# Swaheda

Wilmer Suarez
wilmer.suarez@stonybrook.edu

## 1 – INTRODUCTION

This document will provide a detailed technical overview of the Android application. *Swaheda* is a private messaging app that provides a safe and easy way to communicate with friends/family. The application is to be implemented for use on an Android Mobile device with an API of 23 or higher (Android Marshmallow or higher).

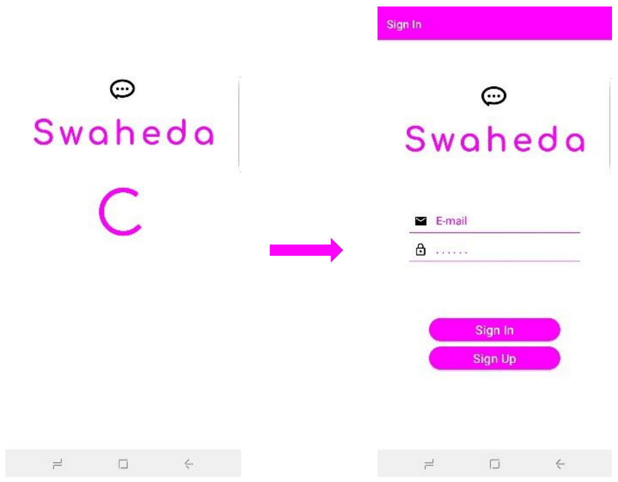
## 2 – FEATURES

The app features include:

- Create private user accounts
- Upload a profile picture
- Users can search for friends
- Users can send/accept/decline friend requests
- Friends List
- Users can unfriend current friends
- Friends List shows when the users became friends
- See other user's profiles
- Users can send and receive text messages in real-time
- Users can send and receive images in real-time
- Time message was sent is shown

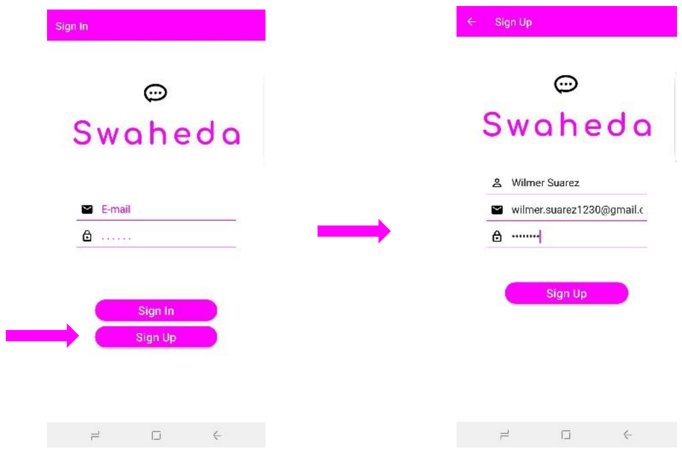
## 3 – SPLASH SCREEN & SIGN IN / SIGN UP

When the user enters the app, they are greeted by a loading screen that redirects to the login screen. The user then has the option to create a new account or sign in to an existing one.

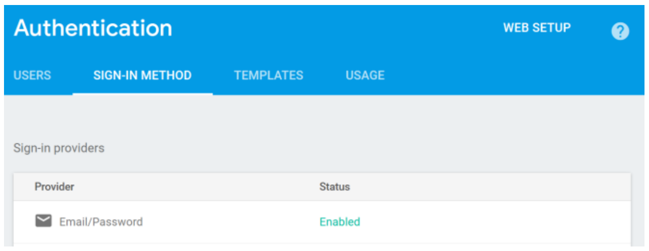


The splash screen lasts 3 seconds by creating a thread to put the program to sleep for 3000 milliseconds (3 seconds).

To create a new account, the user would press the sign-up button. The user gets transferred to a new layout where they can enter their Name, E-mail, and desired Password.



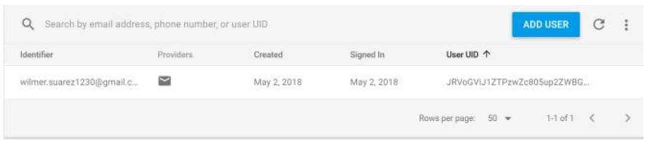
For this functionality, Firebase Authentication is used. This functionality is enabled on the Firebase Console:



Sign-in providers

| Provider | Status |
|---|---|
| Email/Password | Enabled |

Below, in the Firebase Console, after the user inputs their credentials and presses the sign-up button, a new user is created. The user's E-mail serves as their identifier and a Unique User ID is created. The console also shows when the user was created and the date they last logged in.



