

---

**Group J**

---

**WhiteBoard**

**Software Requirements Specification for a Active  
Teaming Application**

**Version <1.0>**

---

***Linda Li, Yifeng Jin, Masuda S. Farehia***

Company Name	Version: 1.0
Software Requirements Specification	Date: 03/03/2020
Document identifier	

## Revision History

Date	Version	Description	Author

Company Name	Version: 1.0
Software Requirements Specification	Date: 03/03/2020
Document identifier	

## Table of Contents

1.	Introduction	3
1.1	Purpose	3
1.2	Scope	3
1.3	Definitions, Acronyms, and Abbreviations	4
1.4	References	4
1.5	Overview	4
2.	Overall Descriptions	5
2.1	Use-Case Model Survey	5
2.2	Assumptions and Dependencies	6
3.	Specific Requirements	8
3.1	Use-Case Reports	8
3.2	Supplementary Requirements	10
4.	Supporting Information	11
	Index	11

Company Name	Version: 1.0
Software Requirements Specification	Date: 03/03/2020
Document identifier	

# Software Requirements Specification

## 1. Introduction

Our application is an interface intended for users to create active teams with other members who share similar interests and skill sets in order to develop their professional skills. Users are able to create and join active groups, make meaningful work-related connections, are able to gain a reputation, and will be able to use this platform to showcase their projects.

### 1.1. Purpose

The purpose of this application is for it to be an online facilitator of academic/professional interest and skills. As students in college, we understand that connecting to others plays a big part in cementing our own knowledge and creating a community within our fields. However, with such busy schedules, peoples from all walks of life have trouble making the time and energy to do so. Therefore, creating an application that allows people to connect on their devices, will allow users to do so at a much faster and much easier rate. As such, by forming groups and developing practical projects that can be rated and showcased -- creative minds will be brought together for the purpose of using their talents to make the world a better place.

### 1.2. Scope

The scope of this application will remain within the realm of users with credentials creating groups. In these groups they have to interact with their peers and these interactions can be viewed and rated by other users. The users will be divided into four sub-sections: Visitor, VIP, Ordinary User (OU), and Super Users (SU) -- which is split between a founding super user and a domestic SU. Each group will have their own respective privileges and access range. The privileges and relations between each user are demonstrated in Section 2 in the form of a use case diagram. Overall, visitors have only the power to look at limited information about groups, OU's have the ability to all the basic functions of the application, VIP's are able to evaluate groups, and SU have admin rights over the system.

Company Name	Version: 1.0
Software Requirements Specification	Date: 03/03/2020
Document identifier	

### 1.3. Definitions, Acronyms, and Abbreviations

Terms	Definitions
ATS	Active Teaming System
OU	Ordinary User
FS	Founding Superuser
DS	Domestic Superuser
VIP	Ordinary Users whose reputation scores exceed threshold set by Super User
Visitor	A user that does not require a login and can view projects/profiles

### 1.4. References

- Application modeled after Active Teaming System outlined provided.
- Website used to format UML (figure 1) shown on page 5:

