Project Proposal

Name: Dac Viet Nguyen
StudentID: 217497618

Overview:

Tavern Chat app is a messaging mobile application for students or workers to have a great and secure conversation with each other. The application is connected to Firebase Database to store user's information as well as message content. This app is not only for entertainments but also it is the greatest app to discuss and exchange knowledge between two people.

The application allows users to add friends with others by using the search function or QR barcode. Therefore, they can start to talk to each other and share PDF files or picture images. Some new functions were added in the project such as undo message that users just send or display time and date of the last message as well as last time a user is online. Every information about the conversation or user's information is saved in Firebase Database storage securely. Moreover, user could save picture in their gallery to share or save for further purpose

Product Purpose:

A. **Target Audience:**

This application was created because of the large social media applications but lack of unsecured chat apps. Every user is required to provide the only email, there is no more information that cyber-attack can collect from users. If audiences want to have a private chat with their friends, that would be a better option. Users can easily connect to friends via the Tavern Chat application without a complex verification. The application allows users to talk, to share documents and pictures several times. Besides, users can be attracted by the balance and responding time than it has in the application. Through there, there will be no cyber-attack could reach to their information.

B. Reasoning of how it demonstrates creativity:

This application uses the firebase database to share documents and information between two people. Firebase was created by Google which responses to the request faster than other server and it has unlimited storage which is free and has a lot of functions to create a real-time application. Firebase supports developers so much to develop back-ends such as storing an image, video, voice recording, or face detection.

The Application was added creation of QR code for each other and QR code scanner which can help users finding their friend without any difficulty. The Scanner could detect user id and connect to Firebase to retrieve user profile information.

Every information from the system is connected to the Firebase Realtime Database. Based on that, users can change information immediately as they want. In the conversation, they can send an image or picture to a friend and receive the same thing that they want to get. Any PDF files from the android device could be uploaded and download whenever they want. When a user does not want to see any message, they can remove from both users and delete a whole conversation. One of the functions that are different from other applications is no requirement to have a conversation. Users can chat with their friends without adding a friend. Whenever users want to access the application, they always require providing a real email and it needs to be verified. Moreover, the gallery function was added for saving user picture. Therefore, it would be a solution for user who want to clear their storage or do not want to lose some essential pictures

Feature:

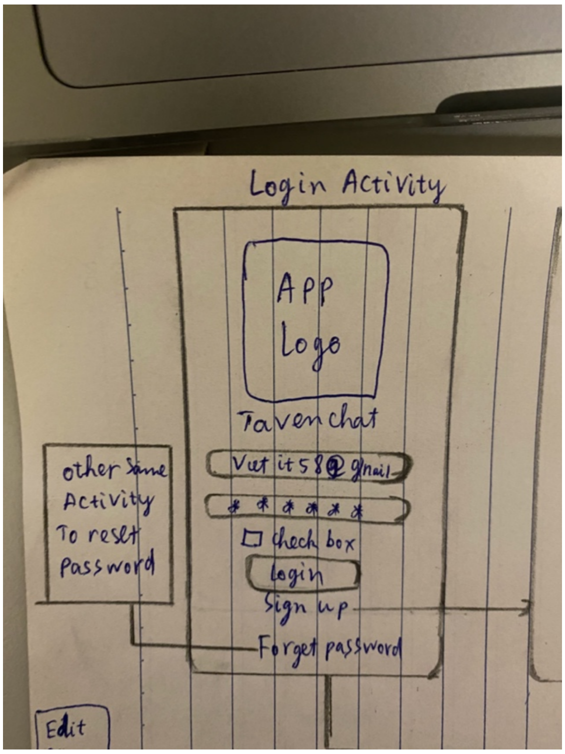
All of the features were included in the Tavern Chat:

A. Login/Sign Up function and forget Password

Any user wants to access and use the application, they need to register by their email. Then, verification will be sent to their email and requires them to verify if they want to use the app.

The register function will record user information and use QR code in the Realtime database due to future purposes after the login function. QR code will be sent to Firebase Storage with the name of each user ID to separate the name of the QR code.

