INF 551 Project Final Report

Hyun Jun Choi {choi797 8236-7159-17}
Wanjin Li {wanjinli 7958-7294-86}
Taoran Ju {taoranju 9791-3943-87}

1. Overview

Our idea is developing an Android application called **DiSCover**. It is designed for USC community includes students, faculty and alumni, to recommend good resources in USC campus and share their extraordinary experiences to each other. Students can post photos, pictures, chose category and rate the item and write several texts to describe it, and other students can ask question and chat with the author through the app. We use Firebase as the database to store all the data and resource, and implemented the app via MVC structure.

2. Architecture

2.1. Components

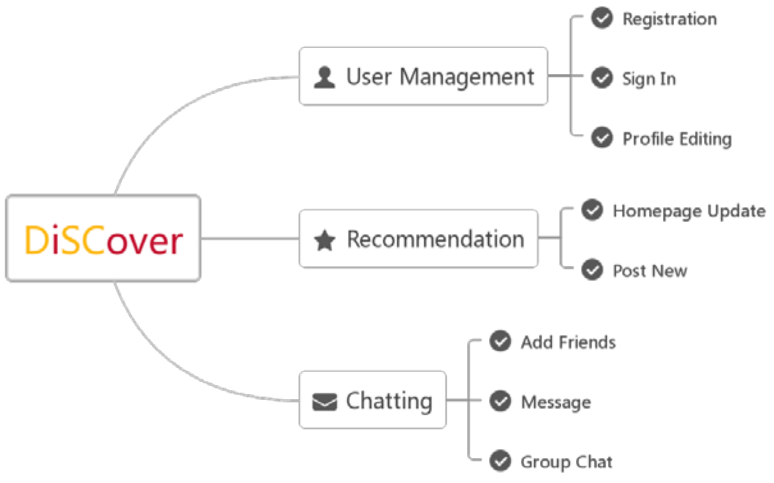


Fig. 1 Major components in DiSCover

DiSCover has 3 major components: user management, recommendation post and chatting. User management component mainly implements the user registration, sign in and profile editing function. Recommendation post component supports user to submit new post includes many details, and can update all the recommendation posts in homepage in real-time. Chatting component realizes the communication between users, contains adding friends, message send/receive and group chatting.

2.2. User Interface

We use Java to develop an Android app, and choose API in level 26 which is the newest edition. When design the UI, we use the representative USC style. We also import some external library like Picasso, Butterknife and Firebaseui to make the user interface more friendly and good-looking.

2.3. Firebase

2.3.1. Database

The realtime Database in Firebase is a NoSQL, JSON and cloud hosted database, we use it to store the user information and post records. Within the Users group, it generates a unique UID for every user when they register at the first time. For each user, we store the url of user's profile image in Storage and the username.

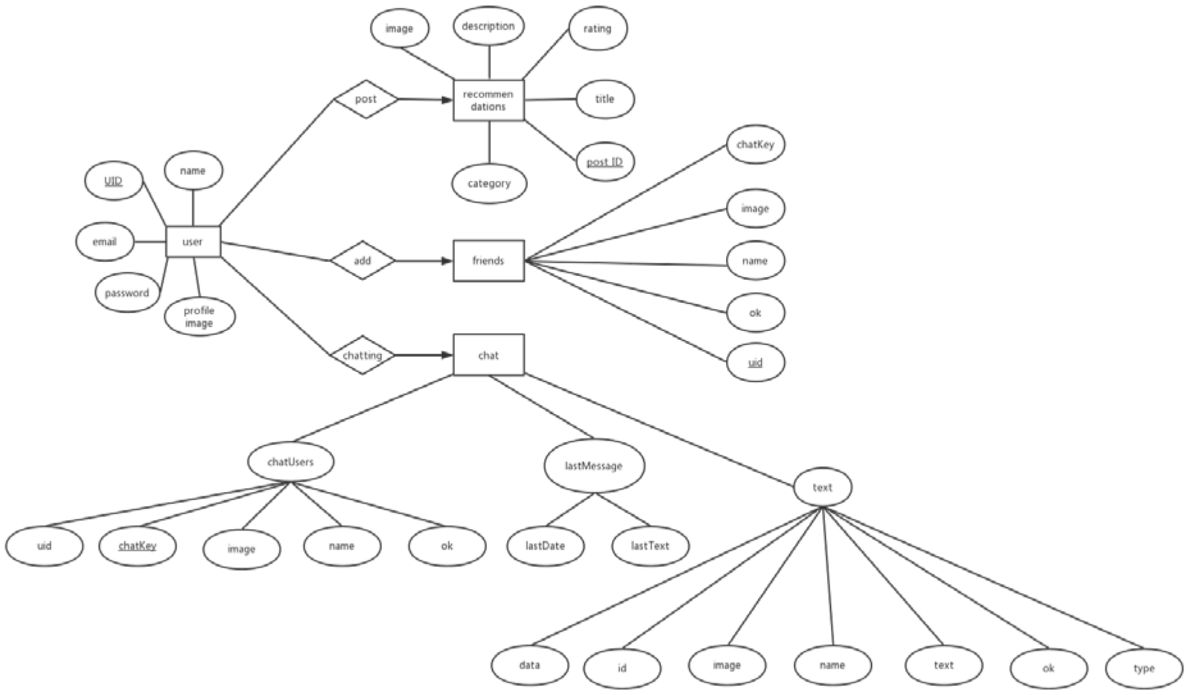
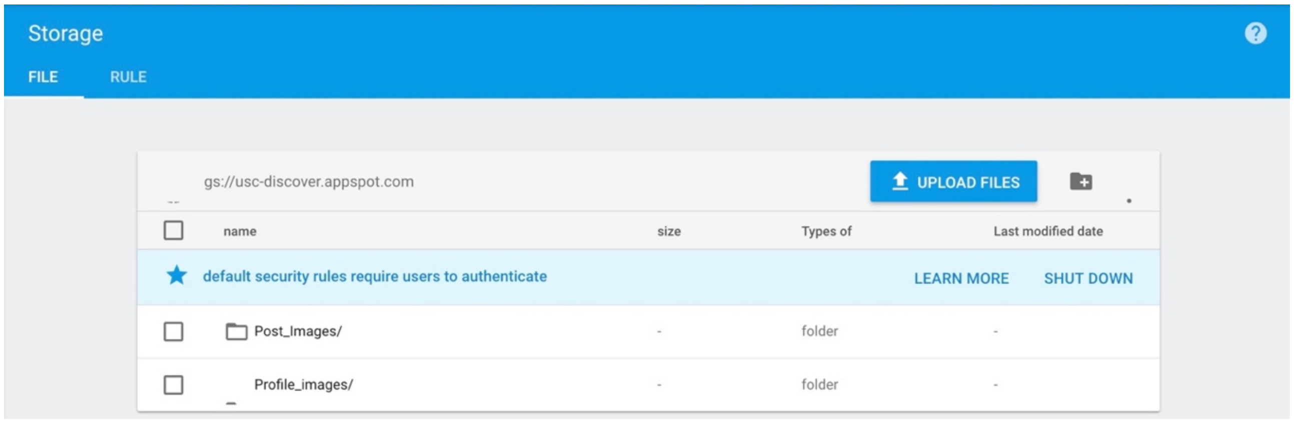


