**Software Requirements and Design Document**

**For**

**Group 21**

Version 1.0

**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) User is able to create an account that will allow them access to groups that they are in. High Priority

2) System allows for the creation of groups of users. High Priority

3) System tracks user in real time and uploads their location to the database

4) Groups of users can chat with others in the group and also see the locations of group members. High Priority

5) System has a button that sends an SOS message to others in the group. High Priority

* **Non-functional Requirements**

*List the* ***non-functional requirements*** *of the system (any requirement referring to a property of the system, such as security, safety, software quality, performance, reliability, etc.) You may provide a brief rationale for any requirement which you feel requires explanation as to how and/or why the requirement was derived.*

* **Use Case Diagram**



* **Class Diagram and/or Sequence Diagrams**

*This section presents a high-level overview of the anticipated system architecture using a* ***class******diagram*** *and/or* ***sequence diagrams****.*

*If the main* ***paradigm*** *used in your project is* ***Object Oriented*** *(i.e., you have classes or something that acts similar to classes in your system), then draw the* ***Class Diagram******of the entire system and Sequence Diagrams for the three (3) most important use cases in your system.***

*If the main* ***paradigm*** *in your system is* ***not Object Oriented*** *(i.e., you* ***do not*** *have classes**or anything similar to classes in your system) then only draw* ***Sequence Diagrams****,* ***but for all the use cases of your system.*** *In this case, we will use a modified version of Sequence Diagrams, where instead of objects, the lifelines will represent the functions in the system involved in the action sequence.*

***Class Diagrams*** *show the* ***fundamental objects/classes*** *that must be modeled with the system to satisfy its requirements and* ***the relationships*** *between them. Each class rectangle on the diagram* ***must also include the attributes and the methods of the class*** *(they can be refined between increments). All the* ***relationships between classes and their multiplicity*** *must be shown on the class diagram.*

*A* ***Sequence Diagram*** *simply depicts* ***interaction******between objects*** *(or* ***functions -*** *in our case - for non-OOP systems) in a sequential order, i.e. the order in which these interactions take place. Sequence diagrams describe how and in what order the objects in a system function.*

**Operating Environment**

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

**Assumptions and Dependencies**

For user testing purposes, we expect users to use a real phone number since it needs to be verified. For our first iteration we expect the use to use a gmail account if they choose email as the preferred registration method. We are also assuming that the phone number will be a U.S number and that it will stay logged in until the app is deleted.