The Forget password function will use Firebase authentication to send a reset password to their mail. The user is able to change a new password via the mailbox. When the password is changed successfully, audiences are able to apply them for login

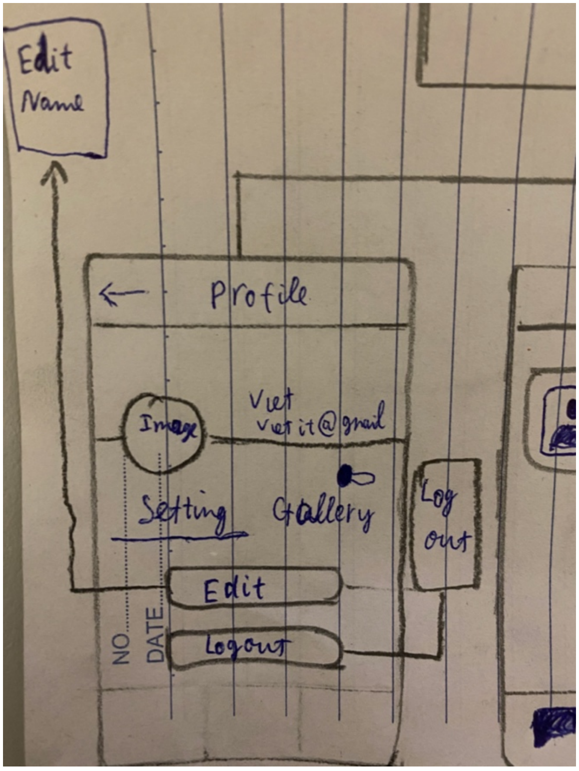


B. Homepage

The homepage displays list of friend chat that user have messaged before. The list could show who is online or offline and the message has not been read by users. Notification will be appeared automatically whenever user have a new message from their friend. Moreover, at the top corner on toolbar, there is a user symbol leading to user profile activity.
If users have logged in to the Tavern application before, the system will automatically let user access to the Homepage and if users want to switch to other account, they need to click on User profile and log out

C. User profile and Friend profile

The user profile and friend profile display every information about users and their email. For user profile, user could change their name or image profile and they can also log out from the system and switch to new account. For the friend profile, it could display friend information with 2 button adding friend or chatting functionality. User could send a friend request to others due to want to make friend. The request will be appearing on request lists. The gallery will display user's uploaded photos. It could be seen by anyone else if users do not click on privacy.

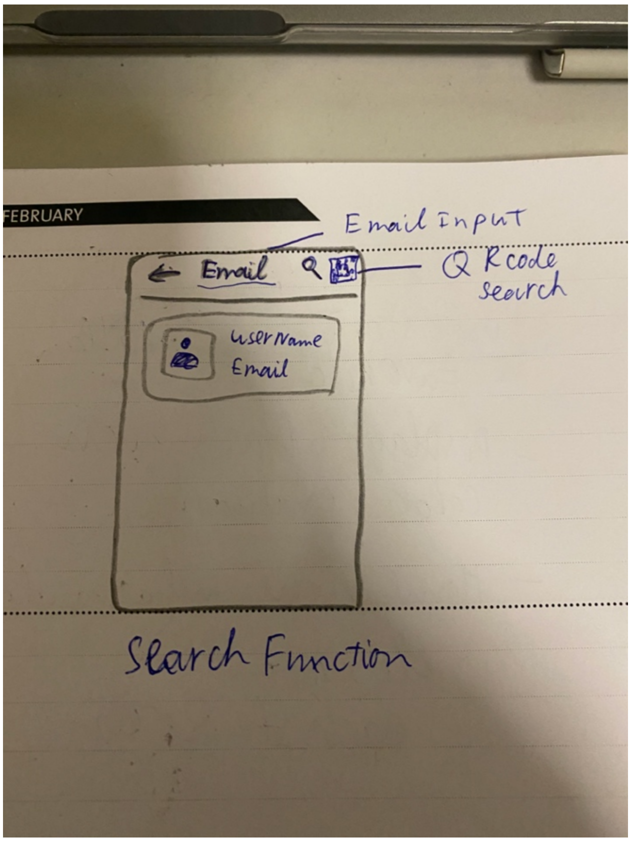


D. Chatting List

The Chatting list displays messages that users have texted with their friend before. The message will appear with friend's name and the latest message have not seen. This make user observing and continuing the conversation without any interruption or having to find in friend list.

