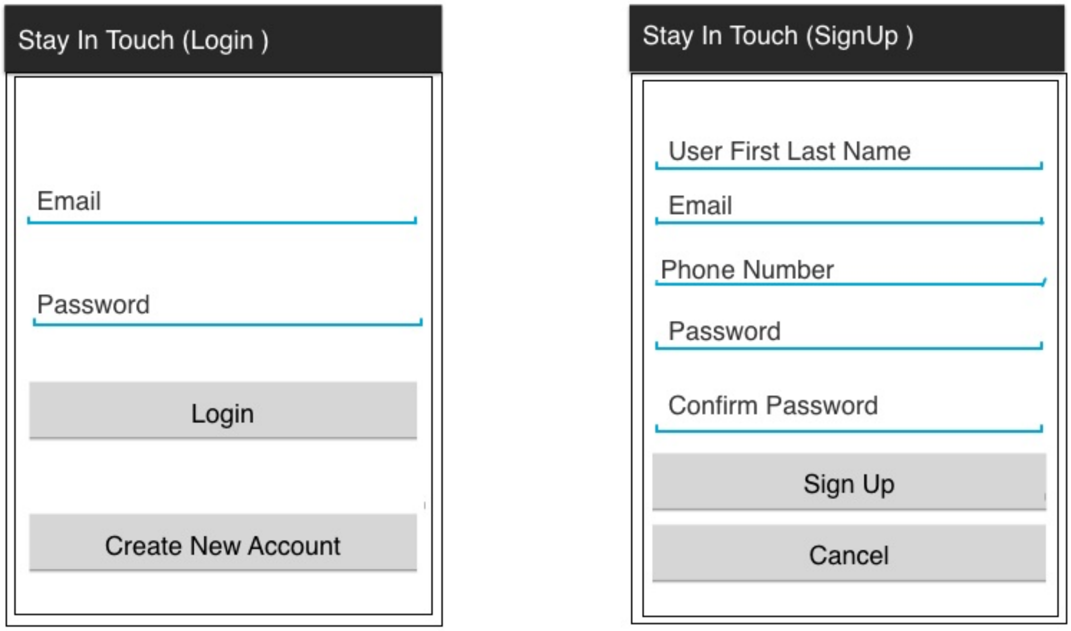
ITIS/ITCS 4180/5180 Mobile Application Development
Homework 9

Basic Instructions:

1. In every file submitted you MUST place the following comments:
   a. Assignment #.
   b. File Name.
   c. Full name of all students in your group.
2. Each group should submit only one assignment. Only the group leader is supposed to submit the assignment on behalf of all the other group members.
3. Your assignment will be graded for functional requirements and efficiency of your submitted solution. You will loose points if your code is not efficient, does unnecessary processing or blocks the UI thread.
4. Please download the support files provided with this assignment and use them when implementing your project.
5. Export your Android project and create a zip file which includes all the project folder and any required libraries.
6. Submission details:
   a. Only a single group member is required to submit on moodle for each group.
   b. The file name is very important and should follow the following format: **Group#_HW09.zip**
   c. You should submit the assignment through Moodle: Submit the zip file.
7. **Failure to follow the above instructions will result in point deductions.**

Homework 9 (100 Points)

In this assignment you will implement an app called “Stay in Touch” to enable users to have conversations with other users, make calls to these users, and see their own contact information using fragments. You will use Firebase to store/retrieve the messages and users’ information.



(a) Login fragment

(b) SignUp fragment

Figure 1, Wireframe for Login and SignUp fragments

Part A: User Signup and Login (10 Points)

Your app should implement both login and signup functionalities. You should use Firebase to register user using email and password and also store the user’s username, phone number, email address, password, profile image (base 64 encoded string) in the User object. The requirements are as follows:

1. Make sure to enable Email & Password Authentication, just as in HW8.
2. The launcher activity should be set to the Login fragment. When the app first starts, the Login fragment should check if there is a current user session, by using the Firebase provided methods to check if there is a valid current user:
   a) If there is a current valid user, then start the Conversation fragment, and finish the Login fragment.
   b) If there is no current valid user, then load the Login fragment.
3. Create a Login fragment (Figure 1(a)):
   a) Clicking the “Login” button should submit the provided email and password information to Firebase to verify the user’s credentials.
      - If the user is successfully logged in then start the Conversations fragment, and finish the Login fragment.

