**Software Requirements and Design Document**

**For**

**Group 21**

Version 3.0

**Authors**:

Leylanni Quijano-Shafer

Felipe Bergano

Liz Parra

Andres Paz Vicca

Raymond Chen

* **Overview**

We aim to create a mobile app that allows a group of people that are headed for a night out to remain connected all in one place. The app would allow users to make groups through which they can share their location and chat with the other members.

* **Functional Requirements**

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

*2) User is able to input their email and password to login to their account. High Priority*

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

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

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

*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*

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

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

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

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

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

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

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

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

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

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

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

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

*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.*

*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.*

*21) The system checks to make sure the email address used to sign up is unique and does not already exist in the database. High Priority*

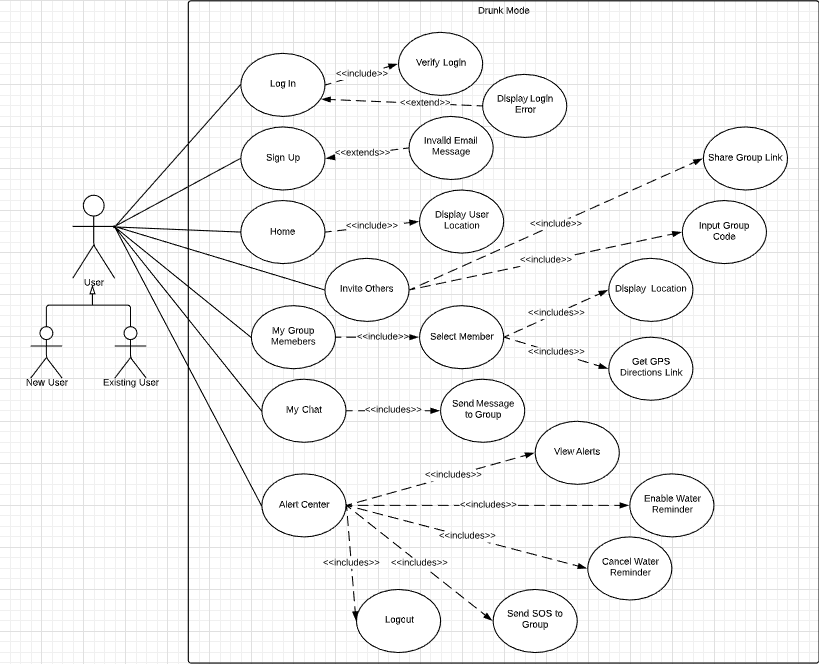
*22) The system verifies that the user exists by making sure the email address entered to sign in exists in the database. High Priority*

*23) The system secures user accounts by making sure passwords are at least six characters long and are not the password. Users can only sign in using correct credentials. High Priority*

**Non-functional Requirements**

1. Scalability:
   1. Verify that the app can work with more than a few users
2. Security:
   1. Verify that email is in the correct format
   2. Verify that the password meets specific security requirements when registering
   3. Verify user exists in the database when trying to log in
   4. Verify the user’s location is only visible to others in the same group
   5. Verify the user is inputting the right password when logging in
   6. 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
3. Reliability:
4. Verify that the location is shared while the app runs in the background
5. Accessibility:
6. Verify that the app works on more than one device
7. Reusability:
8. Verify that the app works in multiple android versions
9. Data Integrity:
10. Verify that Data is stored as a private instead of public

