Team Q

<Csc 32200 – E1 Project>
Active Teaming System – Phase II

Version <1.1>

Revision History

Date	Version	Description	Author
03/06/20	1.0	Phase I – system requirements specification.	Albert, Gabriel, Orion, Ide
4/29/20	1.1	Phase II – Design report & updated Use-Case Diagram from Phase I	Albert, Gabriel, Orion, Ide

Table of Contents

1. Introduction

1.1 Purpose

There are four types of users for this program; visitors, ordinary users (OU), VIPs, and the Super User (SU). All users will be ranked based on their reputation scores given by each other except the visitors who have no rep score but can still communicate with SU. **However**, visitors are allowed to contact the SU directly regarding problems with certain users/ user groups. Upon reaching the 30-point reputation threshold, ordinary users can be promoted to VIP status. There is only **one** Super User at a time and he/she is voted into power directly by the VIPs in a democratic election. OUs are able to establish groups and create dedicated group pages that they can make public or private to other groups/users/visitors. Profanity from any user/group will be censored and penalized with a 1-point reputation deduction upon first encounter. However, ever subsequent slur would decrease reputation score by 5. Regarding new users (currently still visitors), they must be referred in by other users or during the application during the application process. Once the application is approved by the SU, the referrer provides an initial reputation score to the new ordinary user. If the referrer is an ordinary user, they can assign the initial rep between 0-10 and if they're VIP they can assign between a 0-20.

The goal of this software specification sheet is to outline and describe the behavior of users at different levels of privileges such as interactions amongst each other. There are four levels of privileges ranging from most to least respectively: Super-User (SU) → VIP → Ordinary User (OU) → Visitor. VIP and OU can bring in visitors as references to become OU but the SU are the ones to make the final decision whether a visitor's sign-up application is approved of or not. Starting from an OU, users would be rated based on a reputation score which could raise your status to a VIP once it reaches a certain amount but could also drop a VIP's status back to OU if it falls too low. As for the SU, one is elected by the votes of the VIPs. On the main page of the program, top 3 rated projects are displayed right beside the top rated OU and VIP. This program allows OU and VIP users to form groups and collaborate projects within the program. Once a group is established among a few individuals, a dedicated workspace page will be generated for the group where they could share ideas privately amongst each other or publicly with the general populated OU/visitors. Projects on display would also have a reputation rating system.

There should be a slightly different UI for every user type on this program:

- **The Super-User** should include a panel filled with new applications to review incoming potential members where they will be able to review and accept or deny application. The application should have a link to communicate with the VIP or OU that the visitor is using as a reference for the application to verify the new user's application with the referrer. Upon acceptance, an email should be sent to the applicant based on email

provided in the application with an account ID as well as a temporary password where the user would be requested to change ASAP upon signing in. Another panel should include all the applications that have been appealed due to initial rejections as well as the appeals of members of groups who have been voted out and are requesting to bring up their reputation score. There should also be another panel containing messages from OUs and visitors addressing issues on certain groups or other OUs that are up for evaluation. A blacklist should exist underneath the panel to show who has been blacklisted. This Blacklist should have the functionality to block appealed application and users upon the second rejection as well as whitelist individuals to reverse the process.

- **The VIP** is not that different from the OU in terms of place in the system. There are only a few differences such as requirement of a VIP to have a reputation score of 25 or higher to show authority amongst the OUs. In order for an OU to be promoted to a VIP is by obtaining 30 or higher reputation points. The VIPs also have the ability to democratically vote a single VIP up to the SU status. Unlike an OU, VIP could give the new users who referenced then an initial reputation score of between 0 - 20 instead of the common 0 - 10.

- **An Ordinary-User** should have a basic GUI where they could interact with other OU and request, accept, deny groups with the other OU. On an OU's page, there should be a list of groups the OU is participating in for easy access. If the OU rejects joining a group, there should be a prompt asking them why upon rejection. There would also be a couple of check-boxes. The white check-box refers to accepting all invites while a black-box allows the OU to reject all invites. Upon the creation of a group, a page should be established where the group is able to share information amongst each other such as posting updates or meetup schedules. Information posted could be made public or kept private. When scheduling meetups between group members, there will be a voting poll to establish the best time for everyone based on popular vote. Upon two missed meetings, a warning popup will appear saying so and voted out members could also appeal to the SU for updated reputation points. As for the popup warning/praise, any group member could issue it but the actual message won't be issued unless every group member other than the receiver has unanimously agreed upon it. Upon a third warning issue, the targeted member would be kicked and reduced by five reputation points. Every member of the group would have equal rights to modify settings and changes within the group. Members have the ability to close up a group which would start up an exit evaluation on all the members within the group for everyone. The group members are able to give each other reputation points and the median would be the final reputation points each member receives. Group members are also able to categorize each other into white/black boxes for future inclusions of other groups. If the black box has been selected, the group member would be prompted to specify why. Members could also be voted out. If the votes pass

and a member is kicked, they get a reduction of 10 reputation points. If the user is within the negative range of reputation points, they immediately get removed from the system and placed into the blacklist. There should be a separate panel for all the new users who used the OU as a reference. If their applications have been approved, then the OU would be able to set their reputation score ranging between a 0 - 10.

- **Both visitors and OU** can complain up to the Super-user through messages about certain groups or specific OUs that could prompt the SU to take action such as shutting down the group, lowering reputation score or kicking the users out of the system entirely. Once an OU has been kicked out of the system, they are allowed one last time to collect their data before it is terminated for good.
- There will be a library or taboo words that would be censored out if detected between users across the system. Continuous use of taboo words would result in automatic reduction of reputation points. Compliments between OU would be notified to the SU who can then raise the reputation score of the targeted OU.

1.2 Scope

The *Active Teaming System* that is to be developed will facilitate the active teaming of people with similar interest and skill-set to forge groups for a certain project. The Active Teaming System requires the following features.

- Provides a GUI so different users and visitors can interact with it.
- Support the following users: OU, VIP, SU, and visitors.
- Provides a login facility to all registered users, and an option for visitors to register to be OU.
- Provide the SU with the capability to approve or reject a registration application.
- A rating system for OU, VIP, and SU.
- The system's GUI provides a way to showcase the top 3 rated projects and rating of each profile.
- Allows users to form groups, which in effect will generate a web-page for that specific group and will be available and accessible to all group members. In addition, privacy settings should be provided.
- A polling system to facilitate the schedule of meetups, any group member can request a meetup.
- Group users can vote to close the group.
- Provide a communication system, so visitors can communicate with the SU.

- Users with a negative rating will be removed from the group.
- Give removed users one-time log in and then block their access to the system for ever.
- Block taboo words, if taboo words are used again decrease reputation score of such user.
- Allow OUs to compliment other OUs, and SU can increase the reputation score of the complemented OU.

The features described above meet the needs of the requested system. These features are to be included in the software development cycle.

1.3 Definitions, Acronyms, and Abbreviations

Ordinary User (OU) - A user that is self-registered but approved by the super-user. The OU needs to login to perform its activities. The OU can give a score of 0-10 to referred applicants and could form groups as well as invite other OUs to join the group for some specific purposes.

Very Important Person (VIP) - One step higher in privileges compared to the OU. Must maintain a reputation score of 25 or higher to remain in this position. Has the ability to elect a SU or potential to be elected themselves into SU position.

Super User (SU) - User with the highest privilege in the system. The super user level is composed of one user democratically voted by VIPs.

Visitors - Regular users that don't have any other permission to participate or make changes to the system and is basically only allowed to surf around.

Graphical User Interface (GUI) - Graphical visual component that allow users to interact with the other parts of the system. It displays forms, menus, colors and all other graphics to facilitate navigation.

User Interface (UI) - A frontend visual design of the software that serves as a bridge connecting the user to the backend parts of the software.

Identification (ID) - A process of distinguishing who/what is allowed to access and manipulate what component of the system. For the present system, one identification is the credential to use for the login. Another is the position (VIP, OU, SU, Visitor) of each user of the system. It helps manage and secure the system from unwanted activities.