    - If the user is not successfully logged in, then show a toast message indicating that the login was not successful.
  b) Clicking the “Create New Account” button should start the Signup fragment and finish the login fragment.
4. Create a Signup fragment (Figure 1(b)):
  a) Clicking the “Cancel” button should finish the Signup fragment and start the Login fragment.
  b) Clicking the “Sign Up” button should submit the user’s provided information to Firebase to verify the user’s credentials.
    - If an account with the same email already exists, display an error message indicating that the account was not created and the user should select a different email.
    - If an account with the provided credentials does not already exist, then store the new account information and display a Toast indicating that the user has been created. Then start the Login fragment and finish the Signup fragment.
    - Create a User object in Firebase with name, email, password, phone number, and a default profile picture in base 64 encoded string format.
      - Note that on creating a user, the user data should explicitly be stored in a Firebase Users object as data, as this is not done automatically through email/password authentication in Firebase.

<u>Part B: Navigation Drawer (20 points)</u>

The main activity should be having a navigation drawer that helps to navigate to different fragments. Please find this documentation for your reference:

<u>http://developer.android.com/training/implementing-navigation/nav-drawer.html</u>

The requirements are as following:

- On Clicking on the drawer icon (top – left) on the main activity should open the navigation drawer as shown in the figure 2a / 2b.
  - Navigation drawer should have the profile picture of the current logged in user and the user full name.
  - Clicking on **Contacts** option should load the Contacts fragment as shown in the figure 2c and 2d.
  - Clicking on **Conversations** option should load the Conversations fragment as shown in the figure 2e and 2f.
  - Clicking on **Archived** should load the Conversations fragment, loading only the archived ones, as shown in the figure 2g and 2h
  - Clicking on the **Settings** should open the EditProfile fragment as shown in the figure 2i and 2j
  - Clicking on **Logout** option should logout the current user, start the Login Fragment and finish the Conversations fragment.
  - Clicking on **Exit** should finish the application.