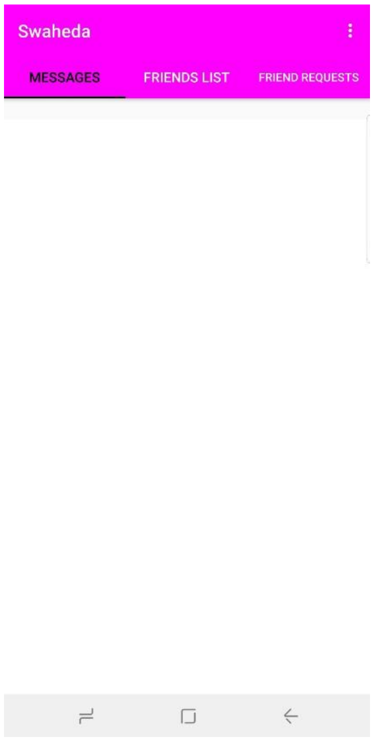
| Identifier | Providers | Created | Signed In | User UID |
|---|---|---|---|---|
| wilmer.suarez1230@gmail.c... | ✉ | May 2, 2018 | May 2, 2018 | JRVoGViJ1ZTPzwZc805up2ZWBG... |

Rows per page: 50 1-1 of 1

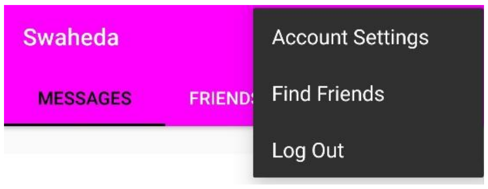
## 4 – HOME

After creation of the account, the user is moved to the main layout where there are 3 fragments:

- Messages Fragment
- Friends List Fragment
- Friend Requests Fragment



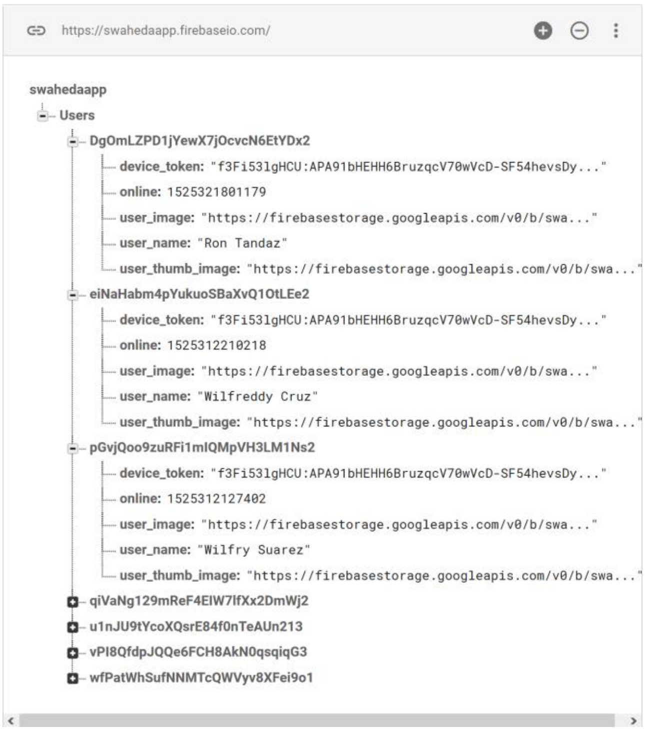
This layout also has a toolbar with a menu. The menu shows the options to view the current users Account Setting, Search for Friends or Log Out.



This feature was accomplished using an Android Toolbar. The toolbar was used together with a ViewPager, a TabsPagerAdapter, and a TabLayout.

These three components are used to differentiate between the three fragments. A different fragment shown depending on which one the user is currently looking at. The adapter overrides the getItem, getCount, and getPageTitle functions of the FragmentPagerAdapter class. This adapter gets and displays the Fragment selected by the user.