Fig. 2 ER diagram

2.3.2. Storage

Storage in firebase is designed for keeping media resources like picture, audio and videos. We use storage to store all the image files. Profile_images file folder contains the photos of user profile and Post_Images file folder contains the pictures user upload when post recommendation. Each image in Storage has a unique URL address which can be store in the Database later to infer to this file.


Storage

FILE RULE

gs://usc-discover.appspot.com UPLOAD FILES

	name	size	Types of	Last modified date
	default security rules require users to authenticate		LEARN MORE	SHUT DOWN
	Post_Images/	-	folder	-
	Profile_images/	-	folder	-

Fig. 3 Storage in Firebase

3. Implementation

3.1. Login

In this page, user can log in with their own account, which includes email and password. After entering email and password, user can click “LOGIN” button, and “Starting sign in …” will appear on screen. After login, application will send user’s email and password to firebase and firebase will check whether the user’s information exists in the database. If the information is wrong, the screen will display “Error Login”. If user has set up profile information, the interface will be switched to main interface, which user can see information posted by other users. What’s more, user can also log in by google account. After clicking the “google sign in” button, user chooses his or her google account to log in.

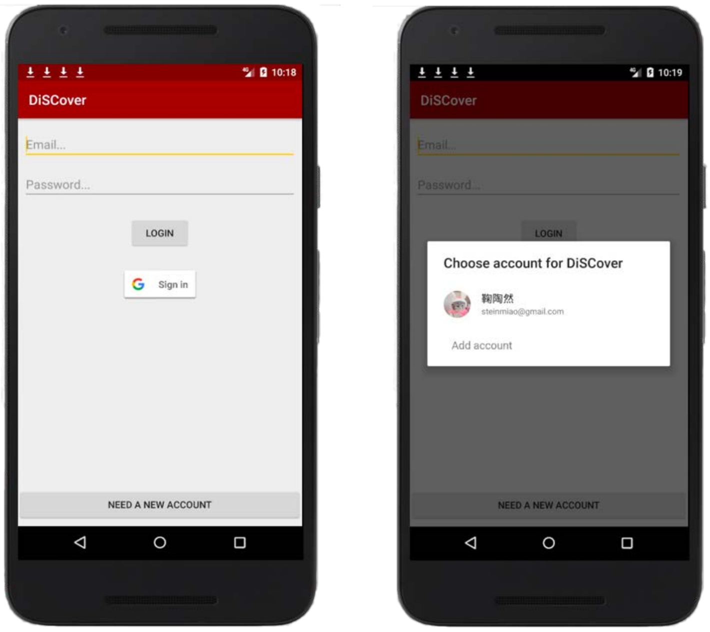


Fig. 4 Login page / Google login page

3.2. Set Up

If user has not set up, after registering account or logging in, interface will be switched to set up interface. People can click the profile image, and then people can choose photos from gallery. After choosing photo, user can adjust the size and direction of photo, and click the "CROP" button. Then user can enter his or her display name. Then click "FINISH SETUP", and the user will see the main interface. When user wants to change the profile photo, user can click the button on the top right of the main interface, and click "settings" button. Then, user can set up again. After setting up, the application will send image and name to the firebase, the information of this user will be updated in firebase.