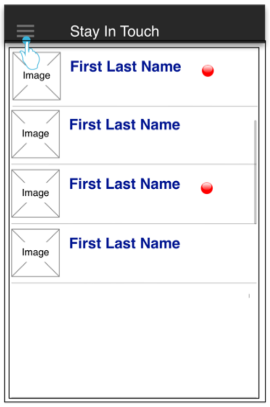


Figure 2a, Click on the drawer icon to open navigation drawer

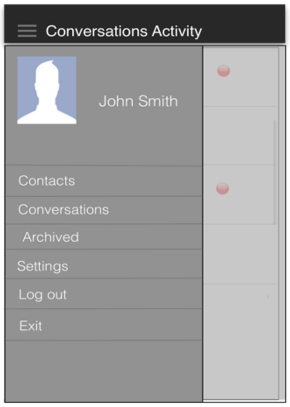


Figure 2b, Navigation drawer

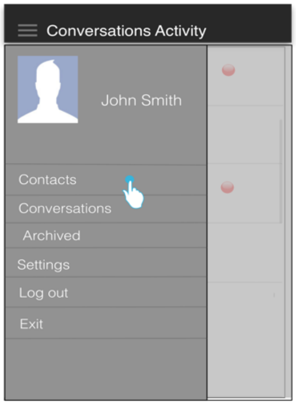


Figure 2c, Clicking on contacts should open Contacts fragment

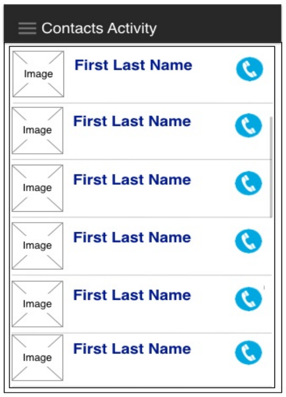


Figure 2d, Contacts fragment

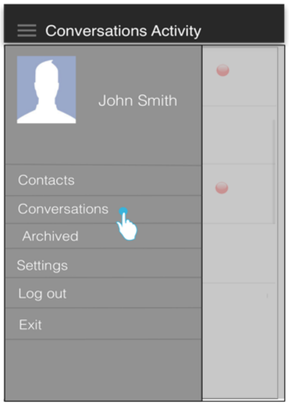


Figure 2e, clicking on conversations should open conversations fragment

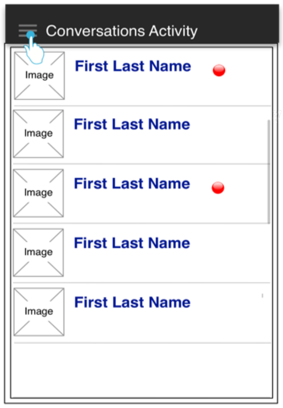


Figure 2f, Conversations fragment

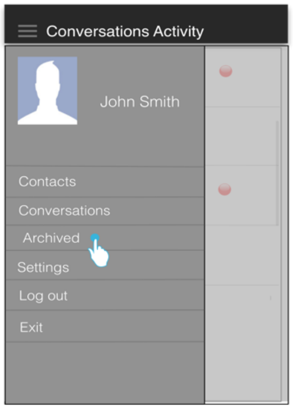


Figure 2g, Clicking on achieved should open Conversations fragment with conversations that are archived

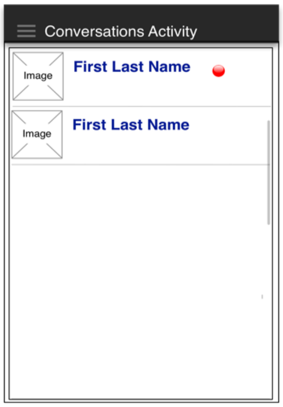


Figure 2h, Conversations fragment with conversations that are archived

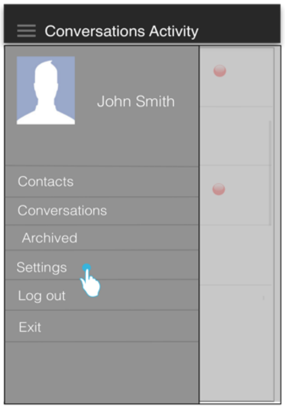
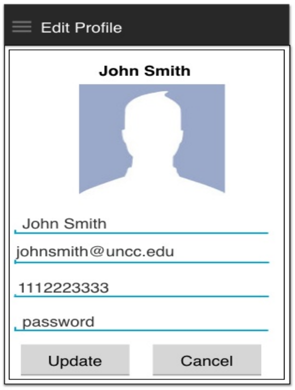


Figure 2i, clicking on Settings should open Edit Profile fragment



John Smith

johnsmith@uncc.edu

1112223333

password

Figure 2j, Edit Profile

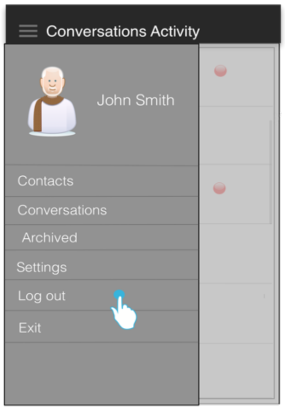


Figure 2k, Clicking on logout will open the Login fragment

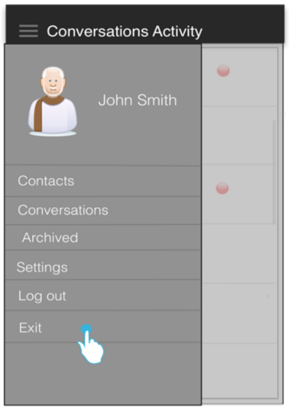


Figure 2l, Clicking on exit will finish the application

Part C: Contacts Fragment (10 Points)