<https://www.lucidchart.com>

### 1.5. Overview

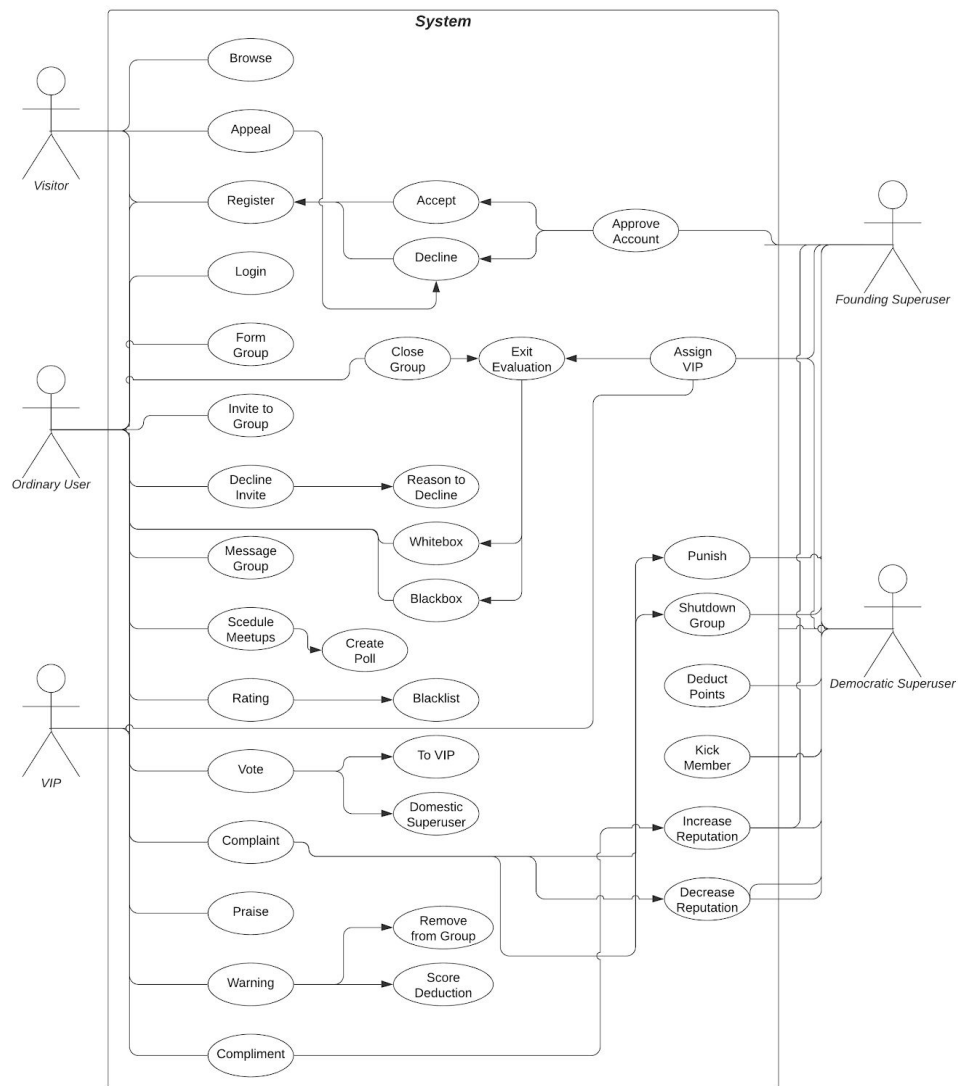
The remainder of this document includes three more sections. Section 2 provides an overview of use-cases by modelling the application with a use-case diagram. The model will display the relations between each of the users. This section also reviews the assumptions and dependencies required to complete the initial prototype of this application. It will further discuss any subsystem or components in which is required by the software. Section 3 goes into further details on the use case diagram from section 2 and supplementary requirements of this application. Section 4 provides supporting information for the document.

Company Name	Version: 1.0
Software Requirements Specification	Date: 03/03/2020
Document identifier	

## 2. Overall Description

This section introduces the general properties of the system, providing a visual of different user permissions/transitions and actions/reactions. System attributes are briefly explained and end with a discussion on assumptions and dependencies.

### 2.1. Use-Case Model Survey



Company Name	Version: 1.0
Software Requirements Specification	Date: 03/03/2020
Document identifier	

The UML representing our system and the purpose of this use-case diagram is to illustrate how each user interacts with the specific system. There are 3 types of regular users in our system: ordinary user, visitor and Vip. In addition, there are 2 types of super-user: founding super-user and domestic super-user.

**Visitors:** The user(s) who are not registered to the system but have the right to browse and complain. Visitors can register to be an OU must be referenced by an OU or VIP and upon being approved by a FS.

**Ordinary User:** The user(s) who already registered in the system and must login. OU(s) can invite other OU(s)/ visitor(s)/ into the group, refer visitors, vote, rate, complain and compliment. OU(s) can be promoted to VIP or demoted based on the reputation scores.

**VIP:** The user(s) who is an OU and the reputation scores exceed a threshold set by SU. VIP(s) are allowed access to all the privileges of an OU(s), and a VIP can evaluate/determine the reputation scores for the group/members. In addition, VIP(s) will be promoted and demoted based on the reputation scores. Based on the vote of VIP(s), one VIP user can be promoted to democratic super user based on VIP user's vote.

**Super-User:** There are only two SUs, one founding SU who initializes the system. Another is democratic SU who is voted by VIPs. SU can handle a complaint from OU(s)/Visitors and is authorized to shut down the team, kick out members(users), and deduct scores.

## 2.2 Assumptions and Dependencies

The following dependencies and assumptions are made:

- The user hierarchy is as follows:
  1. Super-User
    1. Founding Super-User
    2. Democratic Super-User
  2. VIP User
  3. Ordinary User
  4. Visitor User

Company Name	Version: 1.0
Software Requirements Specification	Date: 03/03/2020
Document identifier	

In this system, the user hierarchy is very clear. The top level is Super-User, then VIP user→ Ordinary user → visitor user. Each level is the subset of the next level.