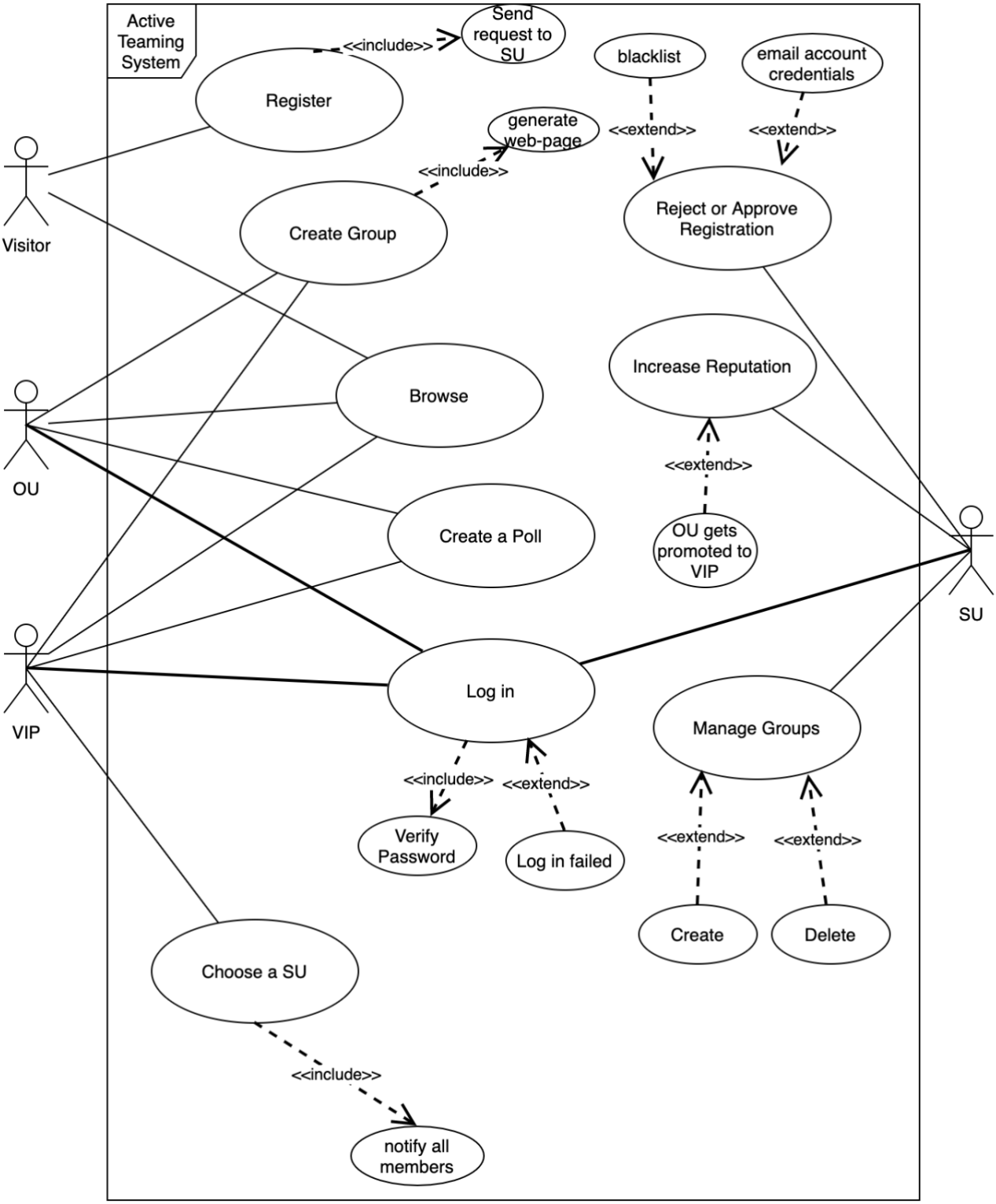
1.4 References

1.5 Overview

From this point onward in the software requirement specification, we will be addressing the structural design and model used to construct the program and its functionality as well as how each part of the program interacts with one another. A few examples would be how the Super-User has the privileges to override and decide on all decisions across every other user and make all the final decisions. Depending on the SU's choices, it may or may not affect changes across the system. The documentation will list out all the unspecified rights of the users even if it is not in the specifications based on logical assumptions of what they are allowed to do. Diagrams would be used to show the relationships between each subsection of the project for the use-case model and a follow up report explaining each portion in detail.

2. Overall Description

2.1 Use-Case Model Survey



2.2 Assumptions and Dependencies

- The system will only work on the following OS: Windows, Linux, Mac
- There can only be a maximum of two SU, one that is voted upon and one that created the system
- Since VIPs can vote on a democratic SU, it is possible for democratic SU to revert to VIP or OU
- Pending on input, SU controls the reputation score of all other users
    - SU should however, not have a reason to raise or lower reputation score of others without reason (Must provide valid reasoning when changing another user's reputation score)
- Assuming the SU was originally selected from a group of VIPs, the reputation score of a SU cannot fall below that of VIP score (below 25)
    - We assume that the score of a SU can still be affected based on the feedback of other OU and VIPs.
    - We also assume given these standards, the SU CANNOT alter their own reputation score.
    - Upon falling below 25 reputation points, the SU automatically demote to OU and the VIP will be notified to elect a new SU.
- Visitors are only able to register as an OU once
    - Since visitor's application can only be accepted or rejected once, there should be a feature in place restricting visitors for trying to register again
    - If a visitor's application is rejected once, they should have a form of direct messaging in order to appeal to the SU
    - If wholly rejected, visitors should receive a message stating that their application was denied and they are no longer allowed to register
- Top 3 rated projects and top rated OU must appear on home page
    - Since this is the case, any alteration in ratings should result in a change if necessary, on the front page.
    - In the case that there are tying scores, the OU and projects that have aggregated the least number of negative scores should be displayed

- Since group members can poll in order to establish a meet-up time, there should exist a polling feature in which a member can create a poll, enter a description, and enter their choice of times. A timer should also be included in order to allow other users enough time to vote
    - Other members of the group should then receive a notification about the poll and they have to option to either vote or neglect it
        - If neglected once, the user will receive no message but the system must record that the user did not vote (a missed-poll counter)
        - If neglected twice, the user should receive a prepared warning statement telling them that they are at risk of being removed. System should record that they missed two polls. Users should also have some sort of reminder whenever they are asked to poll if they have two missed poll votes.
        - If a user neglected a poll 3 times, they should be removed due to inactivity within a group.
        - Group leader should be able to assign tasks to other group members. These tasks should appear to every member whenever they access the project.
        - Users should be able to confirm if they have completed their task, which should notify other members of the project.
        - If a user does not finish their tasks for a project, they should receive a warning. Two warnings result in a similar warning statement as mentioned previously. Three warnings should lead to a removal and appeal system as previously mentioned
    - Group members are allowed to praise or reprimand another group member
        - In order to allow the appraisal or reprimand, all other members of the group must vote to praise or reprimand. If one member has a differing vote, then the vote should be nullified after allowing all members to vote. A message should be displayed to all members regarding the outcome
        - If an individual gets 3 warnings from the group, that automatically results in a 5-point deduction and other members should be prompted with a poll

to kick said member. If there is a unanimous vote to kick, the member loses 10 points and is removed.

- An appeal system should also exist if the removed members feels as if the removal was unjust.
- A system must be put in place to also prevent the abuse of warnings and member removal.

- Users who were removed but were once allowed on the system are allowed to access the system one final time. The period of access should be provided by the SU, spanning a certain length of time. Once the user logs of in that time frame or if time expires, the user will be blacklisted
- A regex system will be put in place in order to find taboo words. If found, it will be replaced by "***", and the reputation score should decrease by 1, by 5 if they repeat the same word or use another taboo word
- Democratic SU should have a reputation score available to be seen by everyone. Founding SU should have a unique rep score indicating they are the founder.
- There should exist an active channel between users and SU, in order to allow compliments and complaints and any sort of query
- There should also be a system in place to check and see if there is an abuse of the praise and reprimand system. If detected or reported, SU can decide the fate of the users abusing the system.

3. Specific Requirements

3.1 Use-Case Reports

1. **SU -** is responsible for the approval or rejection of applications. Also, the SU is responsible for managing all groups in the system.
    - **Use case: Reject or Approve registration -** The SU reviews the applications of visitors who want to register in the system. He can reject or approve such applications.
    - **Use case: Manage Groups -** The SU manages all groups in the system. He can delete or create new groups.
    - **Use case: Increase Reputation** - The SU can increase the reputation score of OUs, who have been complimented by other OUs.

    Name of Use Case: Reject or Approve registration
    Description - A visitor's application will be rejected or approved.
    Precondition - The SU should be logged in the system.
    Normal Flow of Events -
    1. A panel with new applications is present for the SU.
    2. SU reviews the incoming applications of potential members.
    3. Decides whether he accepts or rejects the application.
    4. An email with relevant information will be sent to the applicant.

    Name of Use Case: Manage Groups
    Description - Creates or deletes any group in the system.
    Precondition - The SU should be logged in the system.
    Normal Flow of Events -
    1. A panel with the complete list of all groups in the system is present for the SU.
    2. A button will be present to create a new group
       OR
    3. SU can select any group in the list.
    4. A menu will open with the group's info, he is given the option to delete the group.
    5. An email with relevant information will be sent to all the members in the deleted group.

    Name of Use Case: Increase Reputation
    Description - Increases reputation score of complemented OU
    Precondition - The SU should be logged in the system.
    Normal Flow of Events -
    1. OUs send compliments to other OU

2. When OU receives 3 compliments, the SU will receive an email with relevant information.
3. The SU will increase the reputation score of the complemented OU.

2. OU - can create groups to collaborate on a certain project, OU has the option to vote on certain issues.

- **Use case - Create Group -** OUs can form groups by inviting other OUs for certain projects.
- **Use case - Create a poll-** OUs have the option of creating polls to vote on different issues in the group.

Name of Use Case: Create Group
Description - Any OU can form groups by inviting other OUs in the system
Precondition - The OU should be logged in the system.
Normal Flow of Events -

1. A button for creating groups will be present in the GUI of the OU
2. A form will open.
3. OU fills in basic information, such as group name, description, and other OUs to invite.
4. The invitation will be sent to the inbox of invited OUs.
5. OUs review the invitation. They can decline or accept the invitation. In addition, they can blacklist or whitelist the OU to accept or decline future invites.
6. If rejected OUs reply with a reason.
7. If accepted, a web-page of the group will be automatically generated.
8. Once the group is formed the web-page is available for all group members, and specific information can be public or set to private for visitors.