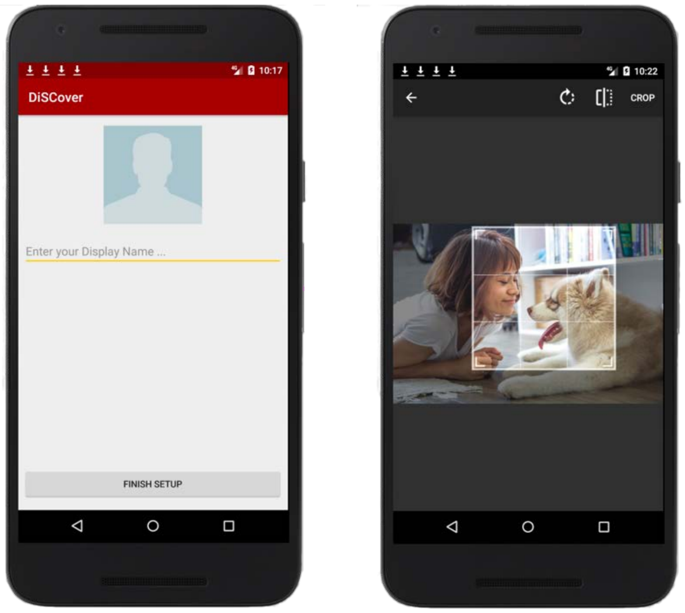
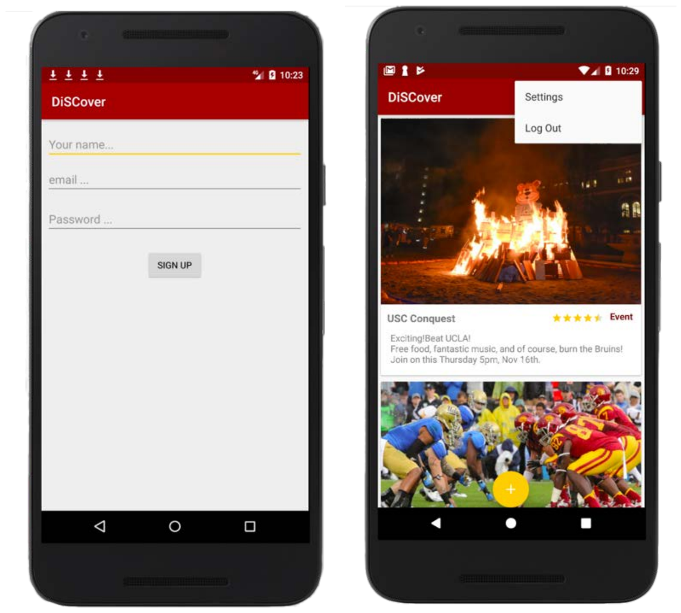


Fig. 5 Set up page / Choose photo for profile

3.3. Register

If user doesn't have account in our application, user can click "NEED A NEW ACCOUNT" on the log in page. And then user can enter information according to hints. Then, click "SIGN UP" button. The application will send user information to the firebase, and the database will store the user's name, email, and create a unique uid for him or her.

After registering and setting up, user can enter into main page. If user wants to log out, just click the button on the top right. Then, click "Log Out" and user will go back to sign in page. In the backend, firebase will sign out too.



Exciting!Beat UCLA!
Free food, fantastic music, and of course, burn the Bruins!
Join on this Thursday 5pm, Nov 16th.

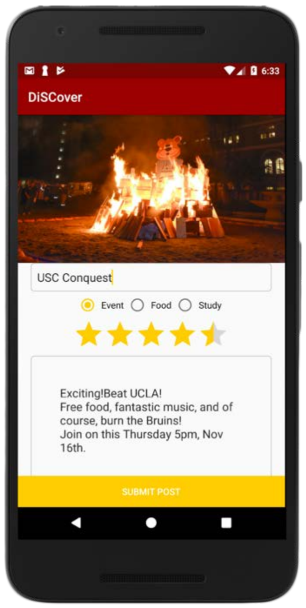
Fig. 6 Register page / Settings and log out function

3.4. Post Recommendation

3.4.1. Post Contents

In the post page, users are able to create and post their own recommendation.

For each piece of post, user can attach a picture, add a title, choose the category of the post, rating for the recommendation issue and write some text description about recommendation. Specifically, we define three categories of post: food, event and study. By using a RadioGroup widget to store them, user can only select one category of three options. And for rating function, the full score is 5 stars, and step size is 0.5.