E. Friend List and Search Friend

After sending the request to a friend and they accept the friend request. This function will save both information in friend list for 2 users. Now users could text with each other or remove from friend list if they want. The search function using looping based on user typing or QR code. QR code is able to scan user ID code and then, it sends user to friend profile where they could make friend request or start a conversation with your friends



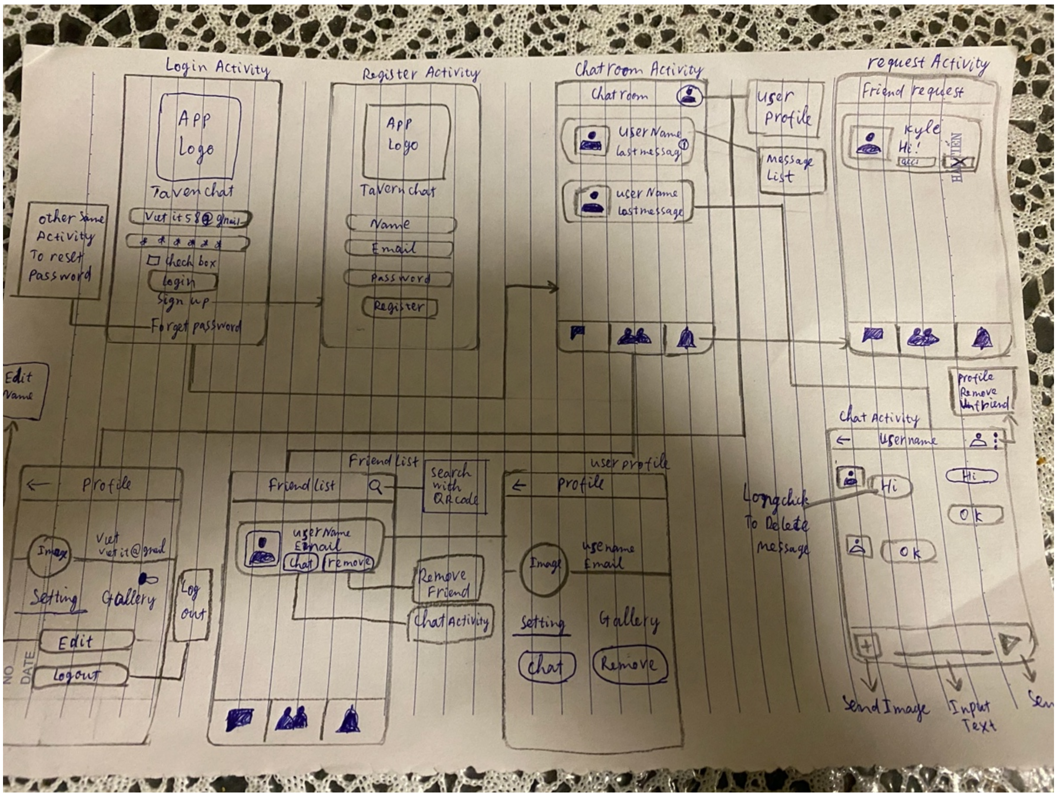
F. Chat Activity

Chat activity records user’s message and time which are label with 2 different colours for sender and receiver. User could send images or file documents to share with a fellow. Chat Activity has some functions which is enable users to delete message from them or from both sides.

G. Keep a user logged in

The system could identify when you do not log out from the system. This function will keep user stay inside Homepage whenever they reopen the applications.

