Chuwen Sun, Xiaoliang Chen, Patrick Cheng
CSE 5236
Professor Champion
12/5/2021

Class-based Forum & Meeting Final Report

Section 1 - Introduction

According to the National Center for Education Statistics (NCES), 19.7 million students enrolled in institutions across the United States in the autumn of 2020. The number of college students in the United States has increased dramatically throughout the years.[1]

Also, nowadays, various technologies have become not only a big portion of our lives but also cover a huge amount of our education activities as well. Since 2020, the Covid-19 pandemic made a huge impact on colleges and universities across the U.S., which made remote teaching and learning become more and more common. To help teaching staff and students to have the best teaching and learning experience, schools need to try every way. And technology is definitely one of them.

As a college student of the OSU, our school has provided a lot of decent technology that we can use daily, such as BuckID, OSU app, Carmen Canvas, and so on. However, a big part of our curriculum will include team working and helping peers. We tend to help and reach for help among our peers when resources are limited since professors and teaching assistants only have a limited time of office hours each week. So other than technologies the school provided to us, to increase study efficiency, we, the students, have been discovering technologies by ourselves to communicate, do research, arrange meetings, and so on. Applications and websites like discord, GroupMe, Google documents, and Microsoft Teams are all great examples of how students work together.

Although we already have a lot of different resources, a lot of students still want a platform that promotes a group-working and class-based system. This is because all the information provided by different people is not integrated and has not been put together, which means all this helpful information is spreading all over the internet randomly and it is hard to find. Also, since the increase of team working process in classes, students have a lot of needs for group meetings.

This is the root of our application design. We want to design an app that can provide students different discussion boards to discuss questions in a certain class, or they can use our app to start an online or in-person meeting in an easier process. All these features will be class-based, which helps students to be able to ask for help from fellow students in the same class.

Class-based Forum & Meeting is our app name. Class-based Forum & Meeting will provide users access to create their own account, create a class forum and ask questions(discussion) or host a meeting in the class forum. Forums and meeting rooms should both provide access to upload texts and files to help better communication. Also, location sharing

can be beneficial to notify fellow classmates if an in-person meeting will be held around the campus.

Section 2 - Design Process

The application is meant to be a forum-based discussion toll that can help college students study more efficiently. Students can post discussions about in-class topics or questions, or set up online or in-person meetings to study as a group. Once the application's target is clear, we aim for an object-oriented design process in Android Studios [2] to deliver the product. What the team did first is we define the nouns and the verbs from our purpose of the app and try to translate them into classes. Nouns like users, database, discussion, discussion board, notes, class, meeting are the most important classes that make up the app. These nouns are followed by verbs that can be implemented into methods to help with the functionality of the app, verbs consist of sign up, sign in, choose, enter, type, participate, upload, retrieve, start, join, end, share, upload and reply will bring all the classes and frameworks of the app together. We chose some of the most important nouns(ex. user, discussion, meeting, database) and verbs(ex. join, upload, start, reply) and first came up with an UML diagram to help us understand the responsibilities of the classes for the app.