USC Conquest

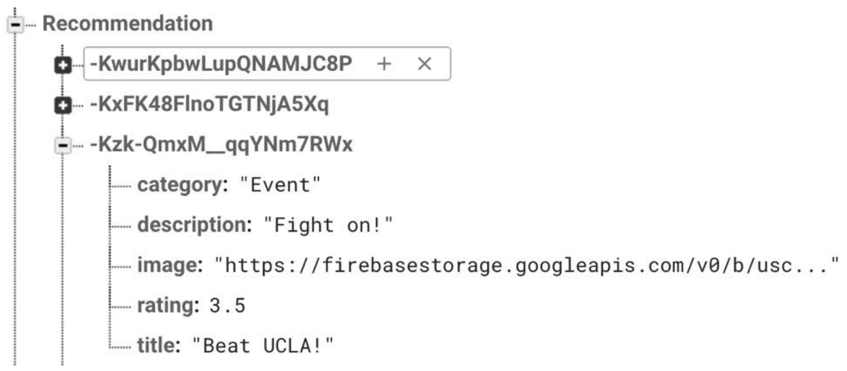
[✓ Event] Food Study

Exciting!Beat UCLA!
Free food, fantastic music, and of course, burn the Bruins!
Join on this Thursday 5pm, Nov 16th.

Fig. 7 Post Recommendation Page

Then by click the “Submit Post” button, this piece of recommendation can be uploaded to the list of all posts, and page will jump back to the home page where user can see this post and all other posts.

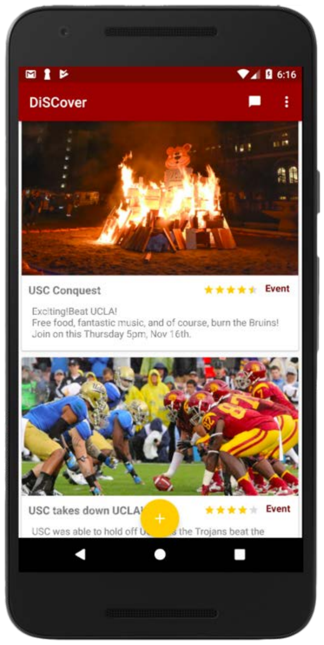
Once the bottom has been clicked, all the contents will be upload to Firebase automatically. It will generate a unique post ID below the branch of “Recommendation”. Inside the branch of this post ID, it stores the keys and values of title, description, category, rating and the url of image in storage (where the image has been upload to the storage in Firebase at the same time).



```
Recommendation
    -KwurKpbwLupQNAMJC8P
    -KxFK48FlnoTGTNjA5Xq
    -Kzk-QmxM__qqYNm7RWx
        category: "Event"
        description: "Fight on!"
        image: "https://firebasestorage.googleapis.com/v0/b/usc..."
        rating: 3.5
        title: "Beat UCLA!"
```

Fig. 8 Each post record in database

3.4.2. Homepage display



6:16
DiSCover
USC Conquest
Event
Exciting!Beat UCLA!
Free food, fantastic music, and of course, burn the Bruins!
Join on this Thursday 5pm, Nov 16th.
USC takes down UCLA
Event
USC was able to hold off U the Trojans beat the

Fig. 9 Homepage

The homepage is also the first page for user who has complete sign in/sign up or has already signed in. Once jump to this page, the app will require all the branches of “Recommendation” in Firebase and user is able to see all the recommendations that display by post time. In each piece of recommendation, it shows the title of contents, category of content, rating of this subject, text description and the picture. If any new recommendation has been upload, user is able to see the update in real time. Specifically, user can rerate the items on homepage by simply drag the starbar, the new rating score can update to Firebase synchronously.

In this page, user can also click the '+' button in the bottom of the page to enter the post new recommendation page, click the message button to entry chatting page or click menu to edit user's profile or log out the account.

3.5. Chatting

3.5.1. Search friends

You need to use friend's email address to find her/him. For example, if you want to find test7@mail.com, just write it on the yellow line and click search button. By using the email address, it can find your friends in firebase.

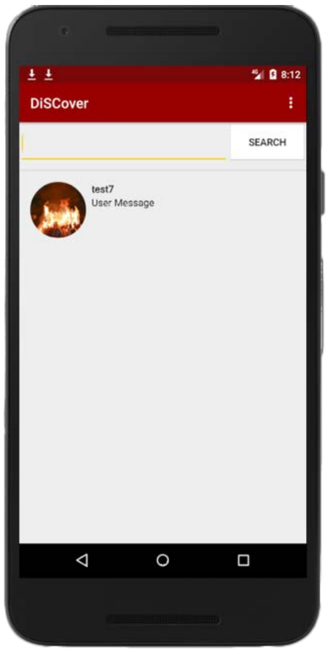
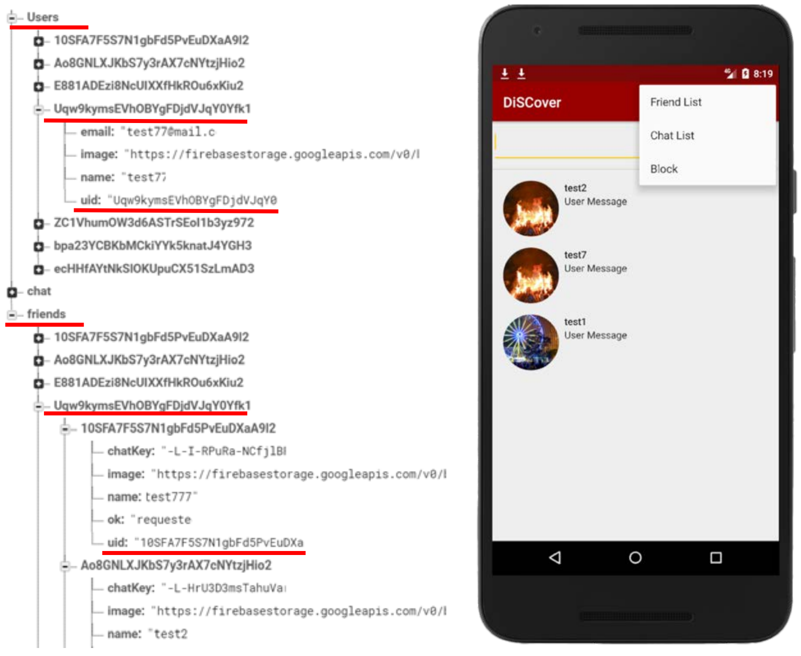