Design



There are totally 8 main activities in the application and two other activities (Forget Password and Search function):

- **The homepage and register activities:** require user the enter their email and password to sign into the system. Those are designed with color grey (#b0bec5) with input text white to distinguish between background and input value. The logo was put on the top of the screen with the colour blue. There is a checkbox to remember the user's email and password

- **Chat room activity:** this includes a bottom navigation bar having 3 items: Chat, Friend, Request. In the middle of the interface, there is a recycle view list display message list that the user has been in conversation with. Users can click on the symbol at the right corner of chat room activity to access to user's profile.

- **User's Profile** (Chat room -> Click image to view at the right corner): this activity displays the user's image profile and set function. The user could change their name by clicking on Edit My profile or log out the system by clicking the log out button. If the user wipes from the left to the right side. They can see their gallery which was posted by them before. The gallery could be locked by the user using the switch button on the right side. if they do not want to share the picture with anyone else, the gallery will display a message "you cannot see this content".

- **Friend List** (Friend Button in the bottom navigation bar): The friend activity displays list friends that users have made friends before. There are two buttons performing chat or remove the friend. Whenever the user wants to see a friend profile or their gallery.

- **Request List** (Request list in the bottom navigation bar): this displays a friend request list who wants to make friends with each other. There are two buttons to perform accepting or rejecting a friend request.

- **Searching Friend** (Friend in the bottom navigation bar -> search symbol -> search bar and search by using QR code): this activity will display existing users who have registered and used the application. The search function on the top help user finding their friend easily when they type friend names. Moreover, users could also use friend QR code by clicking on the QR code symbol at the right corner of the searching activity. This function will scan the QR code and take the user to friend profile.

- **Friend Profile** (Chat List - > Chat activity -> Toolbar menu at the right corner -> Profile) or (Friend List -> click on a friend -> Friend Profile): Friend profile will be displaying of friend information. If two users have not been friends, the Chat button and Add Friend button will display on the screen. Whenever the user clicks adding friends, the request will be sent directly to others. The user could undo the request by clicking on the add button again. From the other side, the user who has the request could see 3 buttons: chat, accept friend, cancel the request. They can accept or reject the request if they want. Moreover, the gallery can display all the pictures posted in the app. The gallery will be invisible when the user's friend switches off to private mode.

- **Chat Activities:** (Chat list -> chat activity, Friend list -> chat button, Friend profile -> chat button) User could see all the messages and files have been sent to others. When a user types input text and clicks to send, the message will be saved in firebase and retrieved to chat activity. Users are able to delete or undo the message by long clicking the message. PDF files and pictures could be sent and retrieved as well. At the corner of chat activity, there is a toolbar menu that shows the option to remove or unfriend if a user wants to.

Data:

There are 2 types of data in this application:

- User database collects username, email, password and Image profile.
- Realtime database collects message, gallery links, and QR code of user

Runtime variables:

- String/Image: single value, all of user’s text, image URL, message are single values, so it O (0) complexity.
- There are totally 6 tables in firebase: User, Friend Request, Friend List, Message, Gallery and Notification. Every table contain specific text values and it will be transfer to image or number based on format in application: 1 byte in RAM.

- The Recycle View list collect user information from firebase as O(n^2) complexity
- There are 3 functions with > O(n^2) complexity: scan QR code(), push data() and retrieve.
- The image value will be loading from firebase storage, it takes about 4 second to get content

Fixed data:

-Most of data are retrieved from firebase storage, so it does not require much data from the local storage devices. However, the app will take about 50 mb to save some picture in the application and contents.

API/class Structure:

There are totally 13 classes, 4 Fragments, 6 Adapters, 3 Models in the project each of them has some essential functions to maintain the application running.

Login/Signup/Forget Password Activities (Main Activity):

- SignInwithEmailandPassword: this function will check if users have authentication in Database. If the username exists in the firebase system, it will take the user to the homepage.

- Checkbox: this function helps the user to save user information for the next signIn. It will remember user email and password.

- firebaseAuth.getCurrentuser() != null: this function identify user if they did not log out last time. It will keep users signing in until the user clicks on log out.

