TeamUp

Active Teaming System

Software Design Report

For Web Application

Version 1.1.0

Active Teaming System	Version 1.1.0
Software Design Report	Date: 05/01/2020
TeamUp Phase 2	

Revision History

Date	Version	Description	Author
02/26/2020	1.0.0	Create Software Requirement Specification	Bryan Rivas Edgar Pineda Shahzeb Rizvi Hyunmin Cho
04/13/2020	1.1.0	Create Software Design Specification	Bryan Rivas Edgar Pineda Shahzeb Rizvi Hyunmin Cho

Table of Contents

Introduction 4

Use Cases 6

ER Diagrams 21

Detailed Design 22

System Screens 26

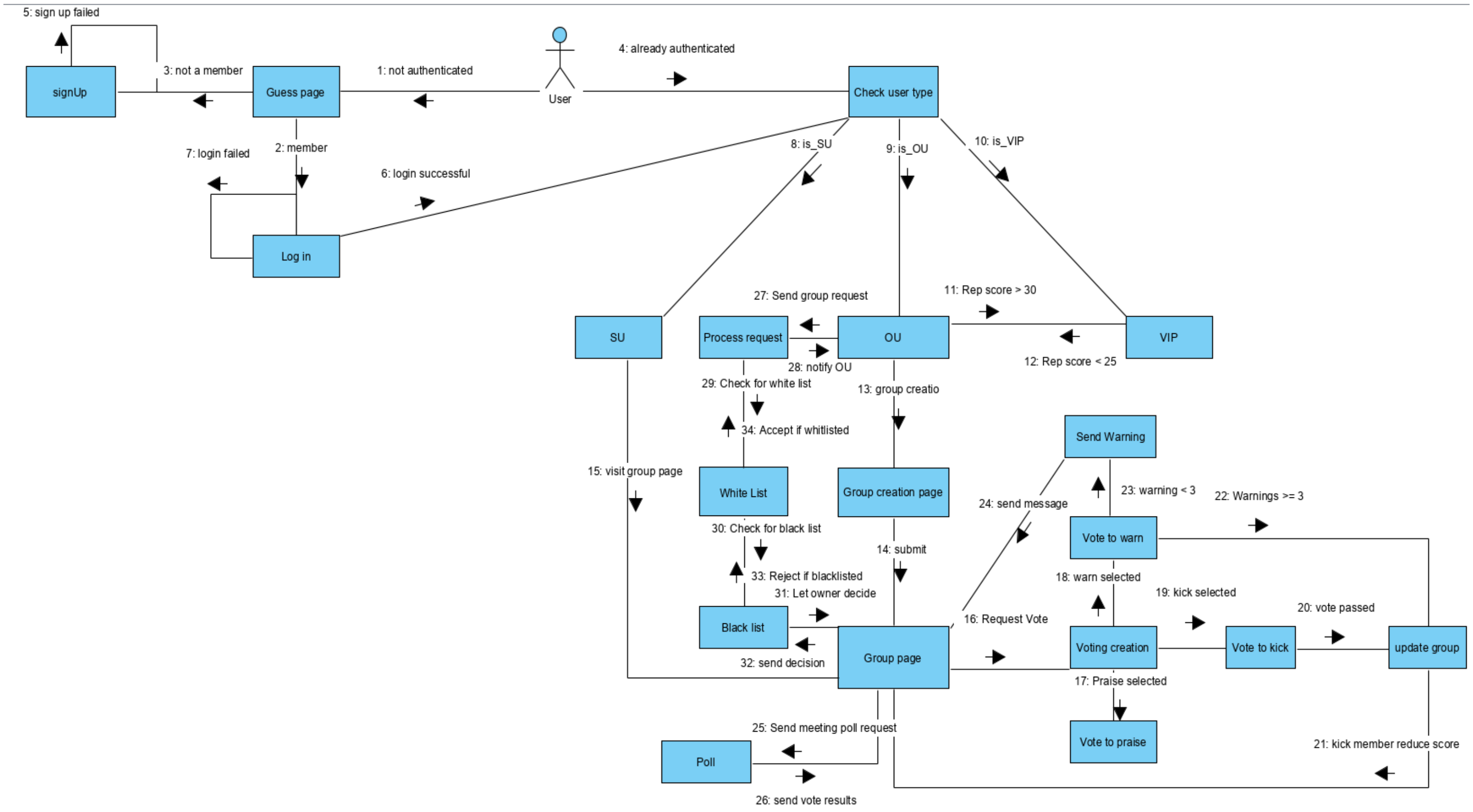
Github repo 38

1. Introduction

1.1 Purpose

This is a report that details the design and functionality of our Active Teaming System App. The goal is for the reader to understand how the Active Teaming System will function and what it aims to achieve.

