



International Journal of Research Publication and Reviews

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

Campus Sight

¹Lavina Mehta, ²Richa Sharma, ³Shreya Verma, ⁴Shruti Sahu

^{1,2,3,4}UG Student, Acropolis Institute of Technology and Research, Indore, India

ABSTRACT

There are numerous activities happening around us in our college, but we cannot get to know about every single one at the same time and we miss out most of them, these unpredictable factors always lead to lack of student approach and attention, although we are aware about it but there is no reliable way to know what is/was happening all at in real-time. To get rid of all these unwanted situations and problems “Campus Sight” can be a help.

Key-Words: Social Network; Real-Time updates; Firebase; Android; Firestore; Kotlin; NoSQL; Collections.

Introduction

A day in our Colleges is so dynamic, there are numerous activities, events, competitions, workshops, official updates, news, notices, announcements, and what not in a single day itself. But there is no reliable way to know what is/was happening where in real-time, we would miss out some or the other thing. Also, when there is Annual events and fests in the colleges, there is a tsunami of events, and it is hard to keep a track on what all is happening where. Not to mention, if we want to update the whole college about something interesting, or happening, or informative, or fun, there is no way that can work in real-time and have such a large network that can cover everyone present in the campus.

To get rid of all these unwanted situations and problems “Campus Sight” can help. Campus Sight is all about the official news, updates related to latest College Events, competitions, and much more. Served with some great utilities. It is a platform where all the students, faculties and staff members can connect, and can share everything from official news, notices, and updates to happening events, sports tournaments, fun activities, etc. that too in real-time, in the form of story updates. It covers everything from notices to Events to even information about college buses and transport all in one application in real-time. This application aims to provide users a medium where they can get all the Important, Interesting, and happening Updates from all over the college campus, all at one place.

Campus Sight is all about harnessing the powerful concepts of social media for productive uses and college benefits, it is something which is by the users, from the users and for the users.

There are numerous activities happening around us in our college, but it’s a tedious task to know about every single one at the same time and we miss out on most of them, these unpredictable factors always lead to lack of student approach and attention, although we are aware about it but there is no reliable way to know what is/was happening all at in real-time. The problem motivated us to craft a solution which can obtain a fresh approach and track all the happening things around us. That’s how we came up with an idea to have an app which can imitate the concept of social media but with the perspective of an institute. Our app will have social media-like features to track all the events, notices, and similar activities of the college campus all in real-time. From official notices to exciting events and necessities like transport, we have got all covered. To get rid of all the unwanted college related situations and problems “Campus Sight” can be a help.

Problem Formulation

Some of the problems we formulated were:

- A major problem students face is that some college events/competitions/auditions go unnoticed
- It’s difficult to Track what all events and competitions are happening in the campus
- There is no one stop place to get official notices and updates
- Missing out on important Competitions and Notices
- No system to Inform the whole college about an event
- Finding college bus with the desired pickup / drop location was difficult.

Literature Review

Information regarding many important events and notice about college could not reach to everyone at the time. These unpredictable factors always lead to lack of student approach and attention. Many people are interested in some of the activities going around but due to inappropriate information they could not reach them. Many a times students miss out on important official news and updates which results in great loss sometimes. Also visiting every event and notice board can be time consuming and a tedious job just to know what is happening where and is it important or not. Management authorities needs a follow up for an event or competition can get a way to track everything from anywhere in the campus. Students are not able decide which event seems interesting or which notice is for their use without visiting that place physically. Students do not get a short glimpse of every college activity and they have the fear of missing out on something. Finding the right college bus which can get you to your desired stop from college is a tough job. And there are lot of subtle issues, problems and situations like these, and our project can provide a solution for these problems as it is specifically developed to eliminate this.