- Signup with Email and Password (Email, Password): the function loading data from firebase to verify user if user email has exited already in the system, users are not able to use that email anymore.

- Firebase: the firebase also creates QR codes for users, online status, and privacy for further purposes in the application.

Homepage Activity:

- onStart(): this method identifies a user's online and message databases in the firebase system. Users could see the name of a friendly chat and the last message from the conversation.

- DatabaseReference.addListenerForSingleValueEvent(): the looping function to get the message list from the message data table in firebase and add those to the array list. The information will be displayed in the recycle list view

- Countable.addValueEventListener(): that checks have seen value firebase, if a user has not seen the message, the number will appear based on number of non-seen message on the left side of the friend list

- Friend extends RecyclerView.ViewHolder: this initialize friend information name, email, have seen, image value in firebase database

- updateStatus(String state): this function performs updating user online status, date, and time currently.

- BottomNavigationView.OnNavigationItemSelectedListener(): this method to switch between items in the bottom navigation bar.

FriendList Activity:

- Find.setOnclickListener(): it takes the user to Search activity with the list and existing user using the application

- FirebaseRecycleOption<Users> options: the function creates recycle view getting user's friend information following using a model class named Users to get Name, Email, ImageUrl

- onBindViewHolder: This method internally calls to update the RecycleView.Viewholder contents with the name, email, image of users at the given position, and also set up some private fields to be used by RecyclerView.

- onDataChange(): this method collects user child information from the realtime database system and adds it to the recycling view.

- onCreateViewHolder(): it is invoked by the layout manager which creates layout inflate for recycle view and returns view holder.

- Remove_Friend(): this method will appear whenever they are a friend, the method is called to remove a friend from a friend list document.

- chat.setOnclickListener(): the chat method takes the user to chat activity to continue or start a conversation with each other.

Friend Request Activity:

- FirebaseRecycleOption<Users> options: the function creates recycle view getting user's friend information following using a model class named Users to get Name, Email, ImageUrl

- onBindViewHolder: This method internally calls to update the RecycleView.Viewholder contents with the name, email, image of users at the given position, and also set up some private fields to be used by RecyclerView.

- onDataChange(): this method collects user child information from the realtime database system and adds it to the recycle view.

- onCreateViewHolder(): it is invoked by the layout manager which create layout inflate for recycle view and returns view holder

- Accept/Cancel OnclickListener(): those methods perform the function to accept user requests or reject user requests. When they click on the button, it will delete data from request table in firebase database

Chat Activity:

- getFileName(Uri Uri): after taking the file from local storage, it helps the system to get the real name of the file, not only the document number. This function is very essential because users want to get exact filename.

- Back_buttonOnclicklistener(): this function call back the previous activity following the key "previous page" which is assigned as name of previous activity
- onCreateOptionMenu(): this creates menu toolbar with items set up in tool_bar_menu layout .

- onOptionItemSelected(): User could choose the option from the menu of the toolbar, it identifies which option is chosen and performs for a single option.
- Displayuserstate(): to calculate the current time and last online time. This provides a calculation to get the exact gap time and how long the user's friend has not been online.

- onActivityResult(): it will take the data from activity results and display the information on the screen. Inside these methods, the function will collect message information from who and send to who, what are the images, and send them to Firebase database.

- onProgress(): this calculates time uploading to the firebase from local storage. When it has done, the dialog will disappear.

- HaveSeen(): when the user has seen the message, it will update have seen the value in message table and update that it has seen by a friend.

FriendProfile:

- tab layout.addOnTabSelectedListener(): this method performs swiping left or right for the user to display 2 different fragment contents which are setting fragment and Gallery Fragment. When the user wants to display their gallery, they just need to wipe from the left to the right side to see the gallery.

- RequestManagement(): the function manages friend status when they are friends or not. If they are not friends, the slogan below friend image will be changed based on the current relationship between two users.

- FriendInfor(): it collects all user's information including name, email, image, and call request management method.

UserProfile:

- tabLayout.addOnTabSelectedListener(): this method performs swiping left or right for the user to display 2 different fragment contents which are setting fragment and Gallery Fragment. When the user wants to display their gallery, they just need to wipe from the left to the right side to see the gallery.