Contacts fragment should be opened when user clicks on the option “Contacts“ as shown in the figure 3a and 3b. The requirements are as follows:

- Contacts fragment should have a ListView of all the users (except the current logged in user), with profile picture, full name and a call button
- On Click of Call image button, should initiate a call using the implicit dialing intent with the contact person phone number as shown in the figure 3e / 3f.
- On Clicking on the contact should load the View Messages fragment, fig 3d
- On Long Click of a contact should load the View Contact fragment

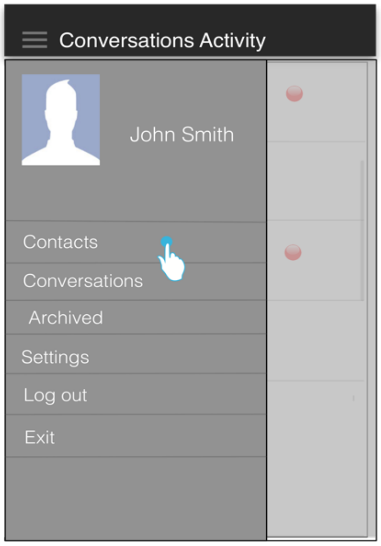


Figure 3a, clicking on contacts should open Contacts fragment

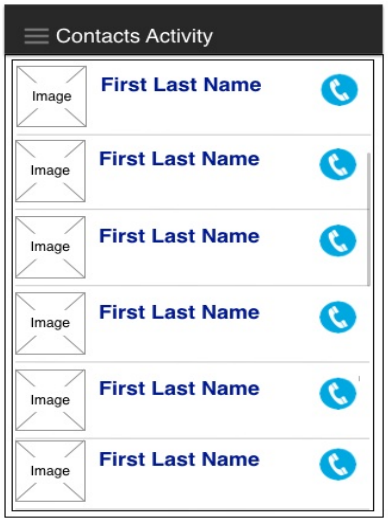


Figure 3b, Contacts fragment

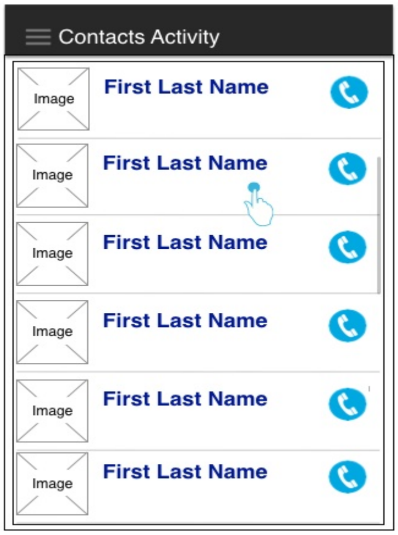


Figure 3c, clicking on contacts should open the view messages fragment

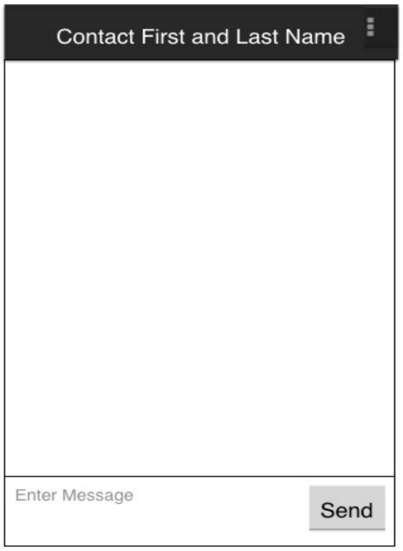


Figure 3d, view messages fragment

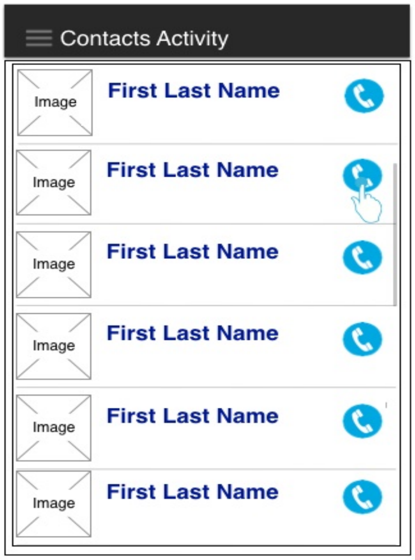


Figure 3e, clicking on the call button initiates the call

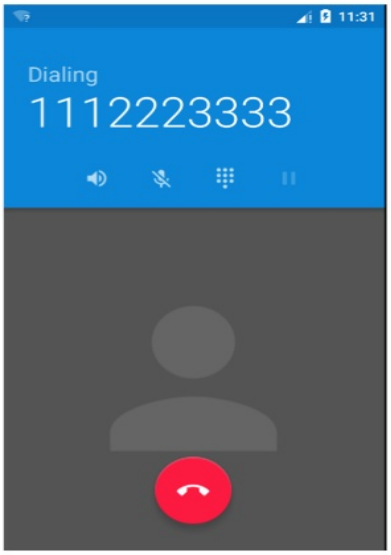
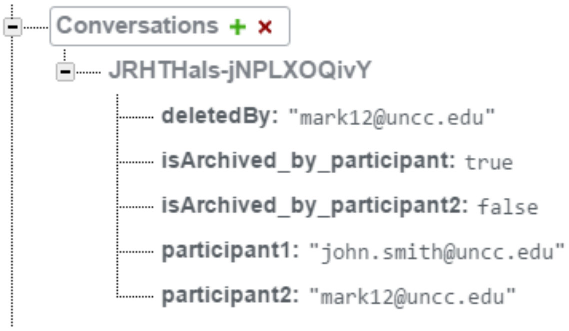


Figure 3f, Calling the contact

Part D: Conversations Fragment (20 Points)

The Conversations fragment should show the list of contacts created in Firebase other than the current logged in user. The requirements are as following.

1. Firebase should contain **Conversations** objects with following fields:
   - **conversation ID** – unique id generated while creating the conversation
   - **participant1** – participant 1 (userID/email) involved in the conversation
   - **participant2** – participant 2 (userID/email) involved in the conversation
   - **deletedBy** – participant (userID/email) who has deleted this conversation
   - **IsArchived_by_participant1** – participant1(userID/email) who has archived this conversation