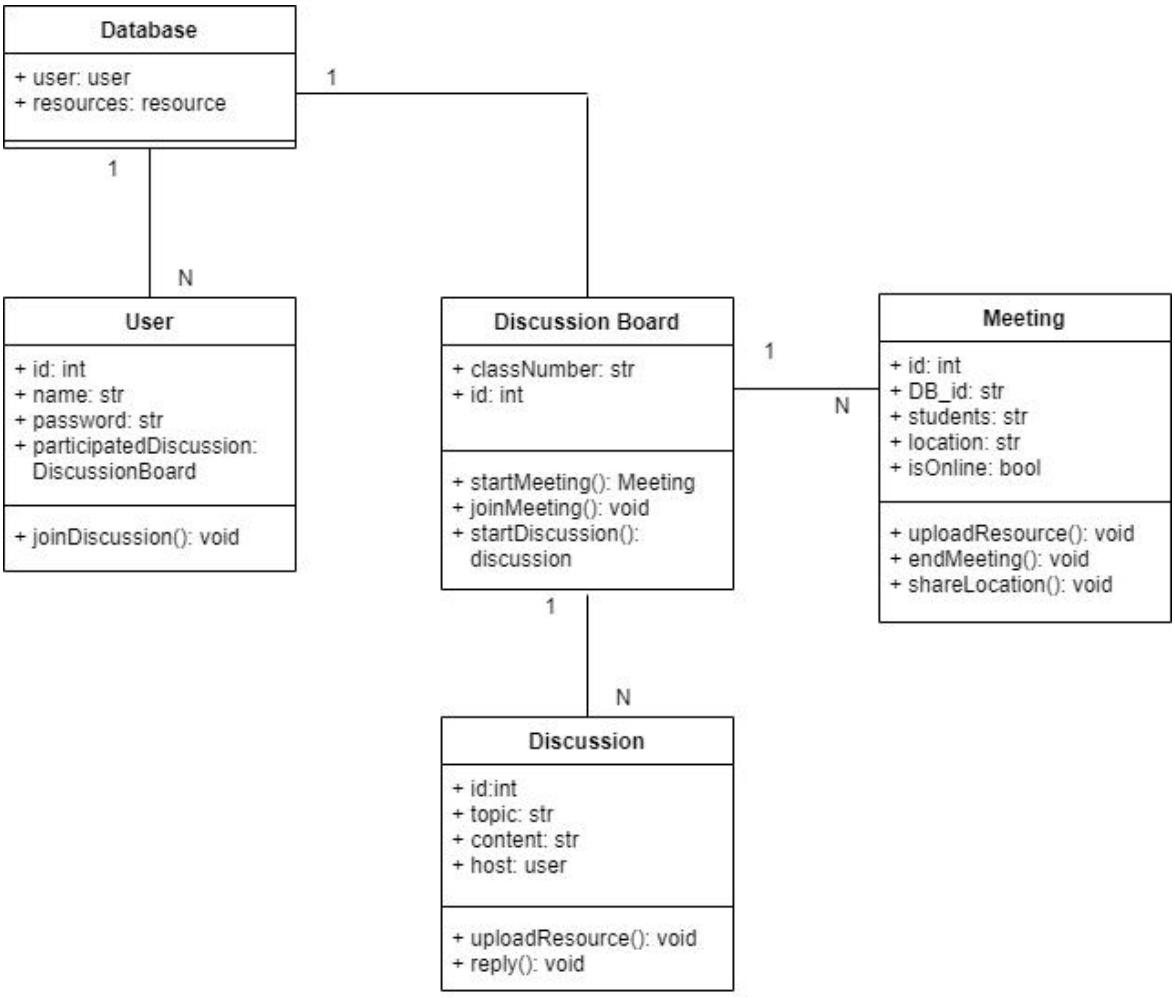

Figure 1: UML Class Diagram

After the team identifies the classes and each of its responsibilities, a simple use case diagram would be helpful to visually see what the app could do. For the standard functionalities, students using the app should be able to register an account and login with it. Users should also be able to enter discussion boards for different classes and either start a new discussion or reply to an existing discussion. Meetings are also a core functionality of the app. Users can launch in-person meetings or online meetings. In-person meetings are for students that feel comfortable studying in a group together. In order for the users to save time on where to meet, the app can share the location of the meeting via Google Maps [3] so that they can be brought directly to the meeting location without wandering around campus and wasting too much time on the way. The app also provides the option of online meeting for students that are too shy to meet with classmates that they might not know. Students can chat or share notes in the online meeting chat box for the online meeting just like Zoom [4] but without the video cam part. After we thought of what the app could do, we came up with a use case diagram.

