**Software Implementation and Testing Document**

**For**

**Group 21**

Version 3.0

**Authors**:

Leylanni Quijano-Shafer

Felipe Bergano

Liz Parra

Andres Paz Vicca

Raymond Chen

# Programming Languages

We are using Java for the entire framework of the app since it is the easiest for us to use because some members are already familiar with it and due to the fact that we are using Android Studio to implement the app, it makes sense to. We are also using Java for the interaction with the database via Google Firebase, for the same familiarity and ease of use reasons previously stated.

# Platforms, APIs, Databases, and other technologies used

We used elements native to Android Studio for the majority of our app. We are using Google Firebase to create our database and we are using Google Maps API for the location services our app provides. We are also using Google Firebase for user authentication and for database management.

# Execution-based Functional Testing

1. **The user is able to input their email, password, name, and upload a picture to create an account. High Priority**

Tested using Android Studio’s Android Emulator and with a physical Samsung Galaxy S7 running Android 8. Tested by entering different combinations of valid and invalid data to ensure proper functionality and implemented restrictions properly.

1. **User is able to input their email and password to login to their account.**

Tested using Android Studio’s Android Emulator and with a physical Samsung Galaxy S7 running Android 8. After creating an account, we logged out of the account and then attempted to log back into the same account to ensure that a user could log back in after account creation.

1. **The system allows for the creation of a group of users. High Priority**

I added people to join my group and shared my code to users in many accounts to verify the groups were being made. Also checked in the database that the groups contained only the people permitted to join by sharing unique code ids. Tested with Android Studio’s Emulator, and with a Physical Samsung Galaxy S7.

1. **The system tracks the user in real-time and uploads their location to the database**

Used the Android emulator provided by Android Studio and fed it many different location data to ensure accuracy and that user location data is correctly uploaded to Firebase. Tested with Android Studio’s Emulator, and with a Physical Samsung Galaxy S7

**5) A group of users can chat with each other in a group chat. High Priority**

Tested with Android Studio’s Emulator, and with a Physical Samsung Galaxy S7

Users of the app can click on the Chat w/Friends and text the group of users, then they can click on enter a message, type what they want, and click the green arrow to send it. The messages are uploaded into the database and properly displayed and stored there, it says who sent it and at

**6) The system has a button that sends an SOS alert to others in the group if it is not canceled within 5 seconds after pressing the corresponding button. High Priority**

Under the emergency alerts fragment, users can request to seek help from the members of the circle, a 5-second countdown begins once the bottom is pressed and the user has the option of canceling the action if it was pressed by mistake, if the user chooses to go ahead with it, an alert will be sent and other users will be able to track him/her. We tested it by having the app open simultaneously on the phone and the computer and sending/receiving the notification. The alert successfully displayed on firebase. Tested with Android Studio’s Emulator, and with a Physical Samsung Galaxy S7

**7) The user is able to upload a profile picture as the second step of registering. Low Priority**

Tested it by creating a new account with the option of uploading a profile picture. Tested with Android Studio’s Emulator, and with a Physical Samsung Galaxy S7. Verified that the profile picture is uploaded by checking the database via the Firebase Console.

**8) The system allows the user to choose when to start sharing their location after pressing the share button.**

Tested using Android Studio’s Android Emulator and with a physical Samsung Galaxy S7 running Android 8. Under my group fragment, you can see those who are in your group and have a status next to their name (green circle when sharing) of when they are currently sharing their location. You can click on their name and then see the user’s location. We also checked with the database that the location was being recorded.

**9) The system allows the user to choose when to stop sharing their location after pressing the stop sharing button.**

Tested using Android Studio’s Android Emulator and with a physical Samsung Galaxy S7 running Android 8. Under my group fragment, you can see those who are in your group and have a status next to their name (red circle when stop sharing) of when they are not currently sharing their location. You can click on their name and then not see the user’s location on the map. We also checked with the database that the location was not being recorded/updated.