   - **IsArchived_by_participant2** - participant1(userID/email) who has archived this conversation



```
Conversations + ×
  JRHTHals-jNPLXOQivY
    deletedBy: "mark12@uncc.edu"
    isArchived_by_participant: true
    isArchived_by_participant2: false
    participant1: "john.smith@uncc.edu"
    participant2: "mark12@uncc.edu"
```

2. Get the list of conversations from Firebase which are not deletedBy nor isArchivedBy the current user, where he/she is either participant 1 or 2.
3. Each list item in the ListView should contains the **profile picture and full name of the participant** who is not the current logged in user in the conversation object. It should also have unread message symbol (red dot icon) if applicable, as shown in the figure 4a.
4. If the other participant has sent a message to this user and if the message is unread (check message_read field in message object) then show the **red dot icon** for the conversation. This is to identity the unread messages.
5. **On Click** of the list item should open the ViewMessages fragment for that conversation.
6. If the contact had unread messages, then upon opening the ViewMessages fragment for that conversation please mark the message_read as true, so that next time when the current logged user opens Conversations Fragment he/she will not see the red icon for that contact.
7. **On Long Click** of the list item, display an alert dialog with two options: “Archive Conversation” and “Delete Conversation”
   a) If “Archive” is selected: check if the current logged in user is participant1 or paticipant2, and set either IsArchieved_by_participant1/ IsArchieved_by_participant2 as true/false.

b) If “Delete” is selected: set the conversation as deletedBy this user, and close this fragment and load the Conversations fragment with the updated list. If this Conversation was already marked as deletedBy the another user, then simply delete this Conversation object from Firebase.