Methodology

Our project is based on the concept of a social media / social network application specifically developed for an institute to capture some domains specific to that institution and use them to create a unique social experience through our resources. And with our modern-day internet and the ever- growing world of social media we are witnessing an era where information is spreading with an exceptional speed and in no time. This makes us realize how impactful real-time updates are and how they are impacting our social media experience. And to harness this functionality we are using Google's Fire Store database which is a functionality of the Firebase Database. Fire store have a lot of features to support the real-time scenarios and brings them to an application which uses it as their primary database. Other than firebase's fire store DB, we have also used firebase Authentication features to authenticate our users via their email address and apart from this we are also using firebase storage to store all the media which is displayed in user's feed. This makes Firebase an integral part of our project and we will be mentioning more about it in upcoming sections. To make the methodology part less complex we will be breaking it down into sections like, User interface, Posting Data, Data in user's feed, etc.

➤ *User Interface :*

The UI we have crafted here is very much inspired by the social media interface that we experience in our day-to-day life. These prominent UI elements makes our app looks very much familiar to the user and makes the user experience much more soothing and well crafted. Our user interface starts from a simple yet powerful login element which starts the campus sight experience for the user. We have provided an email login which is backed up by the firebase authentication which is indeed a cloud-based authentication, and the authentication data is saved in our firebase cloud storage. The authentication is not very complex it just checks for the user data and credentials are correct or not and then if everything is stable, it logs you in with all the app's privileges. The login then brings us to the home activity of our application. The UI at the Home page gives us an overall bird's eye view to the app where you see these three sections and from here you can head onto anywhere. We have placed the events section at our home screen as it is one of the most dynamic and powerful aspect of our application. In the events tab you see all the top 20 events, uploaded recently on the app. The feed is designed to grab user's attention via placing the events cards in a staggered grid layout and to see the most about a post in just one glance. Clicking on any post will open it up in the form of a story where you can see the post clearly in a short and engaging span of time. The events tab is dynamic and updates in real-time with the help of a snapshot listener. In the events tab only we have a hovering button which takes us to the camera activity from where we will create a post, by selecting or clicking a picture of an event and then uploading it to the database with the location tag as its's caption. Next to the event's tab we have the notices tab which displays all the notices. The UI of our notice feed is like the UI we have seen in all the prominent social media apps, it is more of a single post placed feed which focuses on one notice per row and shows the details and timings of it. Here also we have the floating action button and the activity which it takes you is very similar to the camera activity we had in our events tab. We can share the photo of the notice from our gallery as well as click and upload the photo all in Real time. Finally, we have the Transport tab where we display the bus routes in the form of a list with the main bus stops of the bus in the description tab of the bus. This enhances the UX and makes it faster to navigate for bus routes and user's preferred stops. We have also added a floating action button to this activity which links to a pdf which has the detailed route description of every bus. It is the official transport document provided by our institute. And lastly, we have the logout activity in the profile activity from where the user can logout.

➤ *Posting Data :*

The campus sight app allows users to post data just like any other social media application. We have added almost similar posting process both for the events and the notices features. The posting workflow uses firebase extensively as it is our database and is integrated in our app through android studio. We get the root of the database in our backend code and through that root we can run queries from our backend to the database and post and fetch data. The posting workflow starts when the user clicks on the camera floating action button, the button takes us to the posting activity. Now if the user selects the picture from the phone's local storage, it seamlessly loads the picture in the jpeg format, ready to be posted to the database. But if the user wants to click the photo in Real time through the phone camera, we launch an intent to open the phone camera and enables the user to click the picture. Once clicked and approved by the user, we store the image locally first and then load it into the app, this preserves the quality of the image, and a high-quality image is loaded in the ready to be posted. Next, we ask the user to enter the location tag as the caption of the image, basically the location where the photo was taken. Having these fields ready to be posted when the user hits the submit button, we run a query to our No SQL database. Through our query, we create a new post / notice document and fill the predefined fields with the values user uploaded, except the image posted by user. For the image part, we never directly post the jpeg image to our Real time database (Fire store DB) unlike we do for the rest of the data. For the images we take the help of firebase's cloud storage. Form the app, we upload the image on the cloud storage first, being saved in the cloud storage with its unique id, than we generate a URL to that image from the storage and post this URL as a string to the Real time database i.e., the fire store. In the fire store document's image field, we save this URL of the image. And when we unpack the data in the app to show in user's feed, we load the image through the URL string.