Name of Use Case: Create a poll
Description - OUs can vote to issue a warning or a praise to a group member. OUs can vote to schedule meetings, OUs can vote to remove other group members. OUs can vote to close the group, and conduct an exit evaluation to other members.
Precondition - The OU should be logged in the system and should be part of a group.
Normal Flow of Events -

1. A “Create poll” button is available for all group members.
2. A menu will open with the following options; “praise a member”, “warn a member”, “schedule a meeting”, “remove a member”, “close group”, “evaluate a member”.
3. After selecting any of the presented options, a message will be sent to all group members.

4. If the group member receives three warnings, then he will be automatically removed from the group, and get a 5-point reputation score deduction.
5. If a group member is removed from the group, he or she will receive a 10-point reputation deduction. (members with negative rep score will be automatically remove from the group and be blacklisted)
6. If a meeting has been scheduled, the time slots with the most votes will be chosen. (If a member has missed a scheduled meeting twice, he or she will receive a warning.)
7. After every pool a message will be sent to all group members with relevant information regarding the results of the conducted poll.

3. VIP - are simply OUs whose reputation score is higher than 30.

- **Use case - Choose a SU -** VIPs can vote another VIP as the democratic SU.

Name of Use Case: Choose a SU
Description - All VIPs can vote to select a SU.
Precondition - Should be logged in the system and reputation score should be higher than 30.
Normal Flow of Events -

1. A menu is available for VIPs to conduct a poll to select a SU.
2. All VIPs will be notified of the poll, and they can vote who they want.
3. After the polling has ended, a message will be sent to **all** users in the system.

4. Visitor - can surf around, and contact the SU.

- **Use case: Register -** Visitors have the option to register in the system.

Name of Use Case: Register
Description - A form will be available for visitors to register in the system.
Precondition - Potential members should have a reference by other OUs or VIPs who are already in the system.
Normal Flow of Events -

1. A registration form will open.
2. Visitors fill in basic personal information, such as name, email address, interest, credential and reference.
3. When the applicant submits the form, the application will appear at the SU "Applications Panel"
4. The SU rejects or accepts the application.
5. If approved, the SU will send an email with account id and password and will receive an initial score by the reference. (OU can give a score 0-10, a VIP can give score 0-20)
6. If rejected the applicant can make a appeal to the SU

7. If the SU rejects the applicant appeal then he is blacklisted forever.

General Use Case:

- **Use case: Log in - registered members can log in into the system.**

Name of Use Case: Log in
Description - option for OUs, VIPs, and SUs to log in into the system.
Precondition - Should be registered in the system.
Normal Flow of Events -

1. User inputs his email (or username) and password.
2. Input information will be verified.
3. If incorrect he will be asked to double check his credentials.
4. If correct he will be taken to his specific main page (different "menu" for different types of members).

3.2 Supplementary Requirements

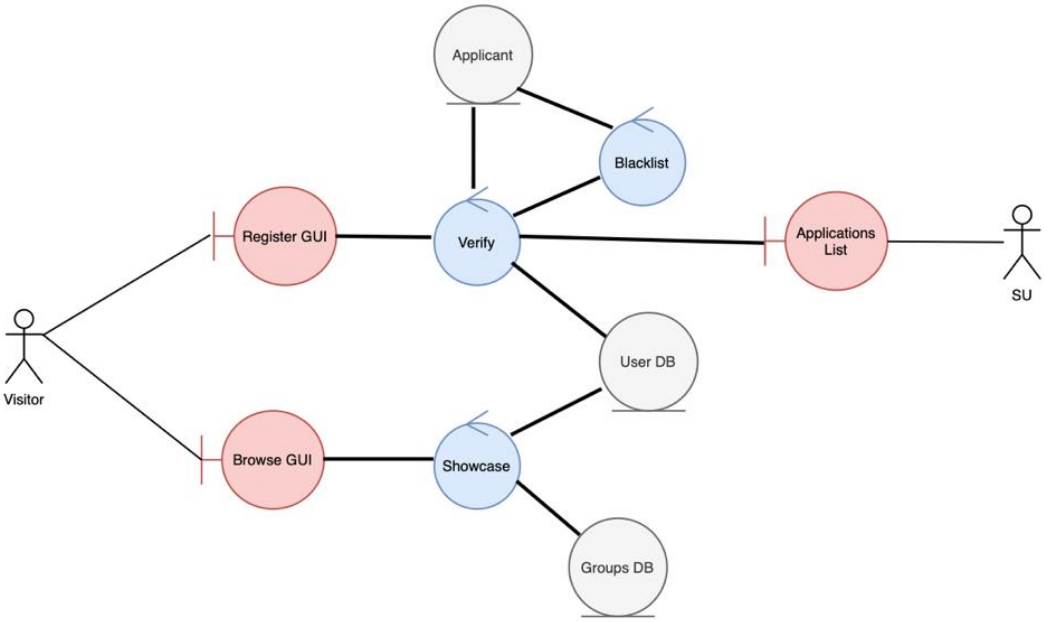
- SU has privileges to override most decisions
- VIP have the abilities to vote someone up to SU or down from SU
- The few differences between VIP and OU is that VIP have a reputation score of 30 or higher when promoting from OU to VIP and if they fall below 25, they demote back to OU
- OU can form groups amongst each other which would generate a separate page for collaboration work on projects as well as a place to update each other on posts and schedule meetings
- These groups can determine what's made public or private within the group
- Both Ordinary-Users and VIPs have the ability to initialize referred members with initial reputation points.
    - The difference is that Ordinary-Users can only give between 0 - 10 and VIP can give between 0 - 20
- SU has panels to review new applications, review current complaints from users and have the ability to blacklist people or modify reputation points for individual users
- OU can collaborate in groups and have the ability to communicate warnings with each other if everyone votes in unison excluding the receiver.
- Members who are kicked or dropped from groups would receive penalties to their reputation score but at the same time also increase depending on the compliments received by others
- Upon receiving 3 warnings from group members, the receiver would be kicked with a point reduction of 5
- If a member is voted out, they get a point reduction of 10

- Any OU who ends up with a negative reputation score would immediately be placed on the blacklist
- These soon to be blacklisted users have one last opportunity to log on and clear out their data
- Visitors would have a button on their GUI to sign up under a referral from an already existing OU or VIP
- Top rated projects and users will be listed out on the front page of the program
- There will be a library of taboo words that will be censored when automatically detected from OU or VIPs.
    - Any use of the words would result in a 1-point penalty. Any further use of the same word results in a 5-point penalty thereafter.