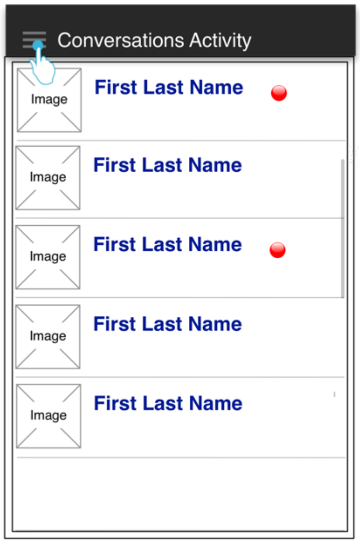


Figure 4a, Conversations fragment

Part E: Edit Profile Fragment (10 Points)

The Edit Profile fragment should be displayed when user clicks on the “Settings” option in navigation drawer. The requirements are as follows:

1. The Edit Profile fragment should look like Figure 5(b). It should display the name, email, phone number, password and the current profile picture of the current logged in user.
2. The values should be populated for the current logged in user.
3. User should be able to change the profile picture by clicking on the picture as shown in the Figure 5(b) and uploading from the device’s photo gallery. The image should be saved as base64-encoded String into Firebase user object.
4. Clicking on the update button should save the changes to the Firebase for that user and the changes should be reflected for the current logged in session.
5. Clicking on the Cancel button should dismiss all the changes, also should open the Conversation fragment and finishes the Edit profile fragment (same should happen up on back button press)
6. Upon successfully updating the data, a toast message should be displayed saying that changes have been successfully updated and should close the Edit Profile fragment and open Conversations Fragment, see Figure5e.

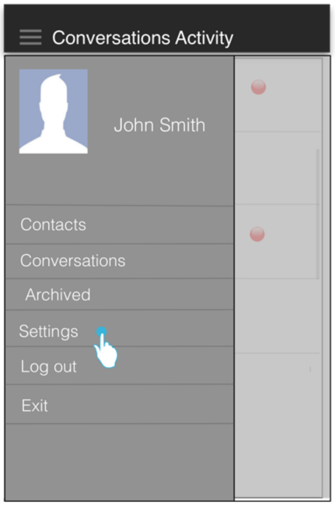
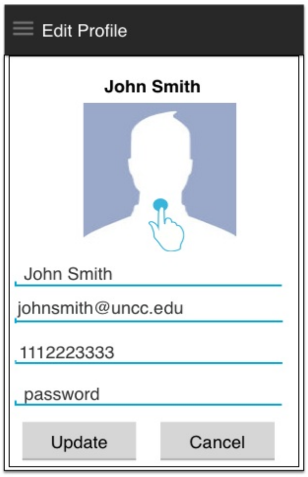


