Project Report – Mobile Development

Riccardo Caprile (4370774) – Giulia Benvenuto (4678610)

Introduction

Our idea for the Mobile Development project was to implement a messenger chat, called “ChatHub”. This app gives you the possibility to create an account, to login with the created account and then to choose another user to chat with.

This project has been implemented in Java language using the Android Studio IDE and Firebase Realtime Database. We tested the application using two different physical devices with different screen sizes: Oppo Reno and Samsung Galaxy Z-Fold, and also using the emulator provided by Android Studio.

Implementation

MainActivity

The MainActivity is the first Activity you will see when you launch the App. It is composed of one ImageView with the logo, two EditTexts one for the username and

the other one for the password and two buttons, one for the login and the other one for the SignIn.

Deeper in the real implementation, in this Activity we only have the *onCreate*() method.
Inside the *onCreate()* method, the two most important lines of code are the initialization of the database using the *getInstance* (it takes as input the URL of our database) and the *getReferences* (it takes as input the name of the key we want to access) methods which are imported from the FirebaseDatabase package.

SignIn Button: the SignIn Button has a ClickListener which in the *onClick*() method redirects you to the SignInActivity through an Intent.

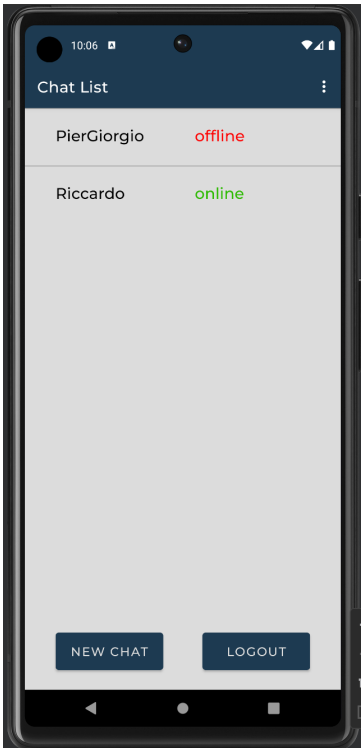
LogIn Button: the LogIn Button has a ClickListener which in the *onClick()* method takes the username and password fileds inserted in the EditTexts and put them in two variables in order to save them. Then we search inside the Database whether the inserted username is already registered or not, if are not registered you will see a warning message telling you “No such user exists”. If the user exists and the password is correct you will be redirected to the ChatListActivity and with an Intent we save the username of the logged user in order to have it always saved and easily access the database, otherwise if the password is wrong you will get an error. In the end the value of the status is set to “online”.

SignInActivity

The SignInActivity is used when you don’t have an account and you want to register to the App. It is composed of three EditTexts, where you write your username, your password and for safety reasons the confirmation of your password. In the end you have a button to complete the registration.

Deeper in the implementation, the code starts with the usual initialization of the Firebase Realtime Database. Then we populate a List where we store all the usernames already registered in the database in this way we can check if the username that you choose is already used or not. Furthermore, if the username does not already exist in the database and the inserted passwords match you can complete the SignIn and your user will be created in the database.
When you create successfully the profile you will see a Toast appearing at the bottom of the screen saying “Profile created successfully” and you will be redirected to the MainActivity (Login) otherwise if the passwords don’t match and you will see a Toast saying “Inserted passwords do not correspond”.

ChatListActivity



The ChatListActivity is used to display all the users with whom you already have a chat, the list of users shows their name and their status (online - offline), if there are many users you can scroll down to access the lasts.

Each time you want to know if someone has texted you and the actual state (online - offline) of your contacts you should refresh by using the three small dots in the right-top corner.
When you click on one item of the RecyclerView you are redirected to the chat with that user.
When you long click on one item you can delete it and the RecyclerView will be refreshed.
In the bottom part of the Activity we have two buttons, one to create a new chat with another user and the other one to logout from the App.

This Activity is based on a RecyclerView which allows you to display the entire list of the chats you have. The Recycler view is implemented with two auxiliary classes: Adapter and ModelClass.

