1. **Code Documentation**
   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.

* 1. **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.

* + 1. **Friend Activity**

In friend activity, user can add friend freely by searching email. 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 data 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.

* + - 1. **Menu List**
         1. **Action Friend**

Show friend list in a recycler view

* + - * 1. **Action Chat**

Show Chat list in a recycler view

* + - 1. **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 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.

* + - 1. **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 require more memory, I’d better use Glide library to handle image from next project.

* + - 1. **Show Friend List Method**

This method queries all the friend data from firebase sorted by user’s own ID. Queried data will be added to friend data array list through Friend data class. And swapData from friend recycler view adapter will be called to refresh recycler view item list. By building MVC, I was able to 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 touch a friend in a list, application starts a chat activity related with selected friend.

* + - 1. **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 touch a chat list, application starts a chat activity related with selected chat group.

* + 1. **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 data from array list by adapterPosition.

* + 1. **Chat List Adapter**

Chat list adapater handle 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.

* + 1. **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.

* + 1. **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 data from data.

* + 1. **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 data 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 update date and chatting text information to firebase data child “lastMessage”. So, user can check each last texted message from chat list.

* + - 1. **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.

* + - 1. **Menu List**
         1. **Action ADD**

Start ChatInviteActivity

* + - * 1. **Action Exit**

Call exit method

* + - 1. **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.

* + 1. **Chat Invite Activity**

Chat Invite Activity will show current friend to invite in a recycler view. When user touch a friend in a list, Chat Invite Activity query friend data from firebase to check weather user get 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 data will be added to array list, and getExistingUsers method will be called.

* + - 1. **getExistingUsers Method**

If there is new member, I designed open new chat group with new member. So 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.