**10) The system allows the user to share their unique code through messaging platforms to others by pressing the share button.**

Under the Join group fragment, users can see what’s their unique shareable ID and decide to share it by either word of mouth or by sharing it with other applications in their phone such as WhatsApp or text message, if they choose to share it through text, then the recipient will receive a text saying “here is my ID, join my circle”. We tested this during our demo by using Andres’s phone to send a shareable invite to Liz and it sent successfully.

Tested using Android Studio’s Android Emulator and with a physical Samsung Galaxy S7 running Android 8.

**11) The system allows the user to input a user’s unique code to add them to the group by pressing the join button.**

Under the Join group fragment, users can input the unique shareable ID at the bottom of the screen, if the code is incorrect then the user will be prompted to input a correct code until it is right, we tested this in the database and once it became successful, the group of users displayed correctly in Firebase.

Tested using Android Studio’s Android Emulator and with a physical Samsung Galaxy S7 running Android 8.

**12) The system allows you to view who is in your group.**

Under My circle fragment, users can see which members are currently under your group, the Firebase displayed the correct members when we were testing it as it was only those who had inserted the shareable code

Tested using Android Studio’s Android Emulator and with a physical Samsung Galaxy S7 running Android 8.

**13) The system allows the user to click on someone in their group and see their location.**

Under My circle fragment, users can see which members are currently sharing their location in the group and if you click on one of them, you can see where they are. We tested this by having multiple users, some of them allowed the location to be shared and others did not, we could only see the location of those users who enable location sharing. the location on firebase indicated that it was the right coordinates. Tested using Android Studio’s Android Emulator and with a physical Samsung Galaxy S7 running Android 8.

**14) The system allows the user to press on log out and log out of the system by pressing the log out button.**

Under the Emergency Alerts fragment, you can click on the logout button to have the account signed off the app, this will take the user to the login page where they can decide if they want to sign in again or not. we tested this button by seeing what would happen after clicking it and we did lose access to the account once we clicked on it.

Tested using Android Studio’s Android Emulator and with a physical Samsung Galaxy S7 running Android 8.

**15) The system allows the user to start the reminders to drink water by pressing the corresponding button.**

Under the Emergency Alerts fragment, you can click on the Set Every 30 Min Reminder button and it will pop up a notification in the user’s notification screen, after 30 mins, a new notification will ring displaying the message “It’s now: ‘The time’. DRINK WATER!”, this message is also stored in the database.

Tested using Android Studio’s Android Emulator and with a physical Samsung Galaxy S7 running Android 8.

**16) The system allows the user to stop the reminders to drink water by pressing the corresponding button.**

Under the Emergency Alerts fragment, you can click on the Cancel Reminder button and it will stop pop up notifications from showing up in the users’ notification screen

Tested using Android Studio’s Android Emulator and with a physical Samsung Galaxy S7 running Android 8.

**17) The user is able to view users who sent an SOS alert by pressing the corresponding button.**

Tested using Android Studio’s Android Emulator and with a physical Samsung Galaxy S7 running Android 8. Under Emergency Alerts fragment, you can click on the Send Help Alert button and seek help from group members, a 5 second countdown will begin and everyone in the circle will receive a notification from the endangered user, we tested this by creating multiple accounts and checking if all of them got a notification when an alert was sent, once clicked, this notification sends the user to the screen with the last location seen of the alert. Additionally, we checked the database, and it updates the message by the user in the circle once a notification is sent with the date and time of last seen.

**18) The user is able to click on a user who sent an alert and see their last location on the map.**

Tested using Android Studio’s Android Emulator and with a physical Samsung Galaxy S7 running Android 8. Under Emergency Alerts fragment, you can click on the Alert Center button and monitor all threats, choose the alert from the list and check on them, we tested this by creating multiple accounts and checking if all of them got a notification when an alert was sent, once clicked, this notification sends the user to the screen with the last location seen of the alert. Additionally, we checked the database, and it updates the data of the user in the circle once a notification is sent.