Adapter: this class is used in conjunction with the RecyclerView to provide the data and create the views for individual items in the layout, it allows us to dynamically display the data (the chat list) in the ChatListActivity.
It contains:

- The constructor
- The *onCreateViewHolder()* method: this method is responsible for creating a new instance of the ViewHolder class.
- The *onBindViewHolder* method: this method is responsible for binding data to the views within a ViewHolder for a given position in the data set.
- The *detItemCount()* method: this method return the number of users.
- The *ViewHolder* class: this class represents a single item view within the RecyclerView and is responsible for holding references to the views within that item. In this class we have the implementation of the ClickListener and LongClickListener for each item of the RecyclerView. In the end we have the *setData* method that sets the data in the TextViews in each item in the RecyclerView.

MessageModel: this class handles the creation of the chats between two users, making the status and the username (two TextViews) visible.

Now let's go through the implementation of ChatListActivity:
The Activity starts with the two most important methods *initData()* and *initRecyclerView().*

initData(): this method is responsible of the update of the chat list and user status. It saves on a list all the usernames and then it checks if the user logged already has a

chat with another user and eventually creates a new ModelClass item that will display in the RecyclerView both the username and the its status.
The method works with two different lists, depending on if the user has deleted a chat with the long onClick that will be explained later.

initRecyclerView(): this method is responsible of the initialization and the update of the RecyclerView infact we use it both when the ChatListActivity is created for the first time and when we refresh the Activity using the three buttons.

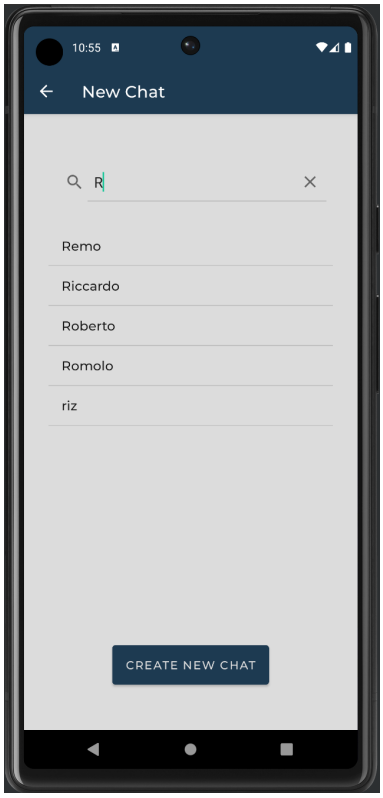
Then there are two possibilities of clicking: *onItemClick(int position)* and *onItemLongClick(int position).*

onItemClick(int position): this method is responsible to redirect you to the ChatActivity with the chosen user by using the putExtra() we save both the sender and the receiver.

onItemLongClick(int position): this method is responsible to delete the chosen chat, it removes the user clicked from the list used in *initData()* and it also removes the data from the database.

The lifecycle of the Activity is managed from the two methods *onPause()* and *onResume().* They are responsible of setting the status of the user when he puts the app in background and when he resumes it.

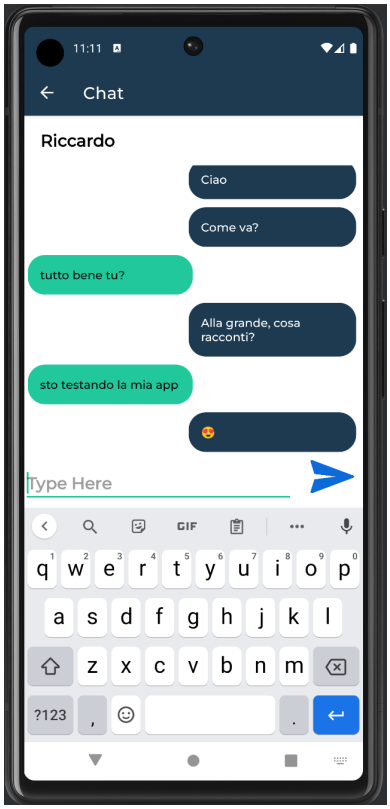
NewChatActivity



This Activity is used to search for new users to chat with. It is composed of a search bar and a ListView in which you can write the starting letters of a name and you will see a dynamic list with all the possible names depending on what you have write in the search bar. If you click on one element of the ListView the chosen name will be completed in the search bar then you can start a new chat by clicking the “create new chat” button which will redirect you to the ChatActivity.