Fig. 10 search friend

3.5.2. Friend list

If you already registered your friends, when you click the chatting function, it shows your friends. And when you want to see your friend list, just click the 3dot buttons. And then you can see the window where there are three kinds of selections. When you click friend list, you can check your friend list. When you click your friend, your friend's uid, chatKey, image, name, and ok are stored under your uid in friends field.

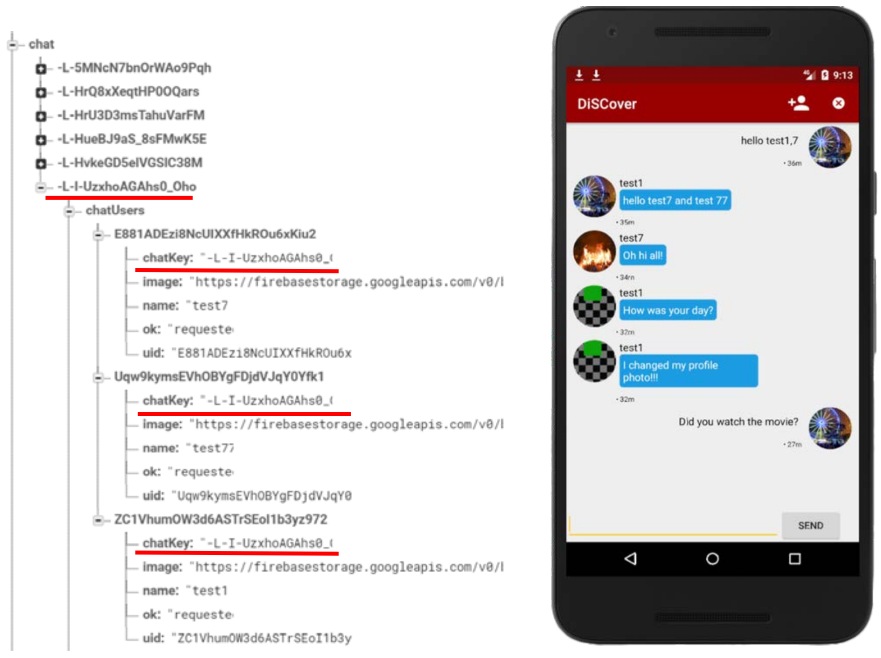


```
Users
  10SFA7F5S7N1gbFd5PvEuDXaA9l2
  Ao8GNLXJKbS7y3rAX7cNYtzjHio2
  E881ADEzi8NcUIXXfHkROu6xKiu2
  Uqw9kymsEVhOBYgFDjdVJqY0Yfk1
    email: "test77@mail.c
    image: "https://firebasestorage.googleapis.com/v0/l
    name: "test77
    uid: "Uqw9kymsEVhOBYgFDjdVJqY0
  ZC1VhumOW3d6ASTrSEol1b3yz972
  bpa23YCBKbMCkiYYk5knatJ4YGH3
  ecHHfAYtNkSlOKUpuCX51SzLmAD3
chat
friends
  10SFA7F5S7N1gbFd5PvEuDXaA9l2
  Ao8GNLXJKbS7y3rAX7cNYtzjHio2
  E881ADEzi8NcUIXXfHkROu6xKiu2
  Uqw9kymsEVhOBYgFDjdVJqY0Yfk1
    10SFA7F5S7N1gbFd5PvEuDXaA9l2
      chatKey: "-L-I-RPuRa-NCfjlBI
      image: "https://firebasestorage.googleapis.com/v0/l
      name: test777"
      ok: "requeste
      uid: "10SFA7F5S7N1gbFd5PvEuDXa
    Ao8GNLXJKbS7y3rAX7cNYtzjHio2
      chatKey: "-L-HrU3D3msTahuVa
      image: "https://firebasestorage.googleapis.com/v0/l
      name: "test2
```