Figure 5a, On clicking settings opens the Edit Profile acitivity



John Smith

johnsmith@uncc.edu

1112223333

password

Figure 5b, For changing profile picture click on the default picture

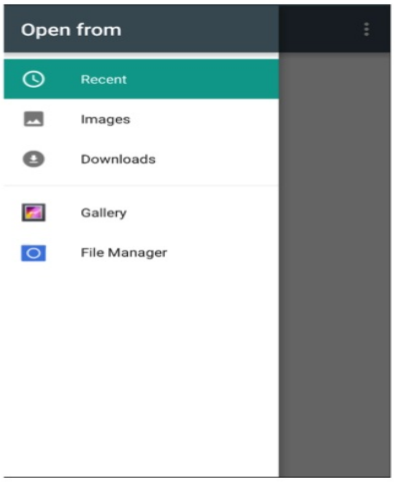


Figure 5c, Select gallery

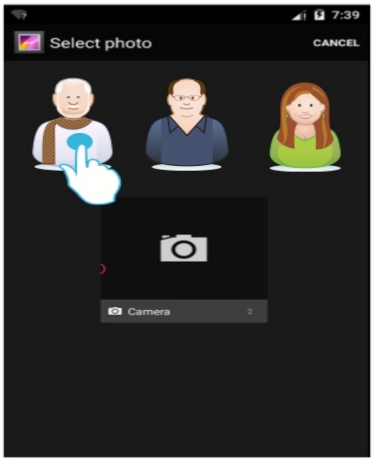
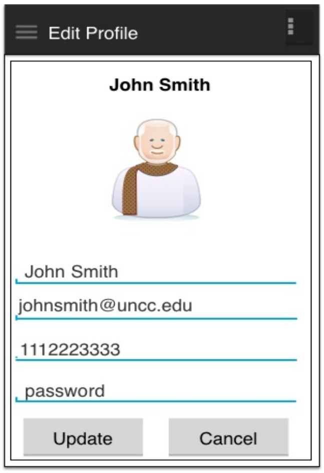


Figure 5d, Select Picture



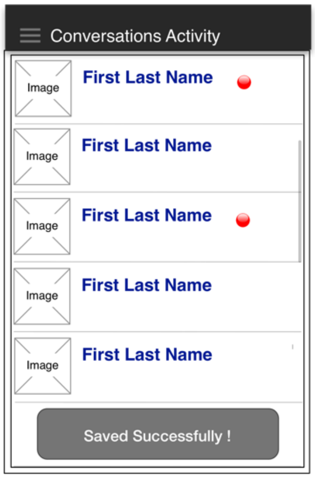
John Smith

johnsmith@uncc.edu

1112223333

password

Figure 5e, Click on update and validate the data entered



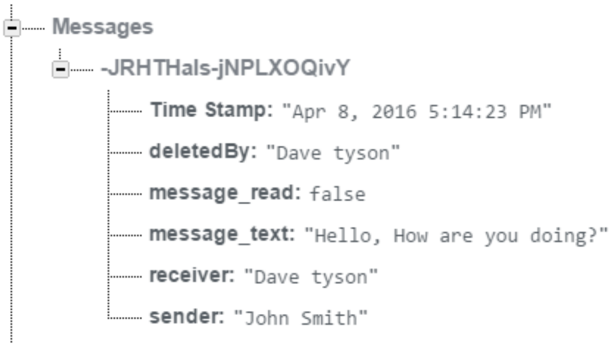
Saved Successfully !

Figure 5f, showing toast message upon successful updating

Part F: View Messages Fragment (25 Points)

The View Messages fragment should retrieve all the messages that are stored in the messages object on Firebase that contains logged in user and a particular contact person as sender or receiver. A ListView should display the items in the messages list, see Figure 6(a). The requirements are as follows:

1. In Firebase create a messages object to store the messages.
2. Each message object should contain sender username, receiver username, message text, message read, timestamp as shown in the below figure.



```
Messages
  -JRHTHals-jNPLXOQivY
    Time Stamp: "Apr 8, 2016 5:14:23 PM"
    deletedBy: "Dave tyson"
    message_read: false
    message_text: "Hello, How are you doing?"
    receiver: "Dave tyson"
    sender: "John Smith"
```

3. By default the ListView should be populated with all the messages for the selected conversation between the current user and the receiver where deletedBy is null or the receiver's info (userid/email. Message objects with deletedBy set to currently logged in user should not be displayed.
4. A Message object will be created when the current logged in user will compose the message in the text box located in the bottom of the fragment and by clicking on **send** button. The message will be created with current logged in user as sender and the current contact as receiver.
5. If the message entered is the first message between the sender and receiver vice versa then create a new Conversation object for sender and receiver as participant1 and participant2.
6. Message object should make the message_read as false while creating.
7. The message text should have a maximum of 140 characters. You should validate the user's input and ensure that the text is provided. If no input is provided, display a Toast.
8. Upon successful creation of the message in Firebase, the ListView should update to show the newly created message.
9. The messages from sender and receiver should be differentiated. For example, if the message being viewed was sent by the currently logged in user, then the background color should be grey, and otherwise, normal white background.