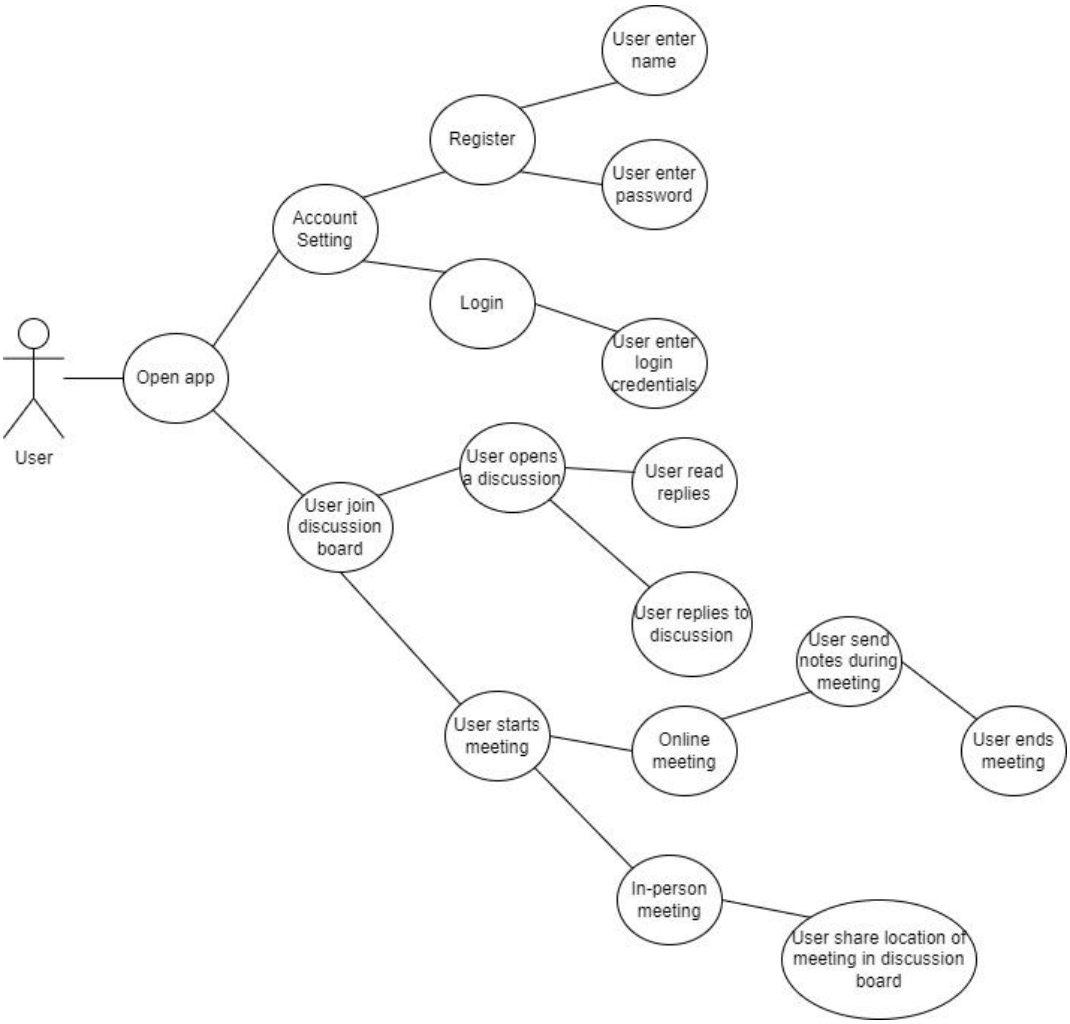

Figure 2: Use Case Diagram

Since this will be a forum-based discussion app, there will be a lot of data being stored in a database. We decided to use Firebase [5] realtime database to store all the information for our app. We first will be storing users' information in the database, that consists of the registered username, user email and password to login. We will also need to store data such as discussion posts and meetings. Each discussion post will be stored in a discussion board based on class. Each discussion post will have a unique id based on the time it was created and the user that created it. Each discussion post will also have the title of the discussion and the content which will be the description given by the host and replies from other students. For the meeting part, meetings will be stored as either an in-person or online meeting. The meeting will have its unique id based on the time it was created and the user that created it. In-person meetings will have a title and the location that the meeting is being hosted via Google Map [3]. Online meetings will also have a title, and the content that the users sent during the meeting. All discussion posts and meetings will all be stored under the discussion board section inside the realtime database.

Section 3 - Translating Design to Implementation

After we completed the design portion of our application, our team decided to use Android Studio to develop our application for the Android operating system [7]. In this project, we only use Java to implement all functionalities of this application[8]. First, we developed the user login page and registration page depending on the images we drew in the design portion. The user login feature is important because the user's information will be used in most of the views in our application. That's the reason why we decided to implement these two pages first. Firebase allows users to register an account using email and provides authentications when the user login [6]. For our registration page, we ask users to enter their email, user name, and password, and re-password. And if there is no account with the same email existing in the Firebase database, Firebase will generate a new user's UID in the authentication connected with the email, and user's information will be added to the Users table in the Firebase realtime database. For the login page, we used the Firebase Authentication to check if the user enters the correct email and password. We also added a password reset page to allow users to change their password by sending a password reset email.

