex = source.indexOf("<"+tag+">") + ("<"+tag+">").length();
    int endIndex = source.indexOf("</"+tag+">");
    if (startIndex == -1 || endIndex == -1)
        return null;
    String out = source.substring(startIndex, endIndex);
    return out;
}
```

Figure 6: The ParseXML function is very simple and does not place any considerable load on the client machine. It also makes homework file storage in the real-time database very simple and effective

Scholastic

The scholastic tab in the teacher app is the upload portal for the content that we show up in the same tab of the parent app. The folders for each subject and the chapters in each are already made. The teacher only has to place the content into the appropriate folder. Educational content is shared between

different sections of the same class by default. However, a teacher can disable this option and share this content with a specific class only. The teacher can upload images, videos, documents or URLs.

Communication

The communications tab in the teacher app is very similar to the parent app. Instead of quick launches to chats of every parent of students, the teacher teaches, the bulk of parent contacts is grouped into classes and further sorted alphabetically. The teacher can initiate a conversation with any parent. The teacher can also send broadcast messages to a group of parents within a class. The broadcast messages are sent to every recipient and each recipient use the message as a personal individual message sent to themselves.

Progress++

The functions of Progress++ remain almost the same except that here the statistics is prepared on a class wise basis. EDM tools are utilized to study the class's performance as a whole in different examinations. This may point out useful information like the areas that majority of students are lacking in and the teacher can understand how well his/her class have understood certain concepts. The teacher can also check the efficiency of any new teaching technique that he/she might have been using by comparing the class performance before and after the use of such techniques.

Settings

The settings tab is exactly identical to that of the parent app.

3. FUTURE SCOPE

Currently, the platform design includes two Android apps[4][5] as clients connected to the Firebase server. This decision was made to facilitate rapid development[6][7]. However, it limits the number of users to only those teachers and parents who own and operate android device. To achieve universal usability, DIECAS must be ported onto the iOS platform and the Windows Desktop platform. The simplest way to do so would be to create two web clients that perform all the functions of the Android client discussed above. A responsive web page would mean seamless user experience mobile devices and desktops. Another future improvement would be to allow local instances of the DIECAS platform to be individually modified and updated to cape up to the very specific needs of that institution. A minimal level of concurrency would still have to be maintained with a stable global DIECAS release[6].

REFERENCES

1. Muhammad Nur Yasir Utomo, Adhistya Erna Permanasari, Eddy Tungadi, Irfan Syamsuddin, **Determining single tuition fee of higher education in Indonesia: A comparative analysis of data**

- mining classification algorithms**, New Media Studies (CONMEDIA) 2017 4th International Conference on, pp. 113-117, 2017.
2. Matte Crane & Jummy Lin ,University of Waterloo, Waterloo, ON, Canada (2017) **An Exploration of Serverless Architectures for Information Retrieval.**
<https://doi.org/10.1145/3121050.3121086>
 3. Ashish Dutt, Maizatul Akmar Ismail, and Tutut Herawan (2017) **A Systematic Review on Educational Data Mining**
 4. Gabriele Bavota, Mario Linares-Vasquez, Carlos Eduardo Bernal-Cardenas, Massimiliano Di Penta, Rocco Oliveto, and Denys Poshyvanyk(2015)- **The Impact of API Change- and Fault-Proneness on the User Ratings of Android Apps**
<https://doi.org/10.1109/TSE.2014.2367027>
 5. **Design and implementation of android base mobile app for an institute**
<https://ieeexplore.ieee.org/document/7755391>
 6. Suhas Holla, Mahima M. Katti, **Android Based Mobile Application Development And its Security** in International Journal of Computer Trend And Technology – 2012
 7. Jianye Liu, Jianka Yu, **Research On Development Of Android Applications** in Fourth International Conference on Intelligent System -2011