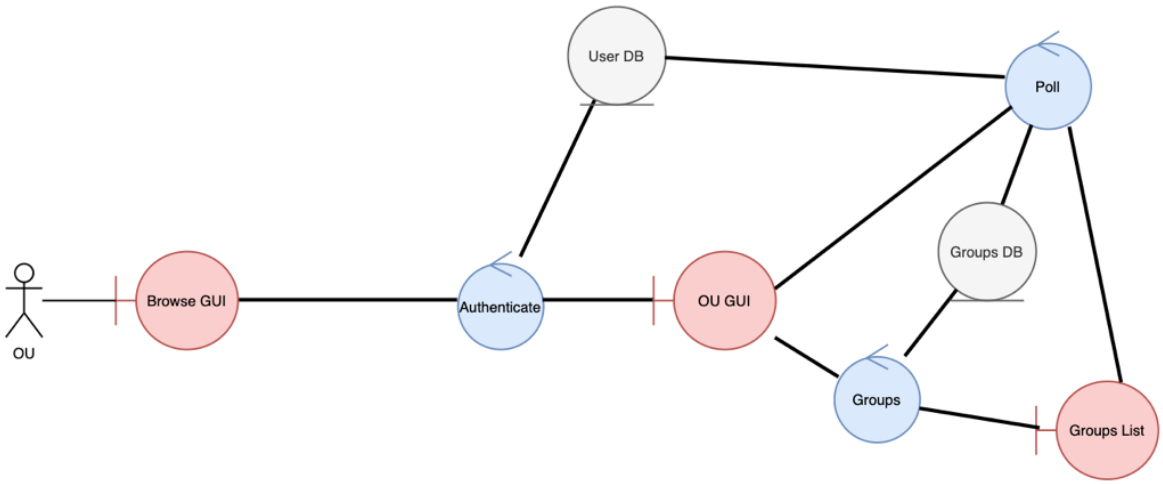
4. Supporting Information

5. Collaboration Class Diagram (Overall picture of system)

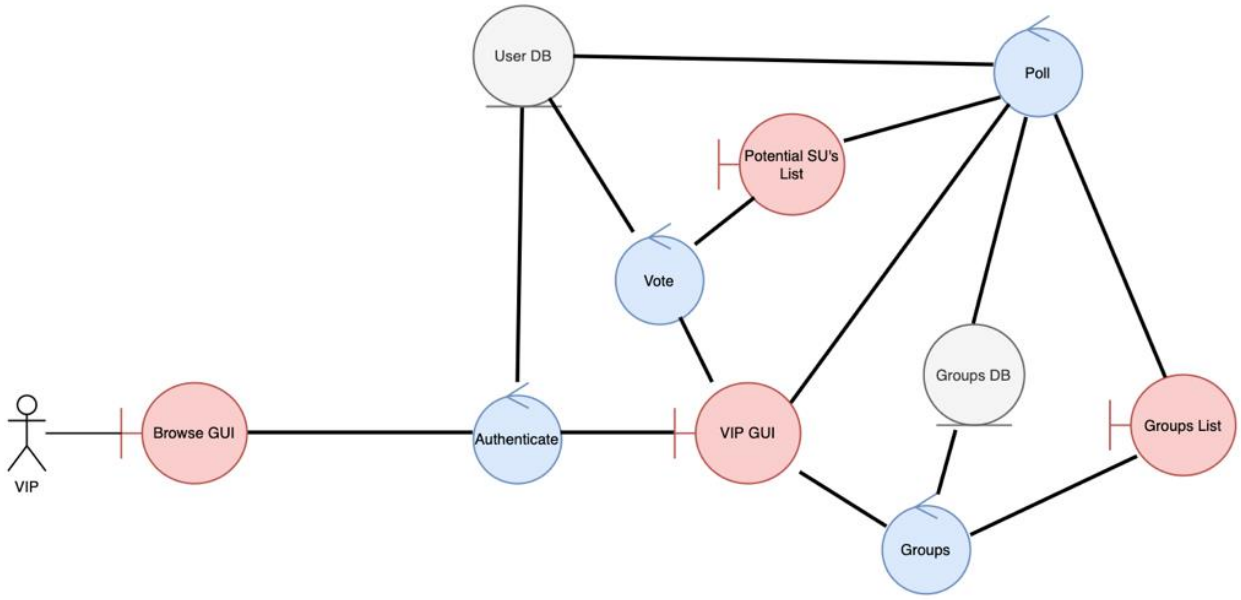
Visitor



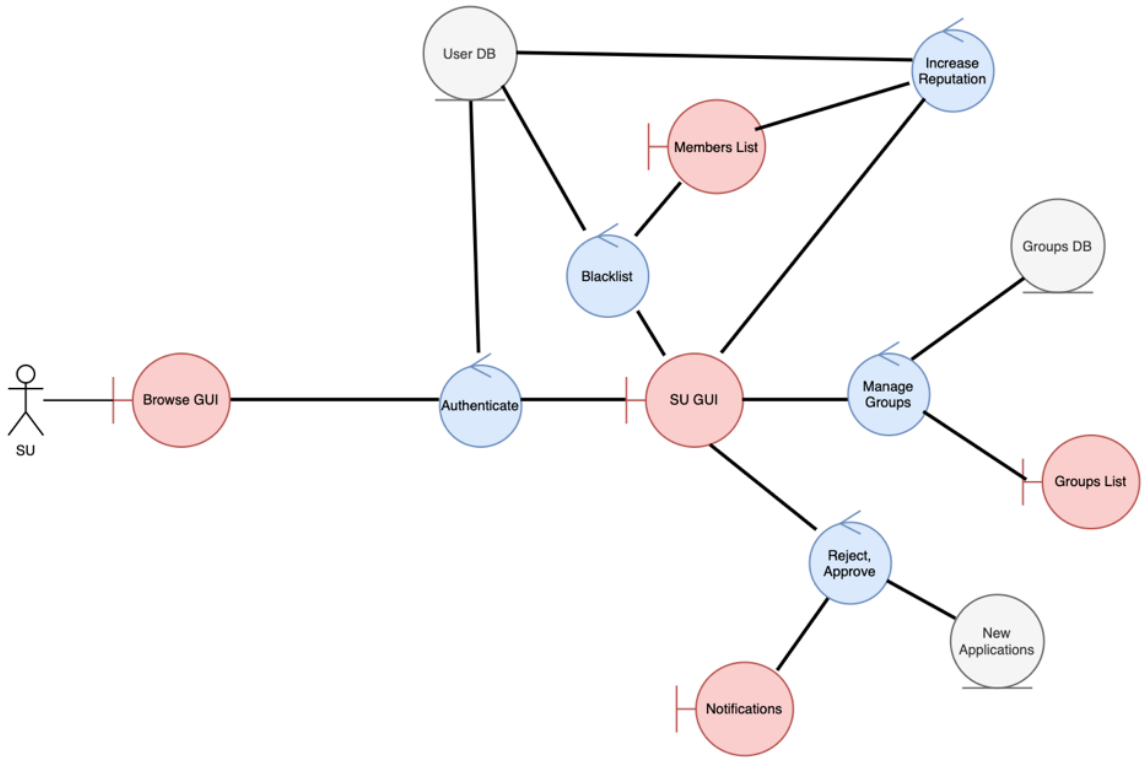
Ordinary User



VIP User



Super User



6. Use Case Analysis

This section provides a more detailed overview of each of the main use-cases illustrated in our specification report. Each use-case is accompanied with a normal and exceptional scenario. In addition, a collaboration class diagram is provided for each use-case. Finally, petri-net is provided for four use-cases.

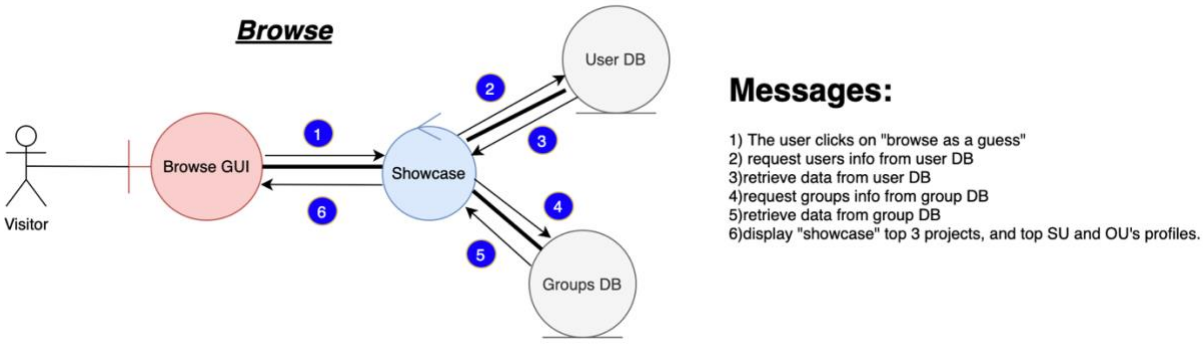
Use-Cases:

1. Browse
2. Register
    - Reject or Approve Registration.
3. Create Group
4. Create Poll
5. Log in
6. Choose a SU (Vote)
7. Manage Groups
8. Increase Reputation

6.1 Browse

Normal Scenario: The user is presented with an option to log in or browse the system as a guess. When the visitor clicks browse, the system will query the user database and groups database to retrieve the top 3 projects and the top featured OU's and SU's. When the records are successfully retrieved the system will display (showcase) those records, so the visitor can browse around.

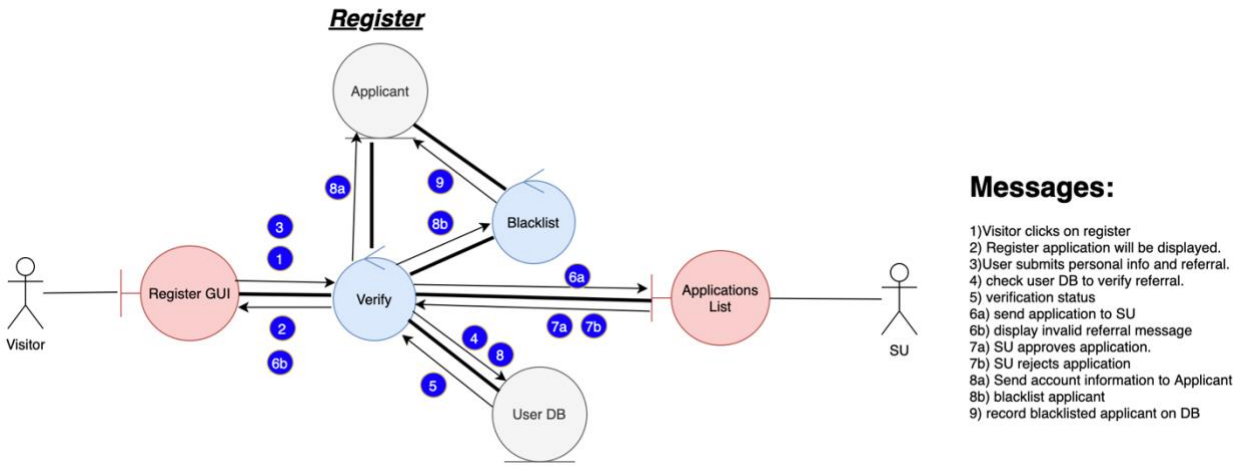
Exceptional Scenario: If the user is already registered in the system, he will have the log in into the system.

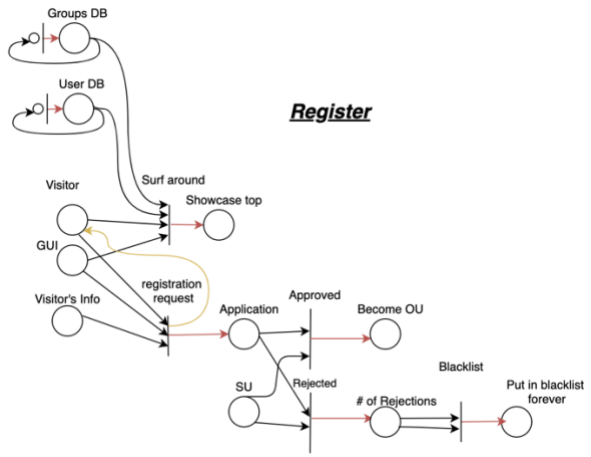


6.2 Register includes Reject or Approve Registration

Normal Scenario: When the user opens the app, he is given the option to browse as a visitor or register. If he decides to register, he will be presented with an application that has to be fill out with personal information. In addition, the user should have a referral from a user who is in the system already. Next, when the application is submitted, it will be sent to a super user, which will review the application and decide if the applicant is approved or rejected. Finally, if the applicant is approved by the SU he will become an OU, but first he will receive an email with account information and password, which he is required to change.