After we finished the implementation of user login and registration functionality, our team started working on pages that allow users to add classes, discussions, and meetings. This process was not hard because it was very similar to putting user's information to Firebase realtime database. We just need to put the information of classes, discussions, and meetings to the corresponding tables in the database.

Once we finished the creation of classes, discussions, and meetings, we started looking at how to use recycler view [9] to list classes, discussions, and meetings. This requires us to pull the information from the database and then put them into the recycler view. The retrieval of information from the database was not difficult, but the use of the recycler view was difficult and it cost our team a lot of time on figuring out how to use the recycler view correctly. One biggest problem we met was that the recycler view did not show the information. We spent a lot of time on finding why the recycler view showed nothing even though we believed our retrieval of information should work. At last, we found that the problem was caused because the getter and setter methods in the Java class were not named correctly. After we solved this problem, the recycler view could successfully show the information.

With the recycler view working in our application, three of us separately worked on implementing discussion's use cases, online meeting's use cases, and in-person meeting's use cases. We made some changes to the layout of discussion, online meeting, and in-person meeting pages because the screen mockups we made for these pages in the design process were hard to implement. The implementation of these pages was straightforward since it was just the creation and retrieval of data, and we were already familiar with interacting with the database. Another change we made was the implementation of the map. We decided to use Google Map [3] in our application to let the user get the current location and find the location of an in-person meeting in the design process. However, we tried to get the user's permission to get the current location but failed. Due to time constraints, our team decided to drop it and only used Google Maps to allow the user to find the location of an in-person meeting.

At last, we added some new functionalities that we did not have in the design progress to make the application more complete. We added functionalities that allow the user to change their username after logging in, allow users to delete the account when they no longer need it to secure their information, and notify users when they do not have network connections.

Section 4 - Suggested Changes and Improvements

When our team was designing and implementing the application, we were mainly focusing on the requirements and functionalities of the app and didn't go deep into the non-functional requirements of the app. Non-functional requirements could make the app more interesting for the users and may attract more people to choose our app over others, thus it is one of the changes that we want in the app. We thought about what kind of app we would use if we have multiple similar functionality apps being compared to each other, and we prefer apps that are easier to navigate within the app, having better response time and better security while providing the most functions.

Though not many layout xmls are in our app, users still might have a hard time finding the function they want in our app. We didn't spend a lot of time working on the layout of our app, but I think it could be improved if we had spent more time on it and made it more interesting to use. When we implemented the app, our target goal was to make the functional requirements usable in the first place. So some code in certain classes and activities might not be optimized when it was implemented.

On the side that isn't visible to the users, there were also a couple things that we could have done better. First of all is database management and how all the data is stored and sorted. For our database, all the data including discussion post data and meeting data is under the big category of discussion board and it can be very hard to find a certain uid if there are a lot of data entries in the realtime database. I felt like we could have done more sub categorizing and sorted each data under a divider that only has the data from a specific date, so that if we are debugging and looking for something in the database we would have a much easier time finding it. We could also utilize fragments in the Android Studio [2] development environment. There were a lot of places in the app where we wrote repeat code and we could have used fragments to optimize the performance of the app.

Our app has the feature to start and join online meetings, users can chat or send notes in the chat box of the meeting. I felt like we could have made this part of the app better since it can only send text data. We could have implemented features such as uploading images or connecting a video through the user's front camera to make the online meeting section of the app more interesting. We also could have used another external service like Zoom [4] for the online meeting section of our app.

References

[1] How Many College Students Are in the U.S., Best Colleges, https://www.bestcolleges.com/blog/how-many-college-students-in-the-us/

[2]Android Open Source Project, Android, http://developers.android.com

[3]Maps SDK for Android Quickstart, Android, https://developers.google.com/maps/documentation/android-sdk/start

[4]Zoom: Video Conferencing, Zoom, https://zoom.us/

[5] Firebase Documentation, Firebase, https://firebase.google.com/docs/database/android/start

[6] Manage Users in Firebase, Firebase, https://firebase.google.com/docs/auth/android/manage-users

[7] Android | The platform pushing what's possible, Android, https://www.android.com/

[8] Java API, https://docs.oracle.com/javase/7/docs/api/

[9] Create Dynamic List with Recycler View, https://developer.android.com/guide/topics/ui/layout/recyclerview