All accounts created will also be reflected on the Firebase Database. A node is created named "Users" with all its children being the User UID's of every account (The user's personal data is stored here):



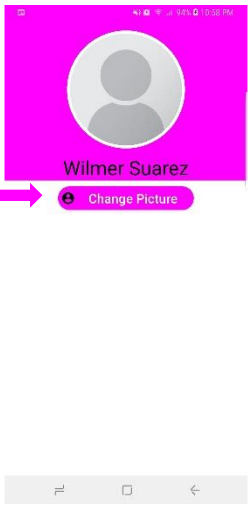
```
https://swahedaapp.firebaseio.com/

swahedaapp
  Users
    DgOmLZPD1jYewX7jOcvcN6EtYDx2
      device_token: "f3Fi531gHCU:APA91bHEHH6BruzqcV70wVcD-SF54hevsDy..."
      online: 1525321801179
      user_image: "https://firebasestorage.googleapis.com/v0/b/swa..."
      user_name: "Ron Tandaz"
      user_thumb_image: "https://firebasestorage.googleapis.com/v0/b/swa..."
    eiNaHabm4pYukuoSBaXvQ1OtLEe2
      device_token: "f3Fi531gHCU:APA91bHEHH6BruzqcV70wVcD-SF54hevsDy..."
      online: 1525312210218
      user_image: "https://firebasestorage.googleapis.com/v0/b/swa..."
      user_name: "Wilfreddy Cruz"
      user_thumb_image: "https://firebasestorage.googleapis.com/v0/b/swa..."
    pGvjQoo9zuRFi1mIQMpVH3LM1Ns2
      device_token: "f3Fi531gHCU:APA91bHEHH6BruzqcV70wVcD-SF54hevsDy..."
      online: 1525312127402
      user_image: "https://firebasestorage.googleapis.com/v0/b/swa..."
      user_name: "Wilfry Suarez"
      user_thumb_image: "https://firebasestorage.googleapis.com/v0/b/swa..."
    qiVaNg129mReF4EIW7lfXx2DmWj2
    u1nJU9tYcoXQsrE84f0nTeAUn213
    vPI8QfdpJQQe6FCH8AkN0qsqiqG3
    wfPatWhSufNNMTcQWVyv8XFei9o1
```

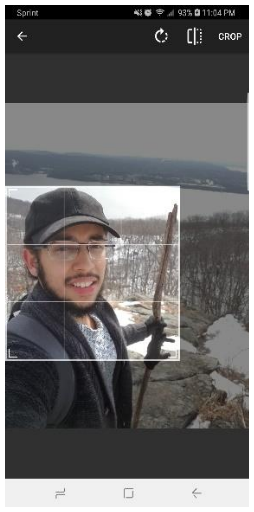
The data stored in the database for each user is their Device Token, a variable named "Online" that is later used to determine when the user was last online, their Profile/Thumb Image locations, and their name.

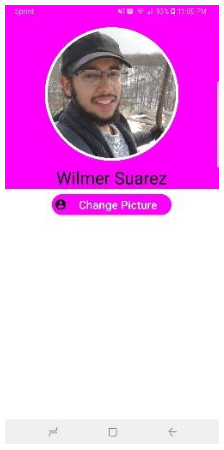
## 5 – ACCOUNT SETTINGS

The Account Settings layout allows the user to change their default profile picture from their gallery into one of their choosing when they press the “Change Picture” button.

The picture is processed using the Circle Image View library [1], a Compressor library [2], and the Image Cropper library [3]. The compressor library allowed for smaller sized images to be stored in the Firebase Storage and allowes for faster processing times. The Image Cropper library allows the user to crop out a section of the image chosen to be used as the profile picture (always with a ratio of 1:1). The Circle Image View library shows the profile picture in a circular frame.



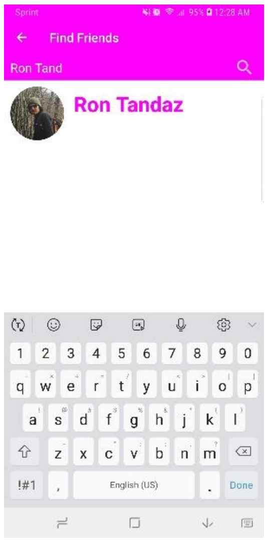


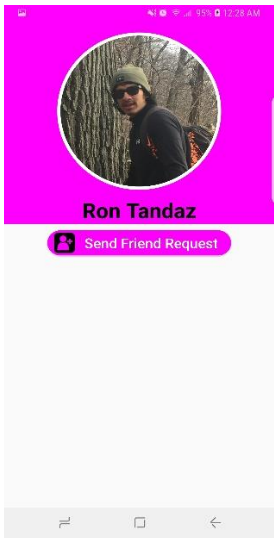


The Picasso Library is used to load the Profile Picture into the View [4].