Exceptional Scenario: If the applicant is rejected then he can make an appeal, and the SU will make a final decision. If the application is rejected again then he will be blacklisted forever.

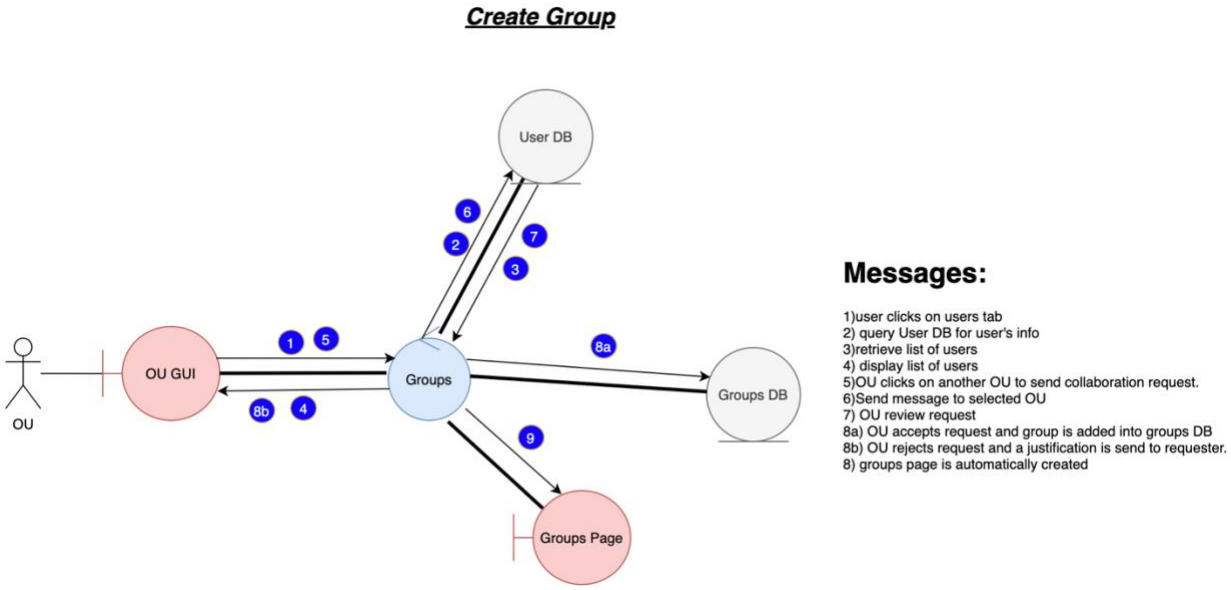


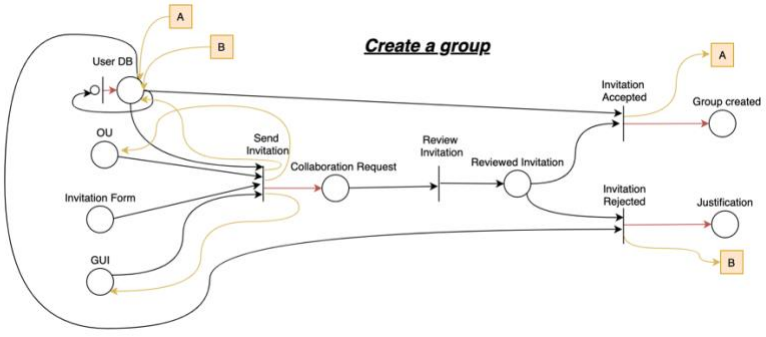


6.3 Create Group

Normal Scenario: OUs have the option to create groups. First, they have to send a collaboration request to another OUs. Second, the invited OU will review the request and if he accepts the invitation the group will be created (including a groups page).

Exceptional Scenario: If the invited OU rejects the invitation then he will have to provide a justification.



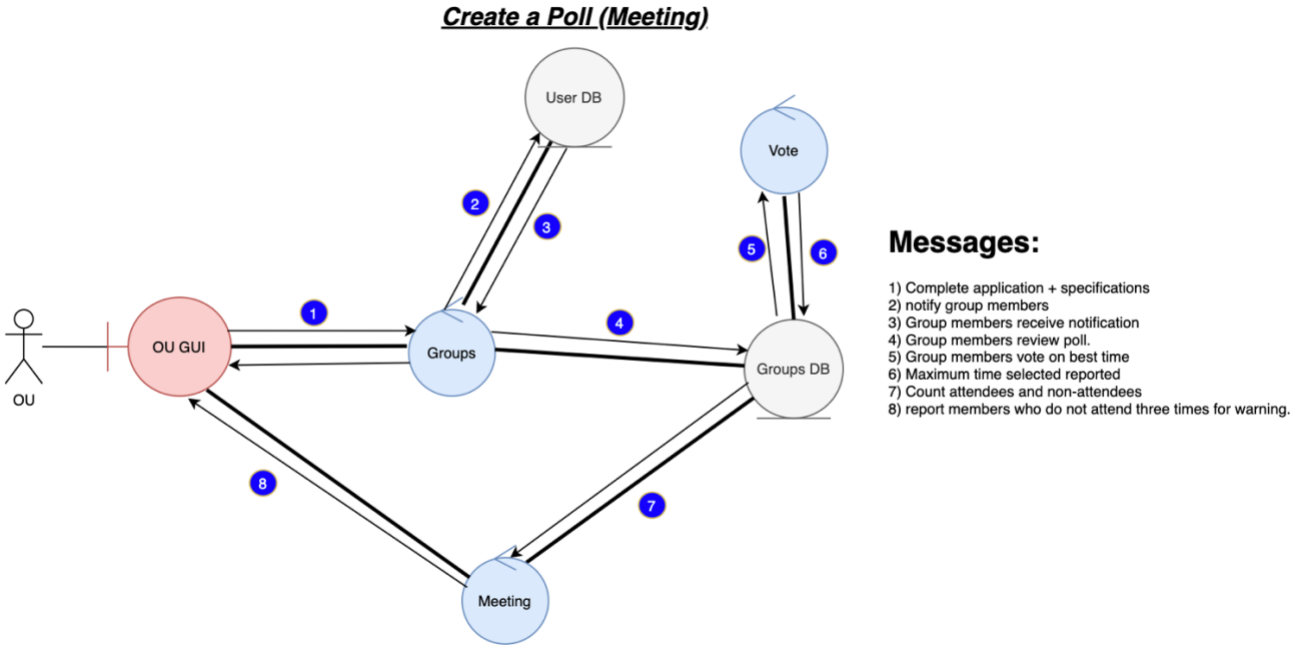


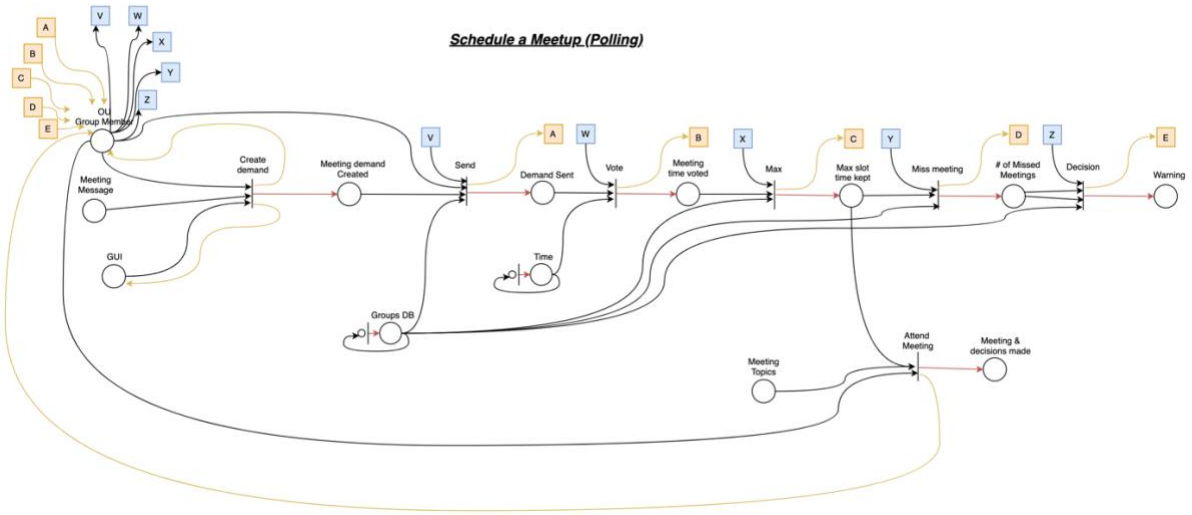
6.4 Create Poll (meeting)

Normal Scenario: Any group member can ask for meet-up polling to find common time for all members to meet. To schedule the meet-up the OU will have to create a poll. The OU will select the date and enter different time slots for the meet-up. Next, all group

members will be notified of the request. Once, all group members respond, the time slot with the most votes will be chosen.

Exceptional Scenario: If a group member does not attend to the meetings three times, he will receive a warning.