1.2 Scope Below is a collaboration diagram that shows how OU's, SU's and VIP's interact with the active teaming system and the level of access that each user has: whitelist, blacklist, guess page, group creation, group page, etc.



1.3 Definitions

GM: Group member, any Ordinary user or VIP user that belongs to a group.

GU: Guest user, a user who has not registered for an account yet. This user has limited access and is only able to see the top three rate projects and top three OU's.

OU: Ordinary user, a user who has register for an account and whose reputation score is < than 25 points. This user can do everything a GU does in addition to being able to create groups by inviting other OU's to join his/her group.

VIP: VIP user, an ordinary user whose reputation score is greater than thirty points. This user can do what the former users can. In addition, VIP's can be appointed by SU's to evaluate a group after closure and determine the reputation score of each group member.

SU: Super user, this type of user is divided into two categories:

1. One founding super user who initializes the system. 2
2. . One democratic super-user who is a VIP user that is voted on by other VIP users.

White Box: A list of pre-approved users who would join a group automatically upon request.

Black Box: A list of users whose request to join a group would get automatically Declined if his/her name is on the group creator's blacklist.

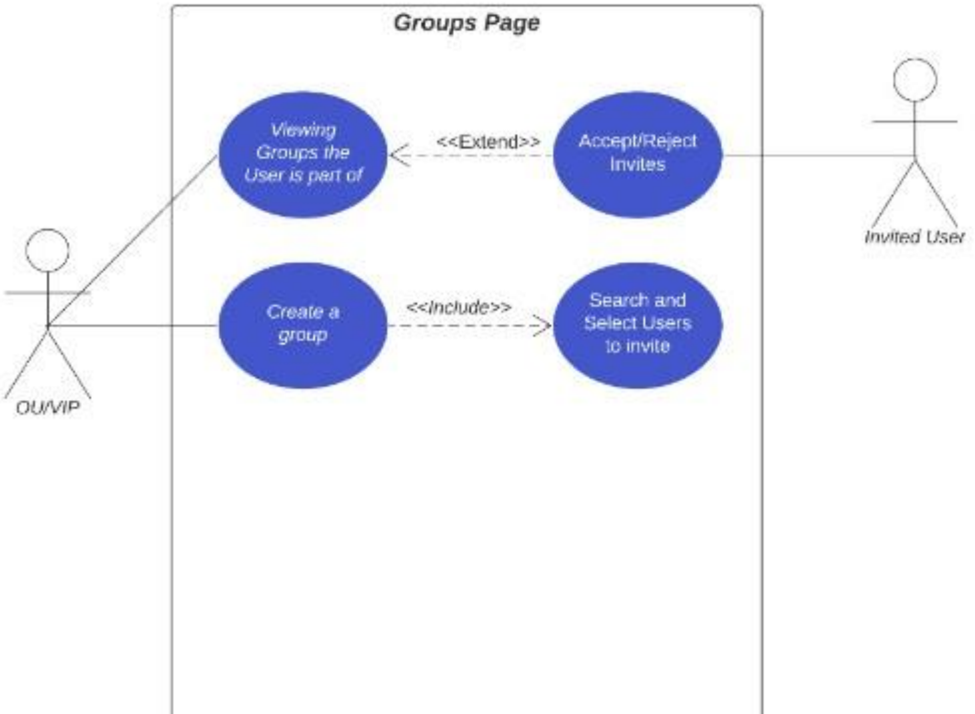
2. Use Cases

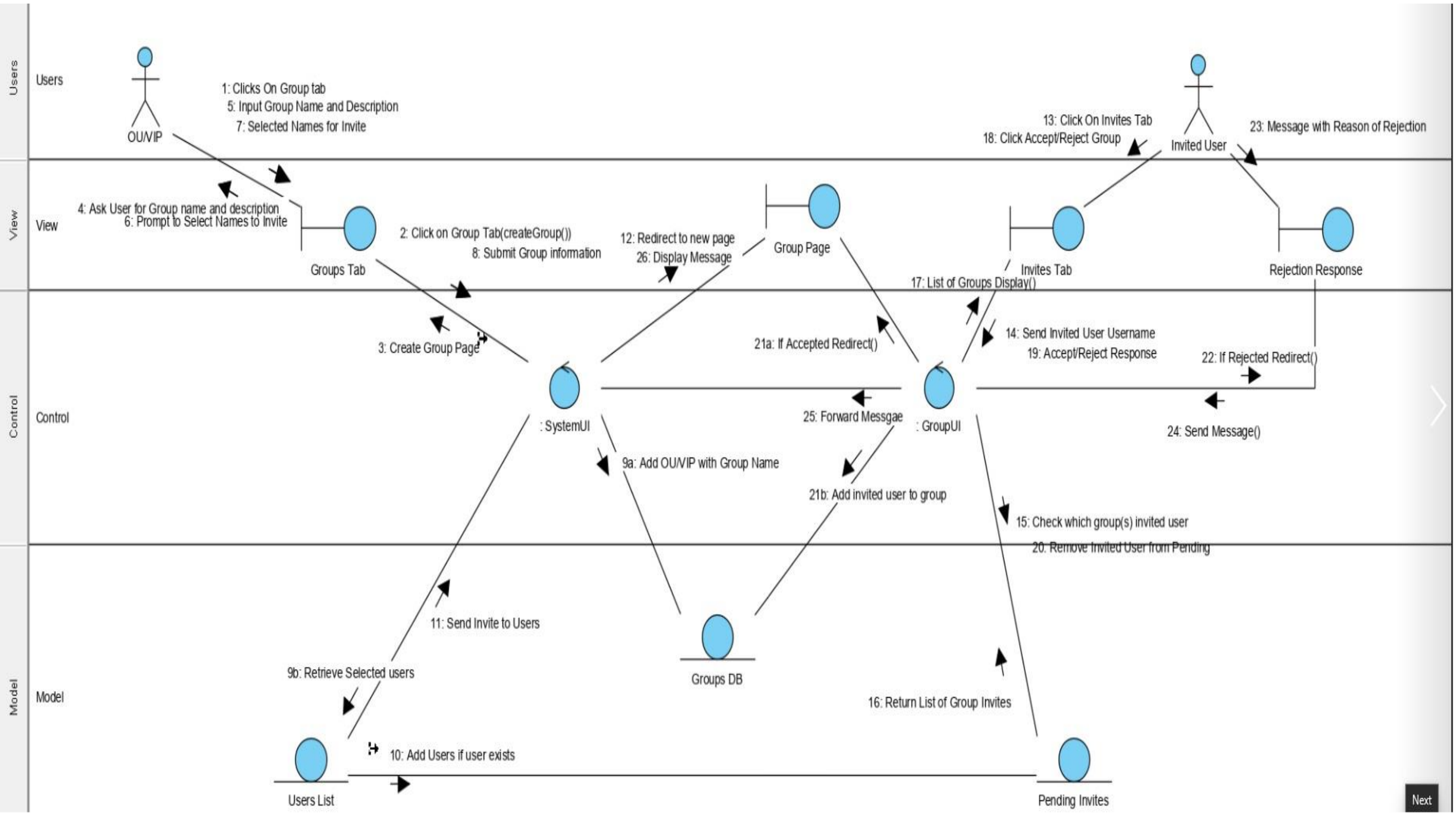
2.1 Creating Groups

Common Case: When OU/VIP reach the groups page they will be presented all the groups they are part of and on the side there will be an option of creating a group. Once clicked on that option the user will get a pop-up prompting them for a Group name and description. Once entered and submitted a second prompt will ask for the users they want to add to the group. The users will be able to enter multiple Usernames. Once all the usernames are added to the list, the user clicks the invite button and will be directed to the newly created group page where the User can take further group actions.

If a user gets a group invite they can accept the invite and will be directed to the invited group page or they can reject the invite and will be required to give a reason as to why they rejected. The response will be sent to the user that sent the group invite.