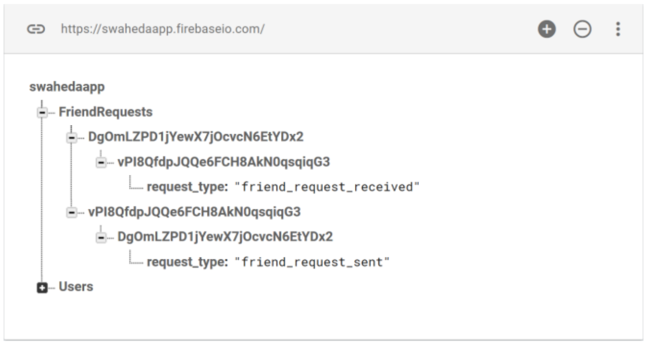
## 6 – FIND FRIENDS

When the user selects the Find Friends option from the menu on the main layout, they are presented with a search field. Here, typing the name of the user you want to add as a friend will display the users picture. Pressing the picture will redirect the user to the other user's profile where a "Send Friend Request" button is present.





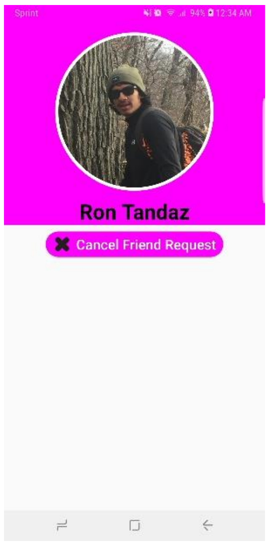
This is also reflected on the Firebase Database:



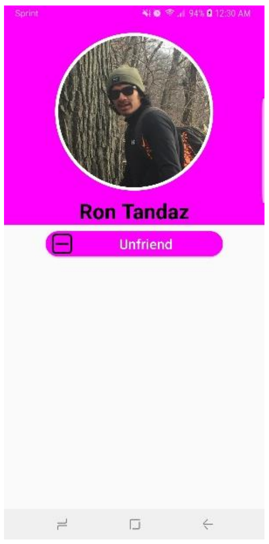
```
https://swahedaapp.firebaseio.com/

swahedaapp
- FriendRequests
  - DgOmLZPD1jYewX7jOcvcN6EtYDx2
    - vPI8QfdpJQQe6FCH8AkN0qsqiqG3
        request_type: "friend_request_received"
  - vPI8QfdpJQQe6FCH8AkN0qsqiqG3
    - DgOmLZPD1jYewX7jOcvcN6EtYDx2
        request_type: "friend_request_sent"
+ Users
```

The user sending the request will be shown as a node presented as the User UID with the User UID of the friend receiving the request as a child and one more child with the type of request. Similarly, with the receiving user.

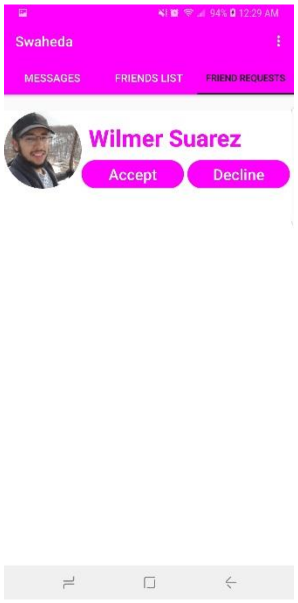
The "Cancel Friend Request" button then appears. If pressed, the nodes in Firebase will be deleted.

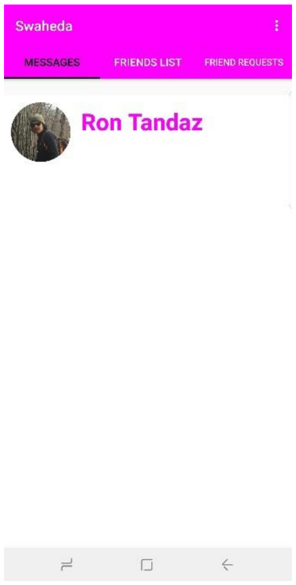


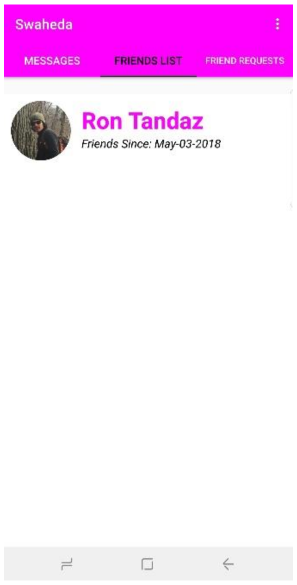
If the user is already a friend, the "Unfriend" button is present.



