**1. Introduction**

**1.1 Purpose**

The use case document describes the specification along with some requires of a console application Study Group Former (SGF). In addition, this document will describe how the program works and what requirements are necessary in the use case specification.

**1.2 Background**

Often times, students find it hard to meet new people in their classes, or find it difficult be able to connect with people of the same classes who are suffering from the same difficulty of the same topic of a course. This desktop application will provide a way for these individuals to get together, across different sections of a class and review the subject.

**1.3 Definition, Acronyms, and Abbreviations**

Study Group Former SGF

**2. System Requirements**

See Attached documentation (System Reqs.pdf)

**2.1 Current “System” (“App”)**

Currently study groups are formed when students are presented with opportunity to introduce, meet, or talk with one another. Therefore, these are informally formed by an individual’s desire to have a group dedicated to do so.

**2.2 Actors**

Students and/or professors will be held accountable for formation of groups and setting a limit or cap on how many attendees there may be. The Students and professors will be of type “User” while administrators will be of type “Admin.” Administrators have more access to the system can perform tasks that students and professors cannot.

**2.3 Basic Use Cases**

2.3.1 User logs in to the system

2.3.1.1 User has a username and password

2.3.1.1.1 User enters username and password

2.3.1.1.2 User clicks login

2.3.1.2 App sends login data to database server

2.3.1.3 App responds

2.3.1.3.1 Server Response: Credentials Correct, App retrieves JSON response

2.3.1.3.2 Server Response: Credentials Incorrect, App informs user

2.3.1.4 User info is updated from JSON response

2.3.1.5 App requests User groups, all available groups

2.3.1.6 Server Responds and sends JSON objects

2.3.1.7 App updates User groups, all available groups

2.3.1.7.1 If user is type Admin, set flag to allow access to admin functions

2.3.2 User updates available groups

2.3.2.1 User clicks update available groups

2.3.2.2 App requests all available groups

2.3.2.3 Server Responds and sends JSON object

2.3.2.4 App updates all available groups

2.3.3 User updates “their” groups

2.3.3.1 User clicks update my groups

2.3.3.2 App requests all User groups as JSON object from Server

2.3.3.3 Server Responds and sends JSON object

2.3.3.4 App updates User groups

2.3.4 User creates a group

2.3.4.1 User clicks create a group

2.3.4.2 App opens “create group window”

2.3.4.3 User inputs group data and hits create

2.3.4.4 App sends request to create group as JSON object to the server

2.3.4.4.1 Server Response: Failure, App informs user to try again

2.3.4.4.1 Server Response: Success, App informs user

2.3.4.5 App displays new group in User Groups window

2.3.5 User joins a group

2.3.5.1 User selects an existing group and clicks join

2.3.5.2 App sends request to join a group to server

2.3.4.4.1 Server Response: Failure, App informs user why

2.3.4.4.1 Server Response: Success, App informs user and saves JSON data

2.3.5.3 App updates User groups with JSON data, displays group in User Groups window

2.3.6 User quits a group

2.3.6.1 User selects an existing group and clicks quit

2.3.6.2 App sends request to quit a group to server

2.3.4.4.1 Server Response: Failure, App informs user why

2.3.4.4.1 Server Response: Success, App informs user

2.3.6.3 App updates User groups, displays User Groups window

2.3.7 User deletes a group they have created

2.3.7.1 User selects an existing group and clicks delete

2.3.7.2 App sends request to delete group as JSON object to the server

2.3.7.2.1 Server Response: Failure, App informs user why

2.3.7.2.1 Server Response: Success, App informs user

2.3.7.3 App updates User groups, displays User Groups window

2.3.8 User edits their profile information

2.3.8.1 User edits profile, clicks save changes

2.3.8.2 App sends request to update User info as JSON object to the server

2.3.8.2.1 Server Response: Failure, App informs user why

2.3.8.2.1 Server Response: Success, App informs user

2.3.8.3 App updates User groups with JSON response, displays updated information

**2.4 Administrator Use cases**

2.4.1 Admin edits group

2.4.1.1 Admin selects group in Group window, clicks edit

2.4.1.2 App opens Group Details window

2.4.1.3 Admin edits group data and hits create

2.4.1.4 App sends request to edit group as JSON object to the server, server checks user level

2.4.1.4.1 Server Response: Failure, App informs user

2.4.1.4.2 Server Response: Success, App informs user

2.4.1.5 App refreshes Group Details window with new data (if available)

2.4.2 Admin deletes group

2.4.2.1 Admin selects group in Group window, clicks delete

2.4.2.2 App sends request to edit group as JSON object to the server, server checks user level

2.4.1.2.1 Server Response: Failure, App informs user

2.4.1.2.2 Server Response: Success, App informs user

2.4.2.3 App refreshes Group window with new data (if available)

2.4.2.4 App updates log with new data (if available)

2.4.3 Admin bans/unbans user

2.4.3.1 Admin selects user in Users window, clicks ban/unban

2.4.3.2 App sends request to ban user as JSON object to the server, server checks user level

2.4.3.2.1 Server Response: Failure, App informs user

2.4.3.2.2 Server Response: Success, App informs user

2.4.3.3 App refreshes Group Details window with new data (if available)

2.4.3.4 App updates log with new data (if available)

2.4.4 Admin edits user info

2.4.4.1 Admin selects user in Users window, clicks edit

2.4.4.2 App sends request to edit user as JSON object to the server, server checks user level

2.4.4.2.1 Server Response: Failure, App informs user

2.4.4.2.2 Server Response: Success, App informs user and sends JSON

2.4.4.3 App opens User Details window with new data (if available)

2.4.4.4 App updates log with new data (if available)

2.4.4.5 Admin updates User details, clicks save

2.4.4.5.1 Server Response: Failure, App informs user

2.4.4.5.2 Server Response: Success, App informs user and sends JSON

2.4.4.6 App refreshes User Details window with new data (if available)

2.4.4.7 App updates log with new data (if available)