- aSwitch.setOncheckedChangeListenter(): perform setting up private or public gallery from friend's visibility. If it is on, the privacy is on and friends can not see the gallery.

- AlerDialog.Builder: whenever user click on image profile, a dialog will open to get the option from users, they could change image profile or display their QRcode.

- onActitityResult: this method gets an image from the device gallery and performs cropping this image into a square shape. When it is done, the profile image will be changed.

- QRdialog(): if the user chooses to Show my QR code, the dialog with a picture of QR code will be displayed to show their QR code. This image will be retrieved from Firebase Storage.

SearchFriend:

- TextChangedListener(): this method performs when user type in text edit search bar. When the text is changed, it will operate the searchUser() method.

- searchUser(): by using looping through Firebase database with child user. It will find the user has the same email put in text edit. And then, those users will be added in the array list and display on the screen

- ReadUser(): display existing users have used the application and add it in an array list to display.

QRcode:

- requestForCamer(): the method will perform the permission to access to camera of the devices, when it is done, the codeScanner will start scanning the QR code arroung and take the result back to activity
-
- setDecodeCallback(): this method will take the result from scanner and intent to user's profile.

4 Fragments:

- FriendGalleryFragment:
  + onDataChange(): getting the image from the gallery and the privacy permission from users table in firebase, if the user's privacy is public, friends can see all the pictures of the user. Otherwise, they are not able to watch it

- FriendSettingFragment:

  + RequestManagement(): the function manages friend status when they are friends or not. If they are not a friend, the slogan below friend image will be changed based on the current relationship between two users. Additionally, it performs adding friend, cancel friend request, Undo Friend Request method based on current state and request state

  + FriendInfor(): it collects all user's information including name, email, image, and call request management method.

  + Remove_Friend(): this method will appear whenever they are a friend, the method is called to remove a friend from a friend list document. After calling, the remove button will appear as adding friend

  + chat.setOnclickListener(): the chat method takes the user to chat activity to continue or start a conversation with each other

  + Accept_request(): when the user accepts the friend request, this method will be called to add a new friend and friend list while the request list will remove this friend request. It accesses to firebase system and removes value based on the address.

  + Cancel_ request(): when the user cancels a friend request, this accesses to the firebase system and remove value based on the address.

- GalleryFragment (For user):

+ upload.SetOnclickListener: this performance asking for the name of the image first. Then, it will ask permission to access local storage to get a picture.

+ onActivityResult: get Uri from the picture and push to firebase storage.

+ uploadFile(): it will transfer the picture to Firebase Storage and get URL from Firebase Storage. Then, it will save information about them in Realtime databases.

+getFileName(Uri uri): get image name from the gallery, not only document number.

- SettingFragment(For user):

+update.setOnclickListener: users are able to change the user name by click on the edit button. A dialog will appear asking for a new name, it could not accept the empty name, so the user should provide a name for it.

+ logout.setonClick: log out the account and update status offline for use.

+ updateStatus (String state): this method will be called whenever the user logs out from the system. Time, data, the state will be updated children in Firebase Database.

3 Models:

- Upload: getting UserID, Image name, and ImageUrl from the Gallery database table.
- Users: getting Username, User Id, Email, and profile Image Url from users table to add in the array list.
- Message: getting all information that has in Message table such as message-id, text, type, have seen, from, time. It will collect all of that information from database.

6 Adapter:

- FriendImageAdapter: Adapter for Friend Gallery display.
- FriengPageAdapter: Adapter for tab layout in friend profile and user profile.
- ImageAdapter: Adapter for user gallery display.
- MessageAdapter: Adapter for Message View in chat Activity
- UserAdapter: Adapter for User List in Friend, Chat room, search Friend and Request
- PageAdapter: Adapter for user setting in user profile activity.

Resources Required:

https://www.flaticon.com/search?word=chat
https://www.publicdomainpictures.net/en/hledej.php?hleda=cover+image