As all the data which is communicated is in text or string format, the posting process is fast and seamless and less costly, and we have a separate cloud storage where all the media data of the app is stored as a repository. Other than image rest all that we query to the database is the location tag, the time in milliseconds when the post was created and the user data who posted the post, just to keep a track and maintain the integrity of the app. All these fields make up a new post / notice document and is now saved in our fire store database, ready to be shown in the user's field as a new post.

➤ **Fetching Data (In Real time):**

Fetching data is less complex than posting data in the campus sight app thanks to firebase and it's Real time database (Fire store). The process to fetch the data is common through all the functionalities be it the events, notices, or transport. We fetch the data through a database query and model the fetched data in the form of a data class, and these classes helps us load the data directly to the user's feed. But before that, the fetching of data happens in real time, which means as soon as a user post something, it should be reflected in the latest feed, through our database. This Real time functionality is achieved by a feature called the snapshot listener from the fire store DB. We can think of the snapshot listener as a live instance of our app's database. The live instance is maintained in the snapshot listener and as soon as something changes in the database, the live instance changes, this notifies the snapshot listener that there might be a change which we maybe is supposed to be fetched and displayed in the user feed. This makes the snapshot listener active, and it runs the fetching query to the database, and if there were any changes, they are first formatted like a data class and then are inflated as UI elements through our backend. This way the user is never required to refresh his/her feed to see the latest post because as soon as database is changed via a posting functionality the live instance of the database changes and snapshot listener automatically refreshes and fetches the new data. This lets our application to have Real time changes in the use's feed and get the latest updates in no time.

An overall workflow of posting and fetching data is shown below:

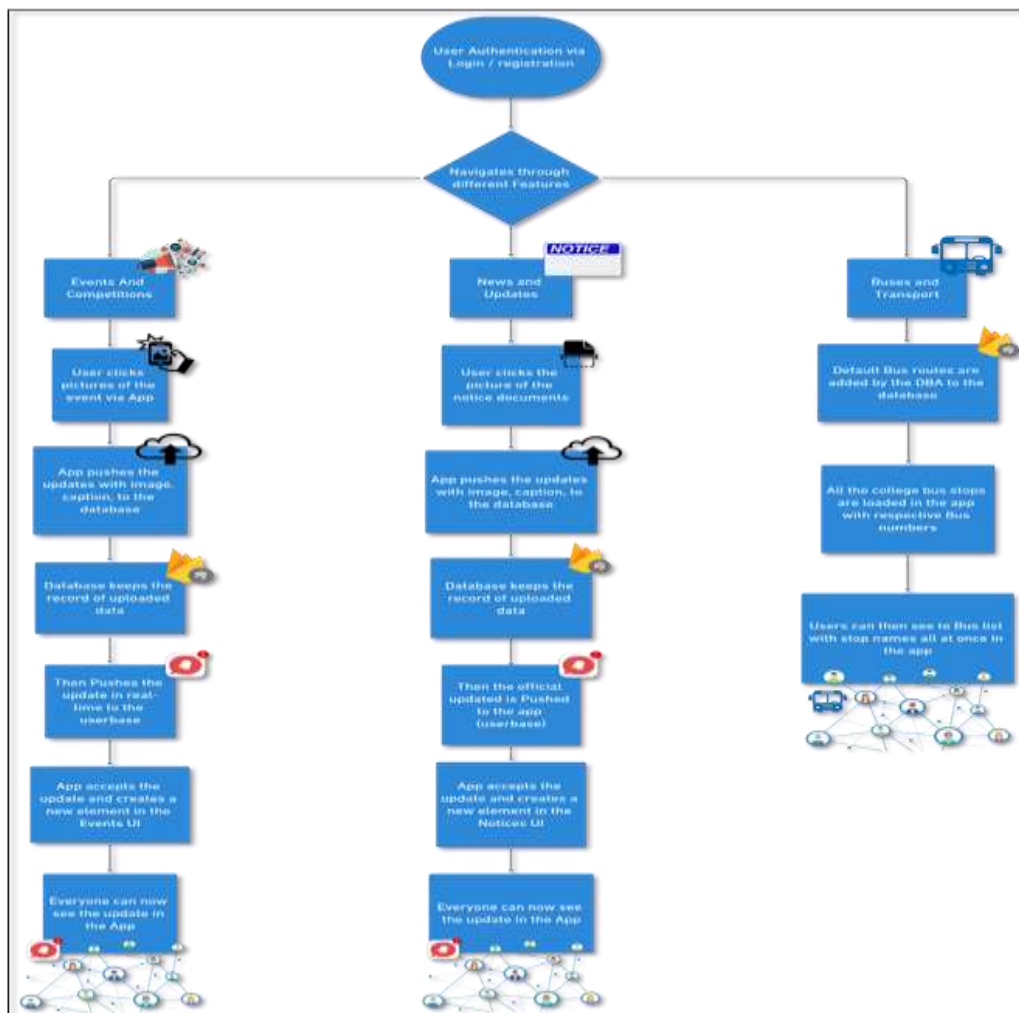


Figure: Overall Process workflow

Result Discussions

The aims of the project are as follows :

1. A Simple and efficient way to know about events

2. To bring every college update at one Platform
3. Showcasing College events
4. Providing Real time updates
5. A Platform to track Notices and Official News
6. Simplifying college bus routes
7. To make college navigation easier
8. Our app will be interactive, intuitive
9. Our app is very user friendly
10. Continuous improvement and engagement

Conclusion

Our Project, College Updates is an idea which will harness the powerful concepts of social network, and with the support of its users, it can do a lot of benefits to the college, and everyone associated with it. It is a platform which is by the users, from the users and for the users. This is to conclude that the project that we undertook was worked upon with a sincere effort. Most of the requirements have been fulfilled up to the mark and the requirements which have been remaining, can be completed with a short extension. Our project "College Updates" will fulfill all the requirements of the college and would be beneficial for the students and the college staff too.

Acknowledgment

The satisfaction and euphoria that accompany the successful completion of any task would be impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success. I have great pleasure in expressing my deep sense of gratitude to our guide Prof. Shivshankar Rajput. I take this opportunity to express my profound gratitude for his constant support and encouragement would also like to thank faculties of Department of Computer Science and Engineering, for his constant support and lastly, we would like to thank our institution for giving us this opportunity to learn even more apart from the curriculum.

References:

- [1.] Antonio Leive, "Kotlin for Android Developers", 21 March 2016.
- [2.] Google | Android Developers, "Android Documentations".
- [3.] Bill Philips, Chris Stewart, Brain Hardy and Kristin Marsicano, "Android-Programming," August 03,2015.
- [4.] Available at <http://en.wikipedia.org/wiki/Android>
- [5.] Introduction to Android: <http://developer.android.com/guide/index.html>.
- [6.] Android API: <http://developer.android.com/reference/packages.html>
- [7.] Fundamentals: <http://developer.android.com/guide/components/fundamentals.html>
- [8.] Android User Interfaces: <http://developer.android.com/guide/topics/ui/index.html>
- [9.] Layout: <http://developer.android.com/guide/topics/ui/declaring-layout.html>
- [10.] Common Tasks: <http://developer.android.com/guide/appendix/faq/commontasks.html>
- [11.] Android Training: <http://developer.android.com/training/index.html>.
- [12.] Android Developer's Blog: <http://android-developers.blogspot.com/>
- [13.] Developer FAQ: <http://developer.android.com/resources/faq/>
- [14.] Developer Forums: <http://developer.android.com/resources/community-groups.html>
- [15.] Android Developer's Group: <http://groups.google.com/group/android-developers?lnk=>