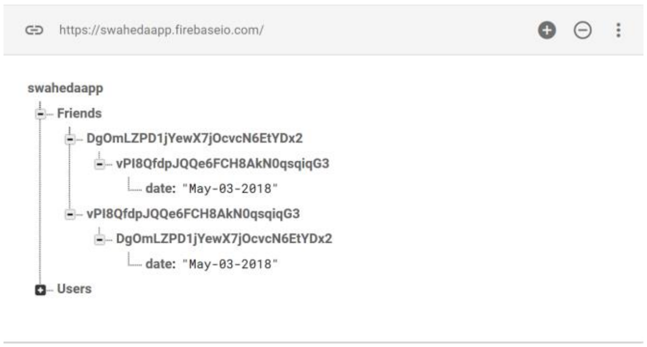
The user receiving the friend request will go into the "Friend Requests" Fragment in the main layout and will see the request, with two buttons: "Accept" and "Decline".







If the user declines the request, again, the corresponding nodes in the database will be deleted. If the user Accepts, this creates a "Friend" node in the database. This node has, as children, all the user's relationships, with the date that their friendship began.



```
https://swahedaapp.firebaseio.com/

swahedaapp
 Friends
   DgOmLZPD1jYewX7jOcvcN6EtYDx2
     vPI8QfdpJQQe6FCH8AkN0qsqiqG3
       date: "May-03-2018"
   vPI8QfdpJQQe6FCH8AkN0qsqiqG3
     DgOmLZPD1jYewX7jOcvcN6EtYDx2
       date: "May-03-2018"
 Users
```

Once friends, the user will appear in the Messages and Friends List fragment with the name and date that the friendship was made.

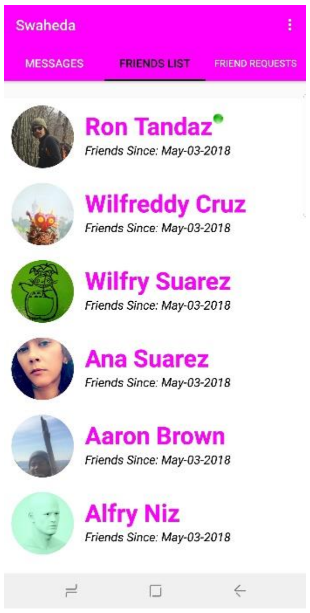
## 7 – LOG OUT

Selecting the Log Out options in the menu will change the user's status to "offline" and, using a FirebaseAuth object, sign the user out and return them to the Login layout.

## 8 – FRIENDS LIST FRAGMENT

The main fragments are all created using a RecyclerView that holds a list of layouts: the representation of the corresponding users.

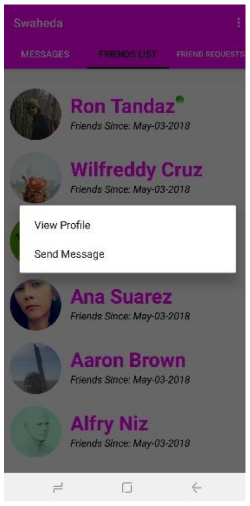
The Friends List fragment shows all the friendships of the logged in user. The friends are displayed, once again, using CircleView for their profile thumb image and the date the friendship was created. If the user is online, a green icon appears next to the user's name:



Ron Tandaz
Friends Since: May-03-2018
Wilfreddy Cruz
Friends Since: May-03-2018
Wilfry Suarez
Friends Since: May-03-2018
Ana Suarez
Friends Since: May-03-2018
Aaron Brown
Friends Since: May-03-2018
Alfry Niz
Friends Since: May-03-2018

In this fragment, when selecting a user, a dialogue box appears with two options:

- View Profile
- Send Message



Ron Tandaz
Friends Since: May-03-2018
Wilfreddy Cruz
Friends Since: May-03-2018
Ana Suarez
Friends Since: May-03-2018
Aaron Brown
Friends Since: May-03-2018
Alfry Niz
Friends Since: May-03-2018

Selecting "View Profile" will redirect the user to the corresponding user's profile. "Send Message" option will open the Chat activity (more on that in section 11).