Fig. 11 registering friends and friend list

3.5.3. Chatting with friends

After you add your friends to your friend list, you and your friend can enjoy chatting. When you make new chatting group, each chatting group has its unique chatKey. People who participate in same chatting group have the same chatKey.



```
chat
  -L-5MNcN7bnOrWAo9Pqh
  -L-HrQ8xXeqtHP0OQars
  -L-HrU3D3msTahuVarFM
  -L-HueBJ9aS_8sFMwK5E
  -L-HvkeGD5elVGSIC38M
  -L-I-UzxhoAGAhs0_Oho
    chatUsers
      E881ADEzi8NcUIXXfHkROu6xKiu2
        chatKey: "-L-I-UzxhoAGAhs0_(
        image: "https://firebasestorage.googleapis.com/v0/l
        name: "test7
        ok: "requeste
        uid: "E881ADEzi8NcUIXXfHkROu6x
      Uqw9kymsEVhOBYgFDjdVJqY0Yfk1
        chatKey: "-L-I-UzxhoAGAhs0_(
        image: "https://firebasestorage.googleapis.com/v0/l
        name: "test77
        ok: "requeste
        uid: "Uqw9kymsEVhOBYgFDjdVJqY0
      ZC1VhumOW3d6ASTrSEol1b3yz972
        chatKey: "-L-I-UzxhoAGAhs0_(
        image: "https://firebasestorage.googleapis.com/v0/l
        name: "test1
        ok: "requeste
        uid: "ZC1VhumOW3d6ASTrSEoI1b3y
```

hello test1,7 · 36m

test1: hello test7 and test 77 · 35m

test7: Oh hi all! · 34m

test1: How was your day? · 32m

test1: I changed my profile photo!!! · 32m

Did you watch the movie? · 27m

Fig. 12 chatting

3.5.4. Chat list

When you want to check chat list, just click chat list in the 3dot button (fig. 11). By using your uid and chatKey, it shows you what chatting groups you have.

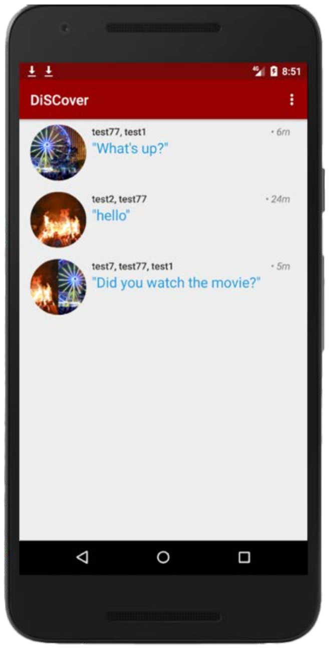
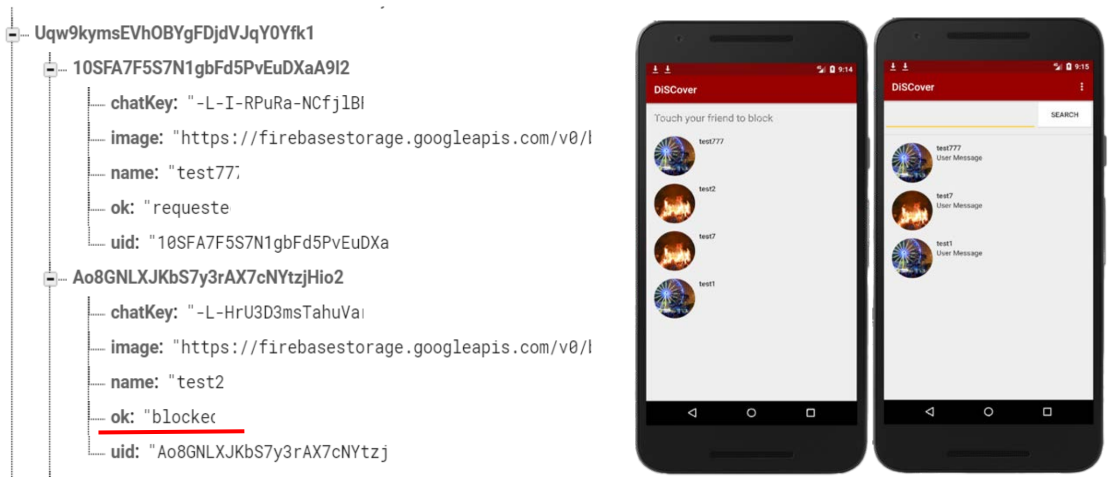


Fig. 13 chat list

3.5.5. Block

If you want to block some friends, just click the block button in the 3dot button (fig. 11). For example, if test77@mail.com user blocks test2@mail.com user, the ok field stores "blocked". Therefore, some friends who have "block" value cannot chat with you.



```
Uqw9kymsEVhOBYgFDjdVJqY0Yfk1
  10SFA7F5S7N1gbFd5PvEuDXaA9l2
    chatKey: "-L-I-RPuRa-NCfj1BI
    image: "https://firebasestorage.googleapis.com/v0/l
    name: "test77
    ok: "requeste
    uid: "10SFA7F5S7N1gbFd5PvEuDXa
  Ao8GNLXJKbS7y3rAX7cNYtzjHio2
    chatKey: "-L-HrU3D3msTahuVa
    image: "https://firebasestorage.googleapis.com/v0/l
    name: "test2
    ok: "blocke
    uid: "Ao8GNLXJKbS7y3rAX7cNYtzj
```

Fig. 14 block

4. Code Documentation

4.1. Sign In / Sign Up

In the LoginAcitivity class, we have six main functions. The first one is onCreate(), which mainly check the activity of buttons. signIn() and onActivityResult() are functions realizing google's log in. firebaseAuthWithGoogle() can get an IDtoken from the GoogleSignInAccount object exchange it for a Firebase credential, and authenticate with Firebase using the Firebase credential after a user successfully signs in. checkLogin() can check whether user enters complete information. checkUserExist() is used to check if user's information exists in database, if user exists and has set up profile, the page will be switched to main page, or set up page if user never sets up profile.