- Assumptions:
- When a user registers, the system needs to verify mobile text messages to avoid a user registering multiple accounts.
- The rating system and the reputation scores will be adopted by average rate/scores to avoid malicious rating.
- Let's assume that if an visitor user really wants to register to be an Ordinary user, but no one refers this user or the application can't be approved for many reasons. What can the user do? The purchased system is going to solve these problems.
- Purchased System:
  - An Visitor who purchased can become an Ordinary User directly without being referenced by OU/VIP.
  - An Ordinary User who purchased can become an VIP User directly(no scores required and the current reputation scores will be 0)
  - Ordinary users/ VIPs can improve the reputation scores by purchasing points
  - Ordinary users/ VIPs can purchase a one-time sale card named "life saving" to avoid being kicked out of the system or be added to the blocklist.
  - New Ordinary Users and VIP Users can't rate/evaluate/determine other groups/members in the 48 hours their account is created.



Company Name	Version: 1.0
Software Requirements Specification	Date: 03/03/2020
Document identifier	

### 3. Specific Requirements

#### 3.1. Use-Case Reports

##### Use-case 1: Create Account

###### **Description**

- Users will be able to make an account.
- They must be able to input their name, email, interest(s), credentials, and reference.
- Those who do not wish to make an account can browse select groups and profile on the main page.

###### **Post-Condition**

- Users information will be saved into the database.

###### **Actors**

- Visitor

##### Use-case 2: Obtain Approval

###### **Description**

- Should the super user approve their account, they will receive an email approval and can create an account/password and be saved to the database.
- Should they be rejected, they can appeal, should it get rejected again, they are added to a blacklist.
- Once the visitor is given an account, their reference can give them an initial score.

###### **Post-Condition**

- A blacklist database will contain all users who cannot create accounts.
- The newly made users will have an account page and be able to access all typical user functions - save for rating other users which will take 48 hours to come into play.

###### **Actors**

- Visitor
- Super User
- OU/VIP

Company Name	Version: 1.0
Software Requirements Specification	Date: 03/03/2020
Document identifier	

### **Use-case 3: Create A Group**

#### **Description**

- Every user can create a group with a purpose, and invite other OU's.
- Should an OU want to reject an invite they must respond with a reason.

#### **Post-Condition**

- A new page dedicated to this group will be created.
- Creator can decide whether the page can be public, protected, or private.
- Each user must be able to access the group they are part of from their accounts page.

#### **Actors**

- OU

### **Use-case 4: Activity in a Group**

#### **Description**

- Moderation, posting, polling for meetups can happen in a group page.
- If a member has missed meetings twice, they will be issued a warning.
- Members can vote to give warning, praise, or kick a member out -- it must be unanimous and these will affect the person's reputation.
- Warnings / voted out members of the group can appeal to the SU for changes.

#### **Post-Condition**

- All this information must be saved into a database in order to be accessible by all the users.
- SuperUsers have access to all groups whether or no they are apart of them or not.

#### **Actors**

- OU
- Super User

### **Use-case 5: Deleting a Group**

#### **Description**

- Unanimous vote to close the group prevails.
- Each member will conduct an exit evaluation for all the other members in the group.
- SU to assign a VIP to evaluate the group and determine the groups reputation.

#### **Post-Condition**

Company Name	Version: 1.0
Software Requirements Specification	Date: 03/03/2020
Document identifier	

- Group and all its related information will be deleted from the database in 48 hours.
- Users will no longer have access to this page.

#### **Actors**

- OU
- Super User
- VIP

### **3.2 Supplementary Requirements**

- An OU can put some OUs to his/her white-box or blackbox.
- An OU with negative reputation score will be removed from the system and put into black-list automatically.
- Visitors and OUs can complain to SU about a group or other OUs, the SU will decide if the complaint merits action.
- The entire system keep a list of taboo word list that will be converted to \*\*\* and the OU's reputation score will be decreased by 1 or 5 depending on frequency.
- OUs can send compliment about other OUs to SU.

Company Name	Version: 1.0
Software Requirements Specification	Date: 03/03/2020
Document identifier	

#### 4. Supporting Information

#### Index

Visitor User (VU)	
.....	
Ordinary User (RU) .....	
VIP User (VIPU) .....	5, 6, 8, 9
Founding Superuser (FS) .....	
Democratic Superuser (DS).....	
Unified Modelling Language (UML) .....	
Use-case diagram .....	
Use-cases Report.....	
Supplementary Requirements.....	