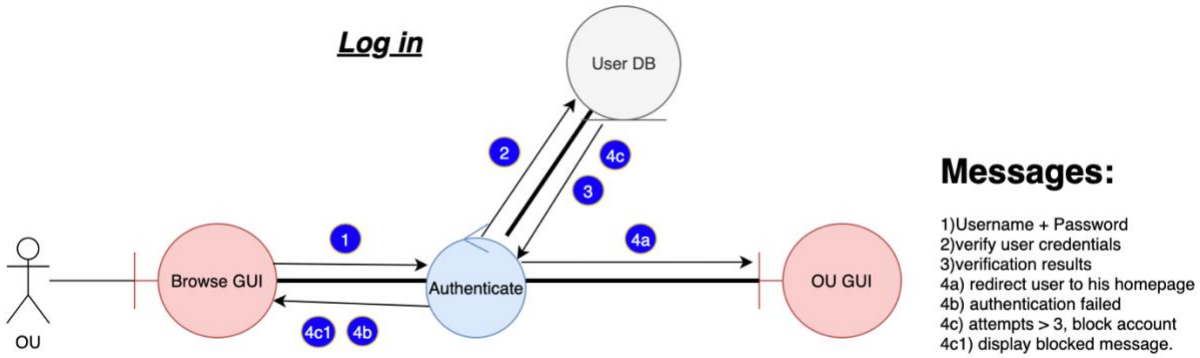


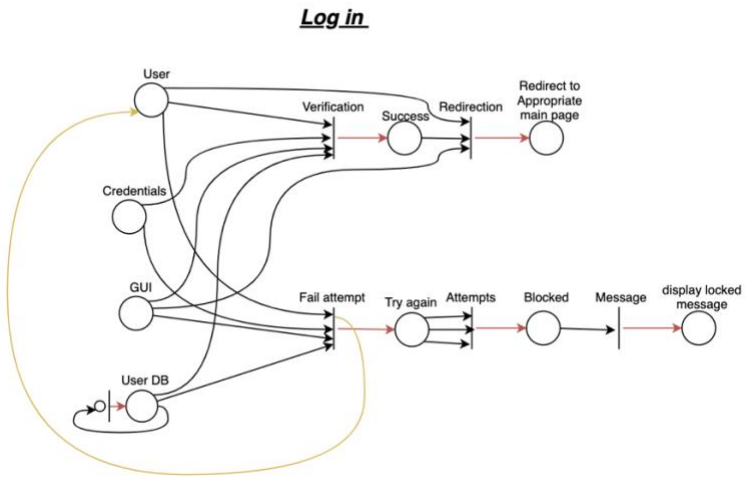
6.5 Log in

Normal Scenario: Registered users will have a log in portal. In order to log in into the system they need to enter their username and password. Next, the system will query the

user database and verify information. If the account exists and the information is correct, then the user will be redirected to their main homepage.

Exceptional Scenario: If the authentication was unsuccessful then the user will be given three more chances, but if he fails to authenticate then the system will block the user and display a blocked message.

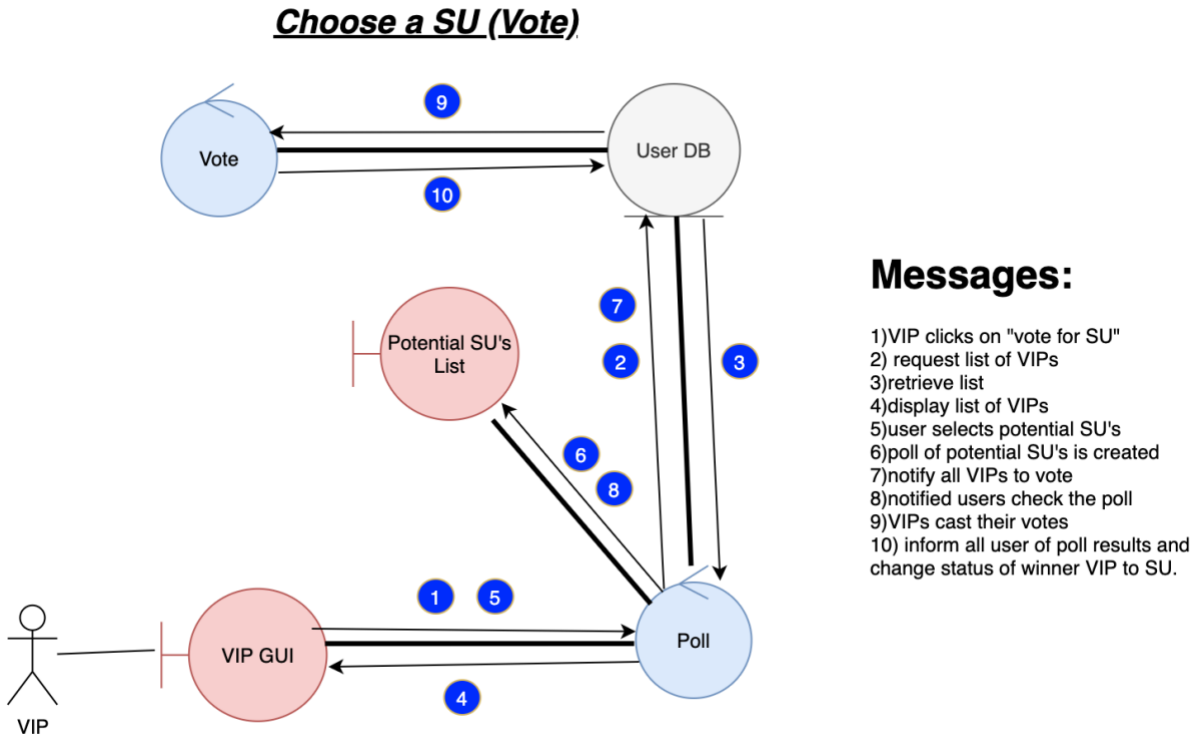




6.6 Choose a SU (Vote)

Normal Scenario: In order to choose a SU, VIPs will need to create a poll. After all potential SU candidates are selected the system will notify all VIP users in the system that the poll is active. Next, at the end of the poll after all VIPs vote, a notification will be sent to all users of the system informing them about the new SU.

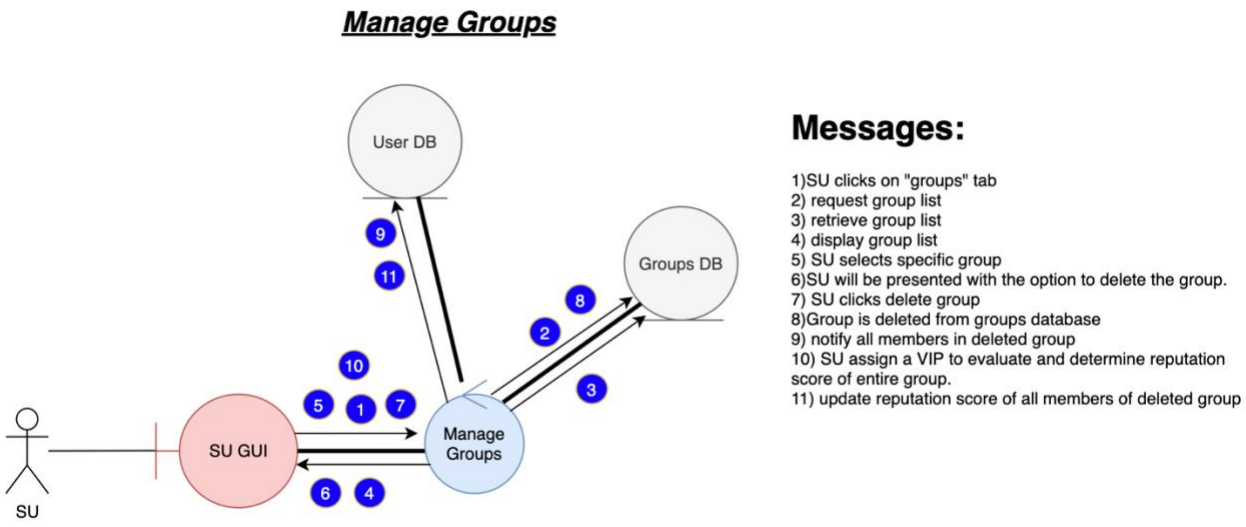
Exceptional Scenario: If the reputation score of a user is not higher than 30, he is not allowed to vote. In addition, the voting VIP should be logged in the system.



6.7 Manage Groups

Normal Scenario: If the SU receives a complain about a group. He has the option to close the group. The SU will have access to all the groups in the system. When the SU clicks the groups tab, the system will query the groups database. Next, the system will display the list of all groups. The SU can select any group and decide to shut down the group completely.

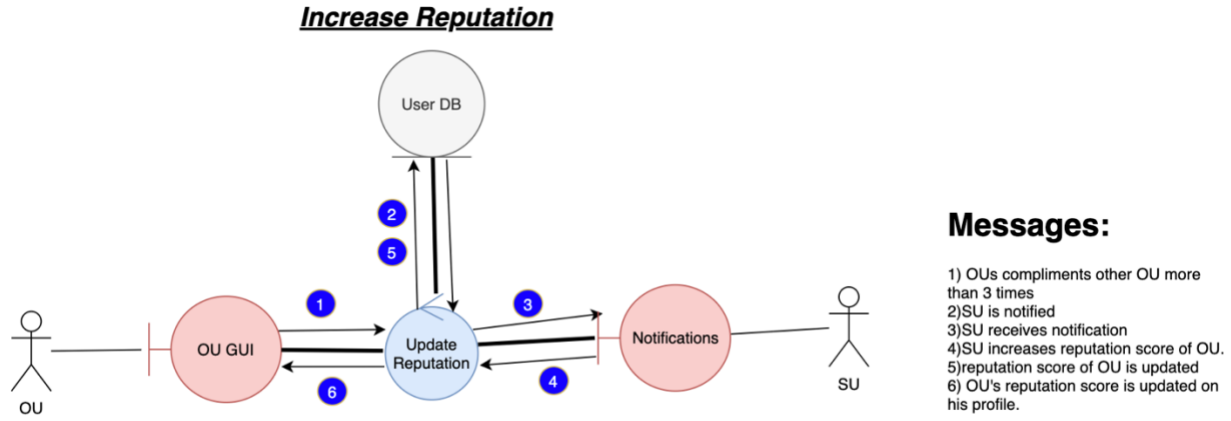
Exceptional Scenario: After group is shut down, the SU will assign a VIP to evaluate the group and determine a reputation score for the entire group to be added/deducted for all members involved.