Deeper in the implementation, at the creation of the Activity we populate a list with all the usernames which will be used in the ListView.
If you try to create a chat with a user who doesn’t exist a Toast will be raised telling you “No match found”, otherwise if you chose a valid username and you tab on the “create new chat” button the ChatActivity will be launched passing to it the receiver username taken from the search bar and passed through the putExtra() method of an Intent.
In this Activity we also implemented the up navigation which allows you to go back to the ChatListActivity.
Also in this case we have implemented both the *onPause()* and *onResume()* methods for updating the users status.

ChatActivity



This Activity implements the chat between two users. It is composed of a TextView in which there is the receiver of your messages, a RecyclerView that shows you all the messages received and sent, an EditText in which you can write your message and a button with an icon that allows you to send the message. The sent messages will be displayed in the right side of the screen and the received on the left side of the screen.

This Activity is based on a RecyclerView which allows you to display all the messages that you have with another user. The RecyclerView is implemented with two auxiliary classes: MessageAdapter and MessageModel.

MessageAdapter: this class is used to display a list of messages in a RecyclerView in the ChatActivity. It contains:

- The constructor
- The *onCreateViewHolder* method: this method is responsibe for creating a new instance in the ViewHolder class and inflating the appropriate layout for the item based on its viewType.
- The *onBindViewHolder()* method: this method is responsible for binding data to the views within a ViewHolder for a given position in the data set. In this

case, it sets the text of a TextView within the ViewHolder to the text value from the corresponding item in the list data set.
- The *getItemCount()* method: this method retursn the size of the list in which are saved the messages.
- The *ViewHolder* class: this class represents a single item view within the RecycleView and is responsible for holding references to the views within that item.
- The *getItemViewType* method: this method belongs to the ViewHolder class and is used to determine the view type of an item at a given position in the data set. It helps in differentiating between different types of item views within the RecyclerView.

MessageModel: this class handles the creation of the messages in ChatActivity.

Now let's go through the implementation of ChatActivity:
The two most important methods are *loadMessage()* and *sendMessage()*.

loadMessage(): this method loads all the messages in the RecyclerView taken from the database of the user logged.

sendMessage(): when a message is sent, this method updates the database creating a relationship between the sender and the receiver putting the text of the message in the username of the sender. The chat in the database is created when the first message is sent.

When you create a new chat with someone from the NewActivityChat you will be redirected to the ChatActivity, at this point if you use the up navigation, generally, you return to the parent Activity (in this case NewChatActivity), but we would like to return to the ChatListActivity so we used *intent.addFlag(Intent.FLAG_ACTIVITY_CLEAR_TOP)*. We also had to override the *onBackPressed()* method used when the back button of the up navigation is tapped.

Also in this case we have implemented both the *onPause()* and *onResume()* methods for updating the users status.

Considerations about the problems

We encountered some problems during the implementation:

- How to use the database: since we never used it we had to understand for example how to create the primary key and then how to add to it the user's information.
- How to save messages between two users.

➔ We solved these two problems by checking the Firebase documentation online.

- How to use the RecyclerView: since we never used we had to understand how to create it and manage it for both the ChatListActivity and the ChatActivity.

➔ We solved this problem by searching for examples and by checking the Android documentation.

- How to implement the search bar in the NewChatActivity and the ListView that contains the usernames.

➔ We solved this problem by searching for examples and by checking the Android documentation.

- How to manage the deletion of a particular chat from the ChatListActivity because we had to refresh the list which was not updated.

➔ We solved this problem using two different lists, the first one when the Activity starts, the second one when a user is deleted.