In the RegisterActivity class, we have three main functions. The first one is onCreate(), which mainly check the activity of buttons. onActivityResult() is used to obtain and adjust image from gallery. startSetupAccount() can send the updated profile image and display name to firebase. When user sets up profile successfully, the page will be switched to main page.

In the RegisterActivity class, we have two main functions. The first one is onCreate(), which mainly check the activity of buttons. startRegister() can get user's information and send it to firebase. When registering account successfully, the page will be switched to set up page if user never sets up profile, or main page.

4.2. Post Recommendation

4.2.1. Post Contents

Post recommendation function is implemented by class PostActivity that extends AppCompatActivity with three functions. Firstly, onCreate() function connects with the layout file activity_post.xml and obtains contents from the widgets like EditText() and RatingBar(). For obtain the image from local image gallery, set the request code for the startActivityForResult() method inside widget ImageButton's OnClickListener as GALLERY_REQUEST. Override the protect method startActivityForResult() that when it receive the request codes both GALLERY_REQUEST and RESULT_OK, it will generate the url for picture and set to the ImageButton. It also.

Once the "Submit Post" button is clicked, the button's OnClickListener in onCareate() function will start the startPosting() function, which is the second function in the PostActivity class. Inside startPosting(), generate a variable of StorageReference to upload image into the "Post_Images" folder in Storage. Then generate another variable of DatabaseReference, use push() method to create a unique random post ID for it, and set the values below their corresponding keys. Most values got from widgets can be transfer as String type except rating content is Float type and image is the url in Storage.

4.2.2. Homepage

The main function for homepage is displaying and updating all the recommendation posts, and it also is the entrance to other pages. For each individual recommendation post in the homepage, its layout is written in layout file post_row.xml. Inside the Recommendation class, all of its private attribute names are same with the key names in the “Recommendation” branch in database. Then it generates the Getter and Setter for each of these attributes.

Homepage is mainly implemented in MainActivity class. For its layout, RecyclerView widget can be used to list all the posts from database. We generate a class that extends RecyclerView.ViewHolder called RecommendationViewHolder, which is used to set the corresponding contents to the widgets. In the onStart() function, create an adapter of FirebaseRecyclerAdapter which can download the contents from Firebase one by one and send to the ViewHolder. We also set a setOnRatingBarChangeListener for RatingBar in the ViewHolder, so once the user rerates the score, the database can update the new score below this piece of recommendation. Finally, using setAdapter() method the RecyclerView in homepage can list all the posts from Firebase in real-time. Here we use the Picasso library (vision is picasso:2.5.2) to get the image from url.

4.3. Chatting

4.3.1. Architecture Design

To build real time chat system, I used google firebase server to store user and chat data. By using firebase server, I don’t need to request response to server from android client. Android application receives data stored in firebase server dynamically. The dynamic data is sorted by user id, and shows to related user only. So, the chatting architecture is much simpler than using socket server.

4.3.2. Chat Design

To build chat function in this application, I decided to show two different lists to users. First one is Friend list of user, and second one is existing chat list. User can make a new chat group by touching a friend from friend list, or join an existing chat group by touching a chat group in chat list.

4.3.2.1. Friend Activity

In friend activity, user can add friend freely by searching email address. In the activity, user can start new chat, or block his/her own friend. Once Activity started, showFriendList method will be called automatically to show friend list.

(1) Firebase Server Handling

To show friend list, I made new firebase data child “friends”. Each friend field will be updated under key value which is user ID. By this, user can receive own friend list from server directly, and no need to handle data in client.

(2) Menu List

- **Action Friend**

  Show friend list in a recycler view

- **Action Chat**

  Show Chat list in a recycler view

(3) Add Friend to Server Data

To add friend to serve, user should type email in edit text. And then, searchPeople method will be called to find user from the database. Value event listener will respond every user in firebase user data, I saved user information to array list first. If there is no matched email in array list, toast will show to let user know that there is no user with written email. If there is same email in array list, checkFriend method will be called to check whether that written email is already added to friend list of user. If not, showSearchUser method will be called to show searched user profile. To show user image, Glide library is used. Once user touch Add to fiend button, showFriendList method will be called again.

(4) Image library comparison

For this project, I’ve used Picasso and Glide library to handle image from firebase server. And I’ve found Glide is much faster to show image at a view. Even though Glide requires more memory, I’d better use Glide library to handle image from next project.