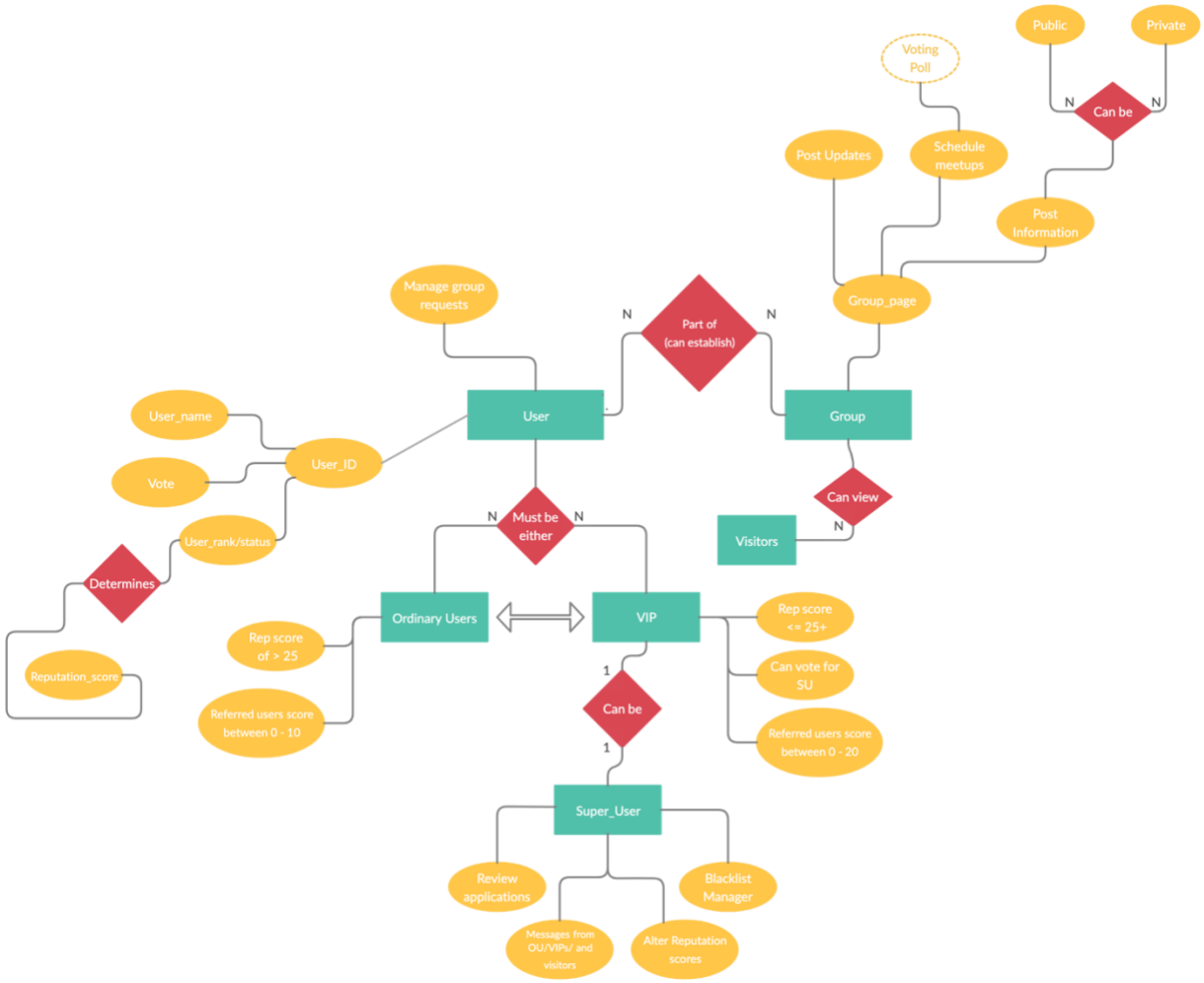
6.8 Increase Reputation

Normal Scenario: When OU's compliment another OU three times, the SU will received a notification. The SU will access the notification panel where he can review all incoming notifications. Next, the user will select the complimented OU and increase his reputation score.

Exceptional Scenario: If OU's use taboo words their reputation score will be decreased by 1, and if they use the same word again his reputation will be decreased by 5. If the user's score falls below zero, he will be removed from the system and put into a blacklist automatically.



7. E-R diagram


Public
Private
Voting Poll
N
Can be
N
Post Updates
Schedule meetups
Post Information
Manage group requests
N
Part of (can establish)
N
Group_page
User_name
User
Group
User_ID
Vote
Can view
N
Must be either
N
User_rank/status
Visitors
Determines
Ordinary Users
VIP
Rep score of > 25
Rep score <= 25+
1
Can vote for SU
Reputation_score
Referred users score between 0 - 10
Can be
Referred users score between 0 - 20
1
Super_User
Review applications
Blacklist Manager
Messages from OU/VIPs/ and visitors
Alter Reputation scores

8. Detailed Design:

Login GUI

```
label usrnm = new label("Username: "); //Create Username label
getcontent().add(usrnm);

label pwrd = new label("Password: "); //Create Password label
getcontent().add(pwrd);

username = new Textfield(); // Retrieve user input username
getcontent().add(username);

pass = new Textfield(); // Retrieve user input password
getcontent().add(pass);

Button login = new Button(login); // Have login button
login.listen(new listen); // Inputs user click on login button

void action (args)
{
        String u = username.getText(); // retrieve both username and password
        String p = pass.getText();

        if (u == usernameindb and p == passindb) // check if valid username and password
        {                                        // combo. Assume DB is present already
                showMessage("Log in successful")

                LoginPage logpage = new LoginPage(); //Access login page
                logpage.main();
        }

        else //Error message
        {
                showMessage("Error. Invalid username and password")
        }
}
```

Browse GUI(Visitor):

```
Establish connection to DB;
Query DB for top 3 rated projects;
Query DB for top 3 rated users;
Display projects;
Display users;
```

Register:

```
label rform = new label("Registration Form:"); //Create Registration form label
getcontent().add(rform);
label fn = new label("First Name*:"); //Create First Name label
getcontent().add(fn);
label ln = new label("Last Name*:"); //Create Last Name label
getcontent().add(ln);
label DOB = new label("DOB*:"); //Create DOB label
getcontent().add(DOB);
label email = new label("Email*:"); //Create Email label
getcontent().add(email);
label interestlabel = new label("Interest*:"); //Create Interest label
getcontent().add(interestlabel);
label username = new label("Username*:"); //Create Username label
getcontent().add(username);
label password = new label("Password*:"); //Create Password label
getcontent().add(password);
label confirm = new label("Confirm Password*:"); //Create Confirm Password label
getcontent().add(confirm);
label reference = new label("Reference:*"); //Create Reference label
getcontent().add(reference);
label refuser = new label("Username of Referrer:"); //Create Referred Username label
getcontent().add(refuser);
label refstatus = new label("Status:"); //Create Referred Username status label
getcontent().add(refstatus);
Button submit = new Button(submit); // Have Submit button
submit.listen(new listen);

firstName = new Textfield(); // Retrieve first name
getcontent().add(firstName);
lastName = new Textfield(); // Retrieve last Name
getcontent().add(lastName);
birth = new Textfield(); // Retrieve Date of Birth
getcontent().add(birth);
```