**Use Case Diagram**

****

**Use Case Diagram Textual Descriptions**



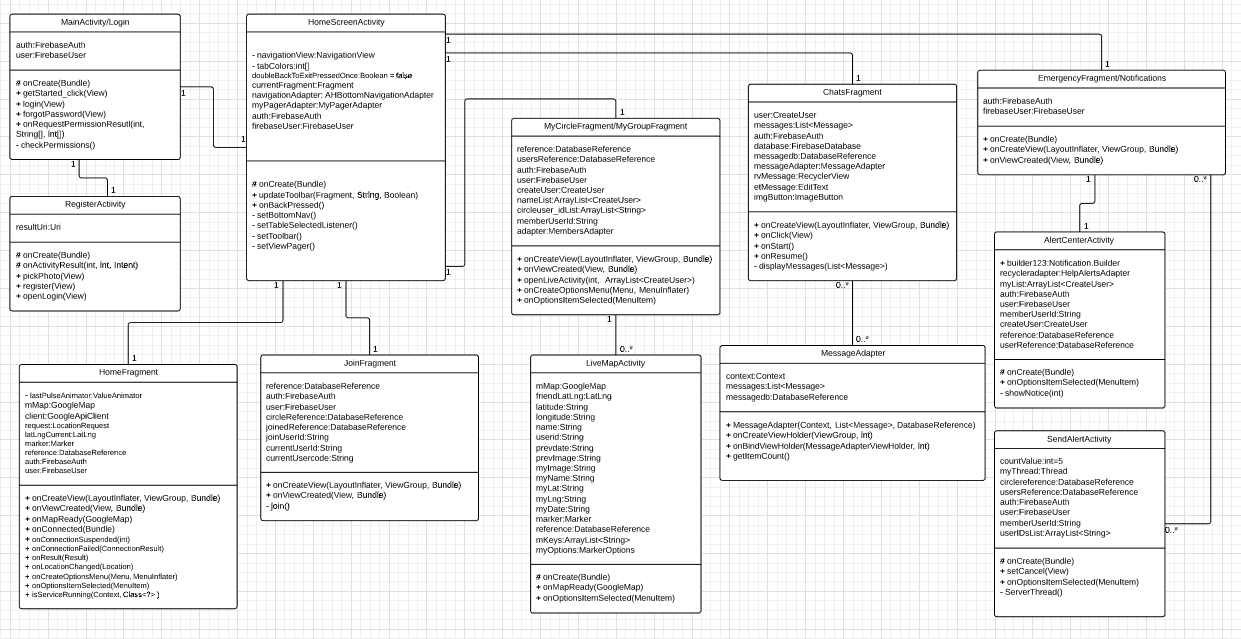








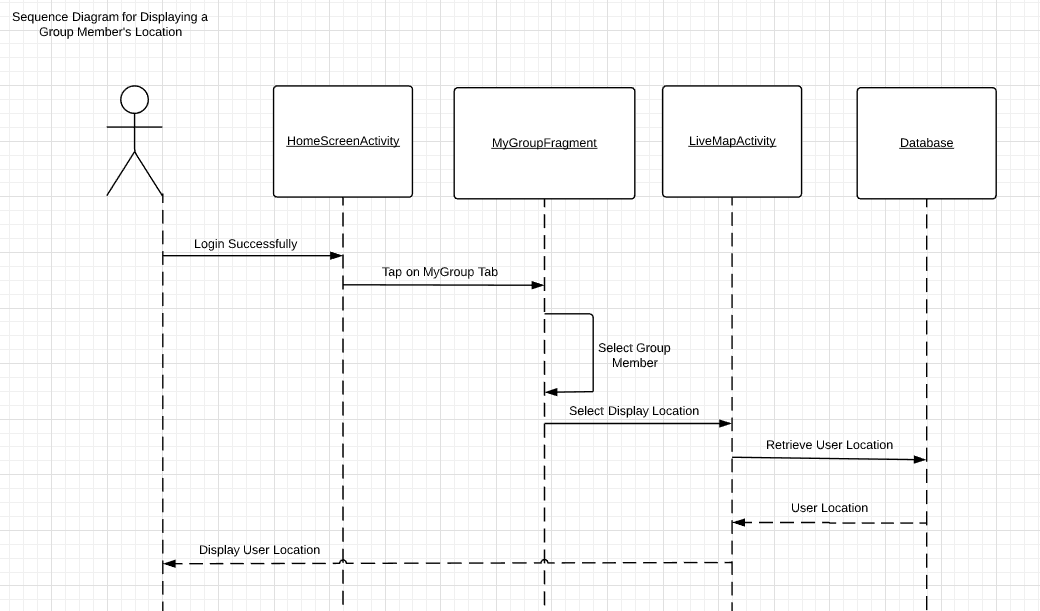
* **Class Diagram**

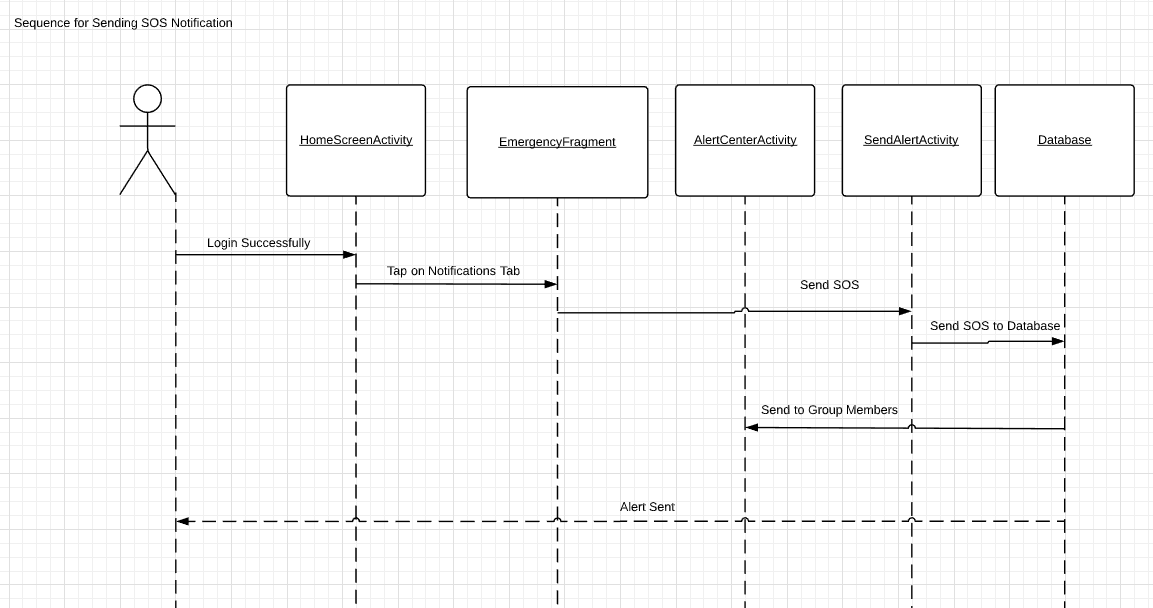
****

* **Sequence Diagrams**

**A close up of a map

Description automatically generated**

****

****

* **Operating Environment**

The system will operate on Android phones, from Android API16 forward.

* **Assumptions and Dependencies**

We expect users to use a valid email address since it needs to be verified for their account. We are also assuming that the user’s phone is able to support all app features on its operating system version.