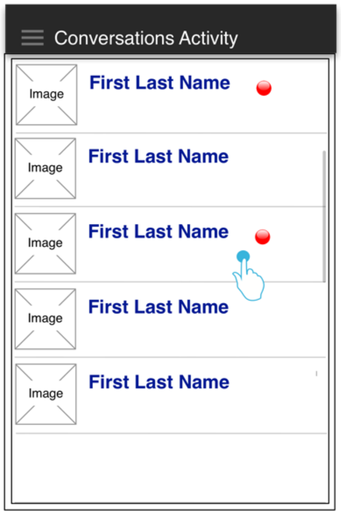
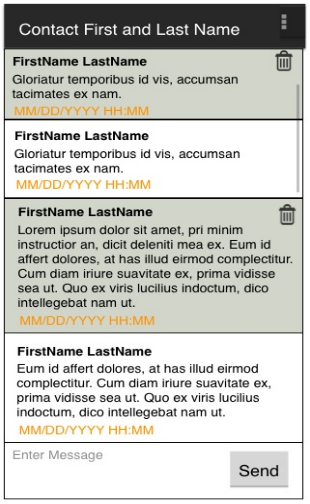


Figure 6a, On Clicking the particular contact will open the View messages



FirstName LastName
Gloriatur temporibus id vis, accumsan tacimates ex nam.
MM/DD/YYYY HH:MM

FirstName LastName
Gloriatur temporibus id vis, accumsan tacimates ex nam.
MM/DD/YYYY HH:MM

FirstName LastName
Lorem ipsum dolor sit amet, pri minim instructior an, dicit deleniti mea ex. Eum id affert dolores, at has illud eirmod complectitur. Cum diam iriure suavitate ex, prima vidisse sea ut. Quo ex viris lucilius indoctum, dico intellegebat nam ut.
MM/DD/YYYY HH:MM

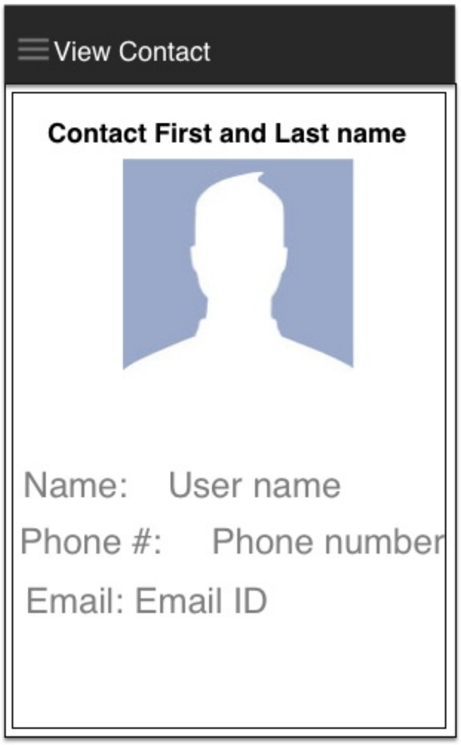
FirstName LastName
Eum id affert dolores, at has illud eirmod complectitur. Cum diam iriure suavitate ex, prima vidisse sea ut. Quo ex viris lucilius indoctum, dico intellegebat nam ut.
MM/DD/YYYY HH:MM

Figure 6b, View Messages Activity

Part G: View Contact Fragment (5 Points)

The View Contact fragment should be displayed once user long clicks on a Contact from the Contacts fragment. View Contacts fragment is displayed in Figure 7a. Implement the following requirements:

- View Fragment displays full name, profile picture, phone number, and email
- On clicking on the back button should go back to Contacts fragment of the contact and finish the view contact fragment.



Name: User name

Phone #: Phone number

Email: Email ID

Figure 7a, View Contact fragment