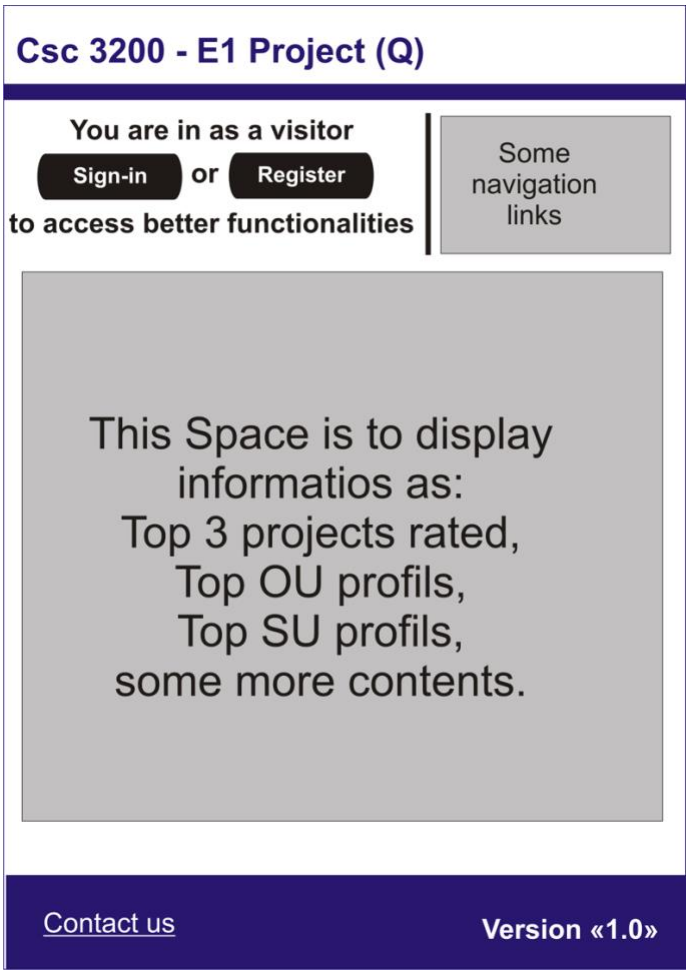
```
eaddress = new Textfield(); // Retrieve email address
getcontent().add(eaddress);
inter = new Textfield(); // Retrieve user interest
getcontent().add(inter);
usnm = new Textfield(); // Retrieve user input username
getcontent().add(usnm);
pword = new Textfield(); // Retrieve user input password
getcontent().add(pword);
conpword = new Textfield(); // Retrieve confirm password
getcontent().add(conpword);
ruser = new Textfield(); // Retrieve referred username
getcontent().add(ruser);
rstat = new Textfield(); // Retrieve reffered user status
getcontent().add(rstat);

Update DB applicants table with input data;
```

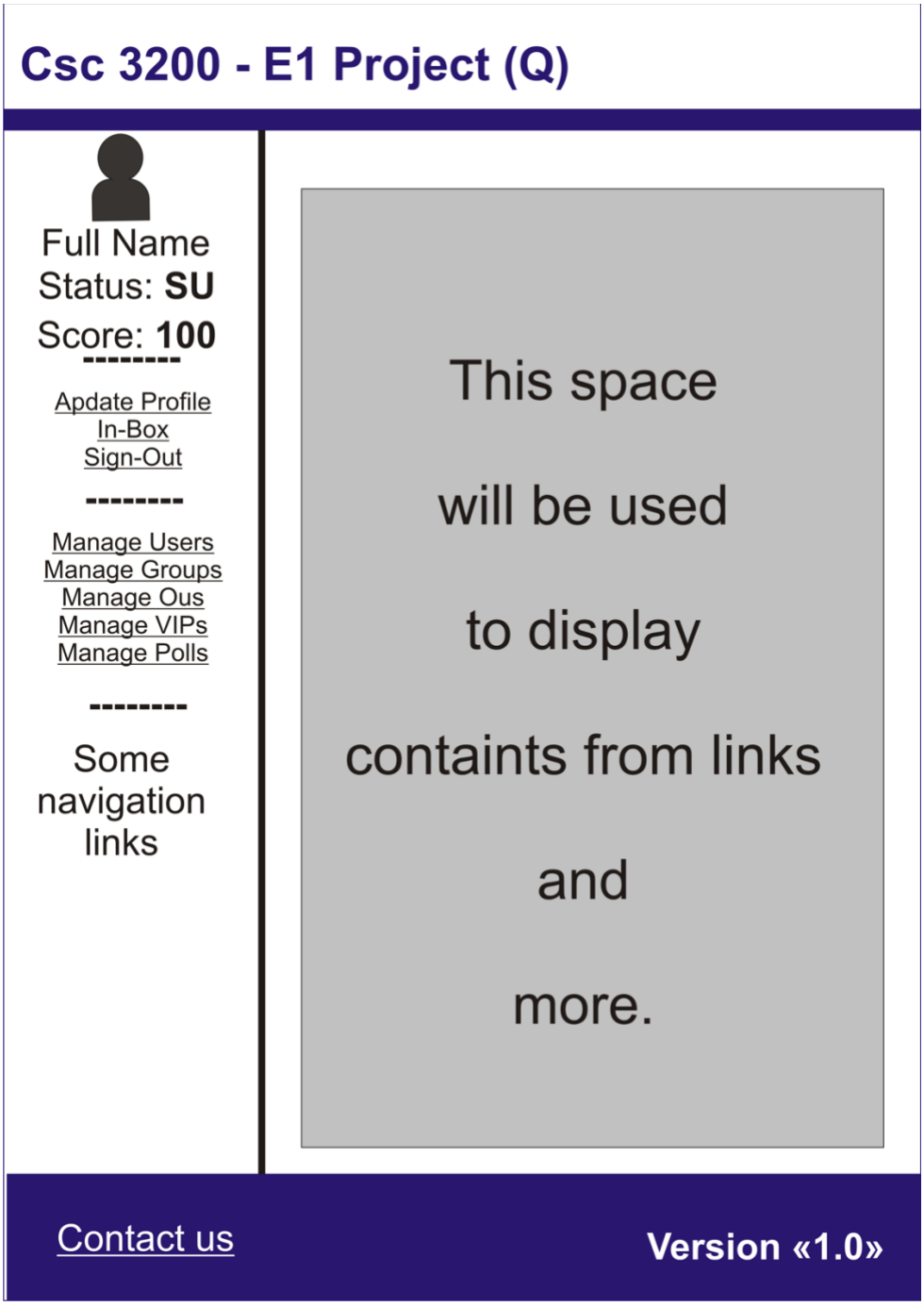
9. System Screens:

Welcome Screen:

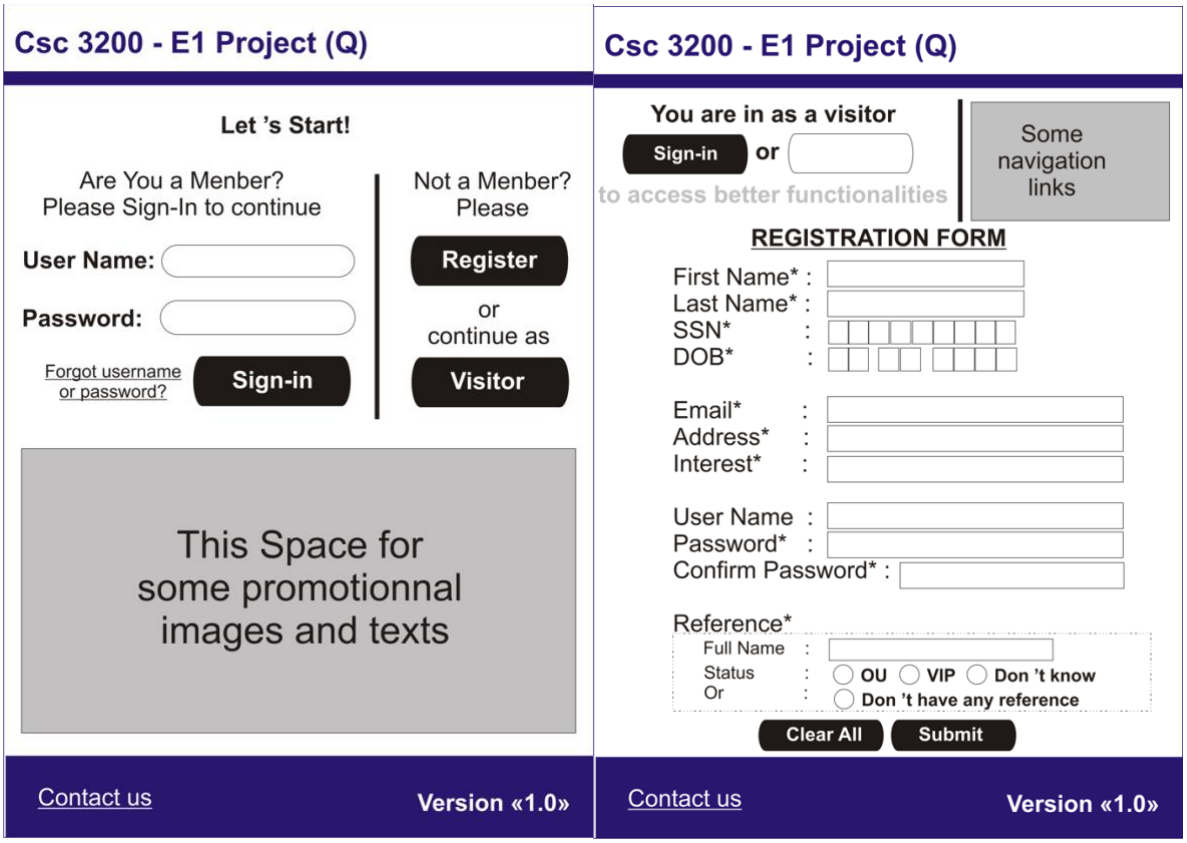
Screen for Visitors



Screen of user logged in the system.



(Registration Prototype)



1. When the user opens the app, he is given the option to browse as a visitor or register.
2. If he decides to register, he will be presented with an application that has to be fill out with personal information. In addition, the user should have a referral from a user who is in the system already.
3. Next, when the visitor finishes filling out the application he needs to is click submit.
4. The application will be sent to a super user, which will review the application and decide if the applicant is approved or rejected.

10. Group Meetings

Meeting #	Topic	Minutes
1	Report and Diagrams	60min
2	Discuss about all the functions required by the system.	60min
3	Discussion about the layout and design of the System.	60min

11. Github Repo:

https://github.com/ZGabriell/ActiveTeamingSystem