The data used to display the name and thumb image of each user is acquired from the Firebase Database. This is done using the FirebaseUI for Android library [5]. This library allows for quick connection to common elements in the Firebase Database.

The Friend list was populated using a FirebaseRecyclerAdapter Object. This object worked in conjunction with a separate class used to get the user details and a View Holder class. The View Holder was used to apply the data received from the database into the Views for the image, name, date… etc.

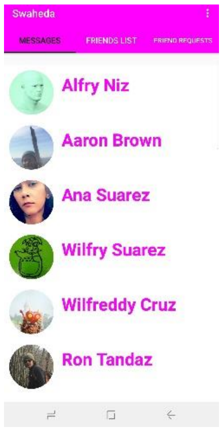
## 9 – FRIEND REQUESTS FRAGMENT

As explained previously, the friend requests fragment will display user data when a request is sent or received.

This fragment is also implemented using a RecyclerView to hold the user layouts. The same FirebaseUI library and method is used.

## 10 – MESSAGES FRAGMENT

The messages fragment is implemented the same way as the other fragments. This fragment displays all the friends the user has.



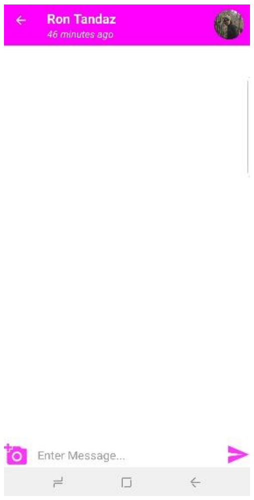
Alfry Niz
Aaron Brown
Ana Suarez
Wilfry Suarez
Wilfreddy Cruz
Ron Tandaz

Selecting any of the friends will open the Chat Activity where the user can begin sending messages to the corresponding friend.

## 11 – CHATTING

Once again, the Chat activity is implemented using a RecyclerView to hold the layout for the messages and images sent/received by the users.

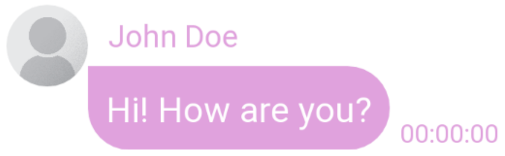
The layout consists of this RecyclerView, an EditText (to write messages), and two ImageButtons (one for submitting an image and another for sending a message).



The top of the layout shows the name of the user receiving messages, their profile picture, and how long ago they were last online. This was calculated using code from [6].

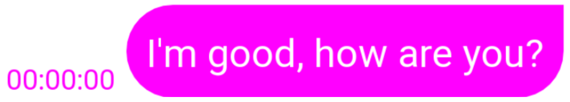
The two layouts created for the users are shown below:

*Receiver:*



Hi! How are you?
00:00:00

*Sender:*



00:00:00
I'm good, how are you?

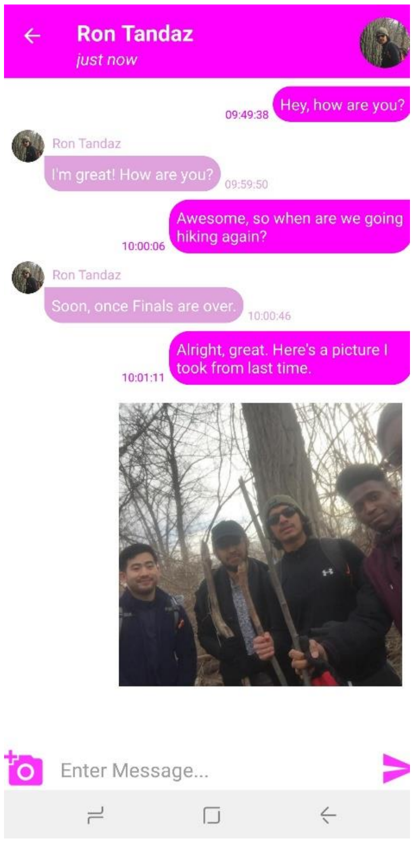
It is redundant to display the profile picture and name of the sender. Therefore, two separate layouts are created.