**19) The user is able to click on a user who sent an alert and see when they were last seen logged on into the account on the map.**

Tested using Android Studio’s Android Emulator and with a physical Samsung Galaxy S7 running Android 8. Under Emergency Alerts fragment, you can click on the Alert Center button and monitor all threats, choose the alert from the list and check on them, we tested this by creating multiple accounts and checking if all of them got a notification when an alert was sent, once clicked, this notification sends the user to the screen with the last time they were seen of the alert. Additionally, we checked the database, and it updates the data of the user in the circle once a notification is sent.

Tested using Android Studio’s Android Emulator and with a physical Samsung Galaxy S7 running Android 8.

**20) The user is able to click on a user who sent an alert and be able to click the google maps button which will open up google maps with directions to the user’s location.**

Under Emergency Alerts fragment, you can click on the Alert Center button and monitor all threats, choose the alert from the list and check on the last seem, at the bottom right of the screen, a user can click on the google maps icon and this will open google maps with directions to the user’s location. This was tested by having multiple users check the correctness of the address and it was always the same and correct.

Tested using Android Studio’s Android Emulator and with a physical Samsung Galaxy S7 running Android 8.

**21) The user is able to click on the forgot password button and change their password by inputting a new password**

On the sign-in screen, a user can click on forgot my password at the bottom of the screen and it will prompt them to insert the new password, this was tested when we clicked on it, assigned a new password, and signed in with the new login information. Tested using Android Studio’s Android Emulator and with a physical Samsung Galaxy S7 running Android 8.

**22) The user is able to delete messages sent only by them by pressing on the profile icon next to the message**

On the chat w/friends fragment, a user can click on the trash button next to his/her own text and the message will be deleted from the user’s screen and from the screen of the people in the circle. we tested this by logging in with multiple accounts and checking if the message still showed up, additionally, the message got deleted from the database. Tested using Android Studio’s Android Emulator and with a physical Samsung Galaxy S7 running Android 8.

When trying to implement my version of the login/signup interface, I kept trying different options with how to implement it. I would run the code and input different users, passwords, phone numbers, and emails, and see how these would get added to the database. Unfortunately on my version, I was not able to get the firebase to add the user with email and password, as well as add user data simultaneously. And so it was left as extra code. //Felipe Bergano

# Execution-based Non-Functional Testing

Scalability:

* 1. Verify that the app can work with more than a few users, we tested this by creating different accounts and ensuring that they were added to the database successfully

Security:

1. Verify that email is in the correct format, we verified this by trying emails without the ‘@’, ‘.com’, and other formats that were not a correct email.
   1. Verify that the password meets specific security requirements when registering, we tested this by trying to register with a password that is too common, such as ‘password,’ trying a password that is less than 6 characters, etc.
   2. Verify user exists in the database when trying to log in, invalid accounts are not able to log in and are told to verify the sign-in information
   3. Verify the user’s location is only visible to others in the same group, we checked that this was correct by creating new accounts and were not able to track users who were not in our circle, additionally, firebase successfully displayed that its members were part of a specific circle
   4. Verify the user is inputting the right password when logging in. When the password is not valid, a message pops up saying that the login information is incorrect
   5. Verify that the user has selected a profile picture at the moment of registration, the user is not able to register unless a picture is uploaded

Reliability:

1. Verify that the location is shared while the app runs in the background, this was tested by having the app run while the screen is off and having the user move its real-life location.

Accessibility:

1. Verify that the app works in more than one device, we tested it in both the emulator and on a physical Samsung Galaxy S7 phone

Reusability:

1. Verify that the app works in multiple android versions, we tested it in API 26-28 and android versions 7 and 8.

Data Integrity:

1. Verify that Data is stored as a private instead of public, we rewrote the permissions in Google Firebase and set it to private

# 5. Non-Execution-based Testing

We reviewed each other’s code when we tried to implement new features to make sure we understand what is trying to be added and point out any mistakes or unnecessary code. We made sure that things were being sent and retrieved from the database in real-time by having someone inspect the database as someone made changes to the data. We ensured to debug during our process of coding and check if there were any errors in the code that were not discovered. Additionally, we peer-reviewed changes in the database as it was causing errors.

Under this requirement, in order to complete the Class diagram, a thorough review of the code had to be made. That way we understood how the different pieces connected and we were able to place all the pieces together.