(5) Show Friend List Method

This method queries all the friend field from firebase sorted by user's own ID. Queried data will be added to friend field array list through Friend field class. And swapData from friend recycler view adapter will be called to refresh recycler view item list. By building MVC, I could handle firebase data very simply. ShowFriendList method maybe called repeatedly by adding friend function, the method clears friendDataArrayList at first not to show duplicated list in recycler view. If user touches a friend in a list, application starts a chat activity related with selected friend.

(6) Show Chat List Method

The main function is similar with showFriendList method. Biggest difference is that chat data is stored and passed to adapter as list. This method queries all the chat list data from firebase. Each chat data can have multiple user data, since this application provides group chat function. Each user information composed of name, id, and image are added to separated list. ChatListData Model is called to store received data from firebase, and this list will be handled in adapter to show item. This function will be explained in adapter sector. If user touches a chat list, application starts a chat activity related with selected chat group.

4.3.2.2. Friend Adapter

Handle received data from firebase to show as a list in a recycler view. Glide library is used to show user image and circleCropTransform function applied to show profile image in a circle form. swapData method replaces existing list with new data and call notifyDataSetChanged in adapter, so that recycler view item can be refreshed. OnClickMethod passes adapterPosition to Friend Activity so that I can get proper friend field from array list by adapterPosition.

4.3.2.3. Chat List Adapter

Chat list adapater handles list data, since each chat list has different number of users. By using switch, Chat list adapter shows different view in an item. Glide library is used to get bitmap from firebase image. Once Bitmap image is ready, combine image methods will be called to combine multiple images. This Glide library is used to show user image and circleCropTransform function is applied to show profile image in a circle form. When swapData method is called from Friend activity, swapData method replaces existing list with new data and call notifyDataSetChanged in adapter, so that recycler view item can be refreshed.

OnClick method passes adapterPosition to Friend Activity so that I can get proper chat data from array list by this adapterPosition.

4.3.2.4. Block Activity

Block Activity shows current friend list in a recycler view. If user touches a friend in list, onClick method will be called to update firebase data child “ok” to set value “blocked”. When inviting friend to chat group or chatting with friend, application checks weather user gets blocked by friend or not.

4.3.2.5. Block Adapter

Block Adapter is built to handle data in recycler view in Block Activity. onClick Method passes proper position to Block Activity, so that Block Activity can handle proper friend field from data.

4.3.2.6. Chat Activity

When ChatActivity is called, chat key was passed from previous activity. Chat Activity receives related chat data from Firebase with chat key.

(1) Started from Friend List

Chat Activity will check user and friend field have been updated to related firebase database. If not, Chat Activity will update related user information to firebase, so that this chat group can be showed in a chat list.

Firebase addChildEventListener will received added data dynamically from firebase database and update recycler view. So, the list will look like real time chat view.

When send button clicked, Chat Activity updates date and chatting text information to firebase data child “lastMessage”. So, user can check each last texted message from chat list.

(2) Started from Chat List

Firebase addChildEventListener will receive added data dynamically from firebase database and update recycler view. Chat groups in chat list already has related user information, Chat Activity will not update user information to firebase.

(3) Menu List

- **Action ADD**

  Start ChatInviteActivity

- **Action Exit**

  Call exit method

(4) Exit

User can exit current chat group. If rest user number is lower than 2, Chat Activity will remove related chat data from Firebase. If not, Exit message will be updated to data, so that rest users in chat group can recognize the user exit.

4.3.2.7. Chat Invite Activity

Chat Invite Activity will show current friend to invite in a recycler view. When user touches a friend in a list, Chat Invite Activity queries friend field from firebase to check weather user gets blocked by that friend. If user is not blocked, inviteToChat method will be called.

(1) inviteToChat Method

This method will check weather invited friend is already in a current chat user list. If not, invited friend field will be added to array list, and getExistingUsers method will be called.

(2) getExistingUsers Method

If there is new member, I designed to open new chat group with new member. Therefore, this method will get all the user information joining current chat group and update to new chat group. When this method finishes, Activity will finish with result data including new chat key, so that Chat Activity will load new chat data from firebase for new chat group.

5. Discussions on Cloud Database

Cloud database has many advantages. Firstly, cloud database has a better price-performance ratio. We can trim costs and cut spending on hardware, software licensing and personnel and annual maintenance. If your database is not big, you can even use the cloud database free of charge. For example, we use the Firebase in data management. Certain Services are provided to us without charge up to the Fee Threshold. Secondly, cloud database is more efficient. You can use any computers, mobile devices to access databases, reducing the use of resources on the whole. Thirdly, cloud database has been configured for automated backups and maintenance. Users do not need to maintain databases, which reducing costs of maintenance.

However, there still exist some disadvantages. At first, our data is accessed on the Internet, which may be leaked. Therefore, data security is the big challenge. What’s more, although cloud database can satisfy most normal demands, when we explore our own applications, the cloud database cannot provide customized service.

6. Team Information

6.1. Responsibility

NAME	WORKLOAD
Wanjin Li	Post / Homepage
Taoran Ju	Login / Create Account
Hyun Jun Choi	Chatting