Unlike the previous implementations, there are two View Holders created, one for the user sending the message and one for the message sent by the "receiver". These View Holders were created inside a MessageAdapter class. This class is used to store the list of messages, determine whether a message is a "sent" message or a "received" message, and inflate the appropriate layout within the RecyclerView.

After determining the type of message, the message is sent to the appropriate holder that holds member views that are then bound to the information contained in the message (from the Firebase Database).

Below is an example of a conversation between two users:

*Sender*:



Ron Tandaz
just now

09:49:38 Hey, how are you?

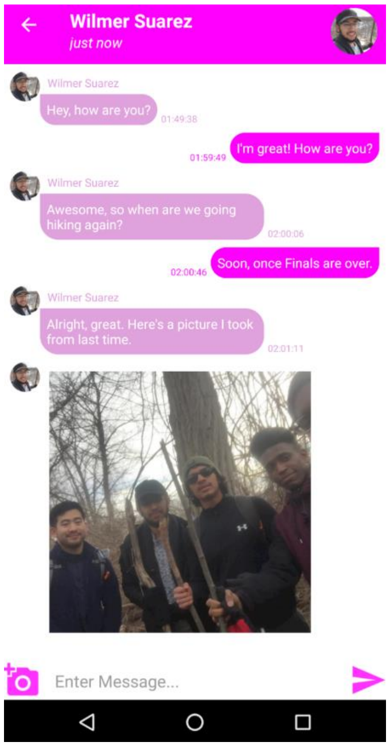
Ron Tandaz
I'm great! How are you? 09:59:50

10:00:06 Awesome, so when are we going hiking again?

Ron Tandaz
Soon, once Finals are over. 10:00:46

10:01:11 Alright, great. Here's a picture I took from last time.

*Receiver*:



Wilmer Suarez
just now

Wilmer Suarez
Hey, how are you? 01:49:38

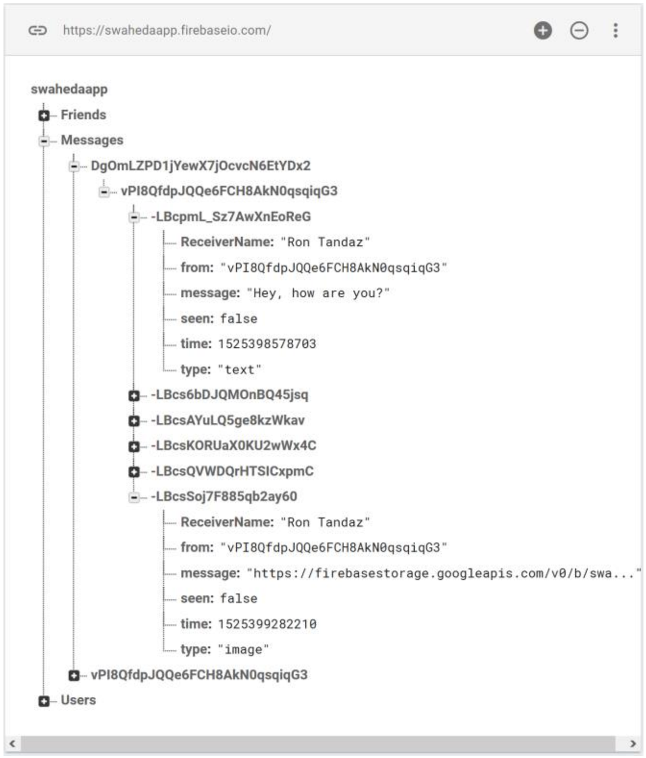
01:59:49 I'm great! How are you?

Wilmer Suarez
Awesome, so when are we going hiking again? 02:00:06

02:00:46 Soon, once Finals are over.

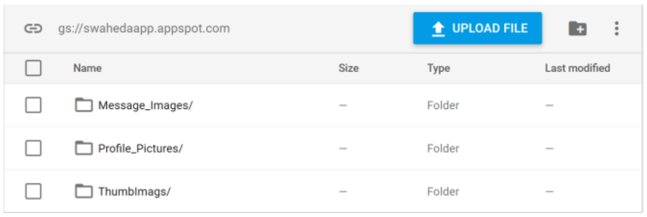
Wilmer Suarez
Alright, great. Here's a picture I took from last time. 02:01:11

All messages are stored in the Firebase Database:

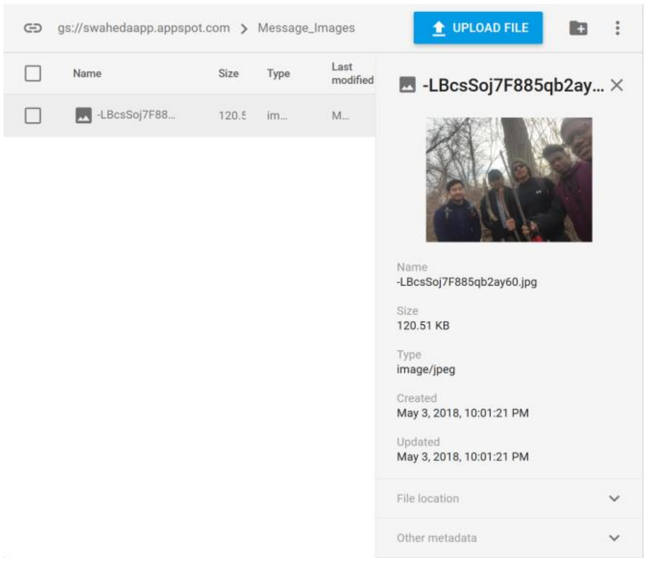


```
https://swahedaapp.firebaseio.com/

swahedaapp
  Friends
  Messages
    DgOmLZPD1jYewX7jOcvcN6EtYDx2
      vPI8QfdpJQQe6FCH8AkN0qsqiqG3
        -LBcpmL_Sz7AwXnEoReG
          ReceiverName: "Ron Tandaz"
          from: "vPI8QfdpJQQe6FCH8AkN0qsqiqG3"
          message: "Hey, how are you?"
          seen: false
          time: 1525398578703
          type: "text"
        -LBcs6bDJQMOnBQ45jsq
        -LBcsAYuLQ5ge8kzWkav
        -LBcsKORUaX0KU2wWx4C
        -LBcsQVWDQrHTSICxpmC
        -LBcsSoj7F885qb2ay60
          ReceiverName: "Ron Tandaz"
          from: "vPI8QfdpJQQe6FCH8AkN0qsqiqG3"
          message: "https://firebasestorage.googleapis.com/v0/b/swa..."
          seen: false
          time: 1525399282210
          type: "image"
    vPI8QfdpJQQe6FCH8AkN0qsqiqG3
  Users
```

All types of images (Thumb images, Profile Pictures, and Message Images) are also stored in the Firebase Storage (compressed):



| Name | Size | Type | Last modified |
|---|---|---|---|
| Message_Images/ | – | Folder | – |
| Profile_Pictures/ | – | Folder | – |
| ThumbImags/ | – | Folder | – |



| Name | Size | Type | Last modified |
|---|---|---|---|
| -LBcsSoj7F88... | 120.5 | im... | M... |

Name
-LBcsSoj7F885qb2ay60.jpg

Size
120.51 KB

Type
image/jpeg

Created
May 3, 2018, 10:01:21 PM

Updated
May 3, 2018, 10:01:21 PM

## 12 – FUTURE FUNCTIONALITY

Future additions to the app include:

- Scaling for different screen resolutions
- User security (Message/Image Encryption)
- Allow users to sign up using Gmail, Facebook, Twitter.
- Chat heads
- Message "seen" functionality
- Allow users to send and save other type of files
- Allow users to delete their account
- Notifications
- General Speed/Memory usage upgrades
- General UI upgrades

## References

[1] Dodenhof, Henning. Copyright 2014 – 2018 "CircleImageView" github.com/hdodenhof/CircleImageView

[2] Zetra. Copyright 2016. "Compressor" https://github.com/zetbaitsu/Compressor

[3] Arthur Teplitzki. Copyright 2016. "Android-Image-Cropper" https://github.com/ArthurHub/Android-Image-Cropper

[4] Square, Inc. Copyright 2013. "Picasso" http://square.github.io/picasso/

[5] "FirebaseUI-Android" https://github.com/firebase/FirebaseUI-Android

[6] Google Inc. Copyright 2012. "GoogleIoGetTimeAgo" https://gist.github.com/erwindetorres/

[7] Google LLC. https://developer.android.com/reference/org/w3c/dom/Document

[8] Terry & Alek Jeziorek. May 9th, 2017. "Android Chat Tutorial: Building a Messaging UI" https://blog.sendbird.com/android-chat-tutorial-building-a-messaging-ui