Exceptional Case: GU will not have the groups page option since they can not form groups nor be part of them.

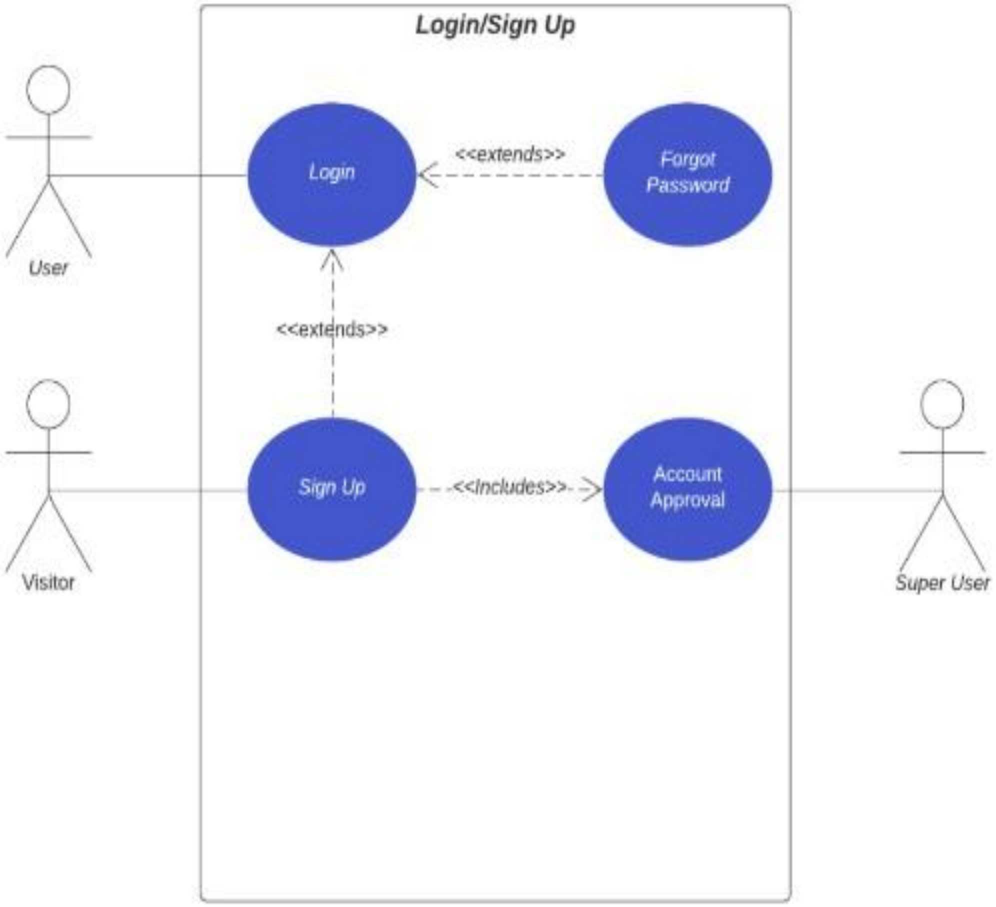


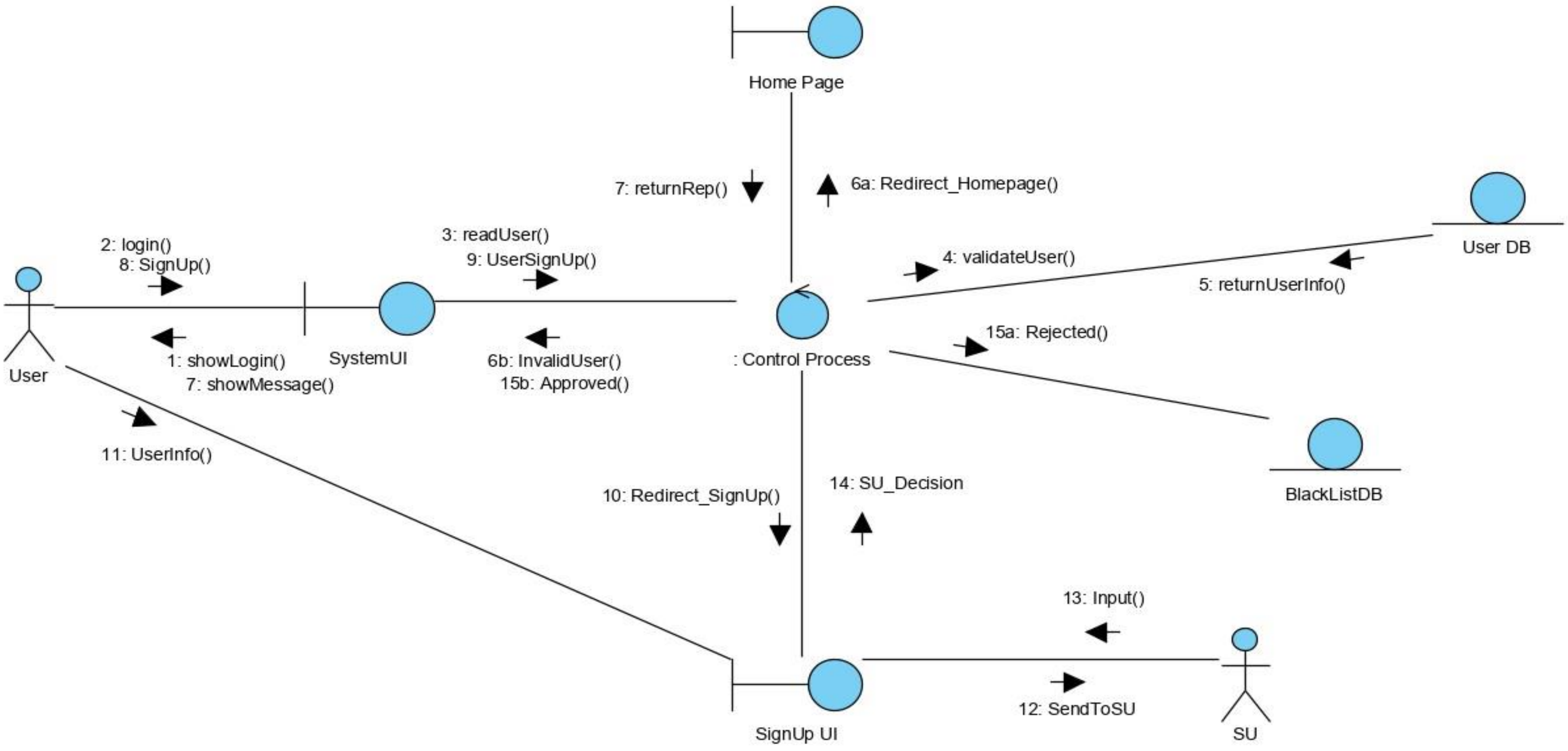

Users
View
Control
Model
OU/VIP
1: Clicks On Group tab
5: Input Group Name and Description
7: Selected Names for Invite
4: Ask User for Group name and description
6: Prompt to Select Names to Invite
Groups Tab
2: Click on Group Tab(createGroup())
8: Submit Group information
3: Create Group Page
12: Redirect to new page
26: Display Message
Group Page
: SystemUI
9a: Add OU/VIP with Group Name
25: Forward Messgae
21a: If Accepted Redirect()
21b: Add invited user to group
: GroupUI
17: List of Groups Display()
Invites Tab
14: Send Invited User Username
19: Accept/Reject Response
13: Click On Invites Tab
18: Click Accept/Reject Group
Invited User
23: Message with Reason of Rejection
Rejection Response
22: If Rejected Redirect()
24: Send Message()
15: Check which group(s) invited user
20: Remove Invited User from Pending
11: Send Invite to Users
9b: Retrieve Selected users
Groups DB
16: Return List of Group Invites
10: Add Users if user exists
Users List
Pending Invites
Next

2.2 Accessing Home Page

Common Case : When users reach the homepage they will be greeted with a welcome screen with a login box where they can input their username and password. Upon entering the correct credentials and based on their User role they will be transported to their home page where the which will show them the projects that they are currently working on. If they are a SU they will have a part of the page showing certain requests they have to fulfill.

Exceptional Case: New users will have the option to visit the site as a guest user. They will be transported to the homepage showcasing the top 3 rated projects and top rating OU profiles and SU profiles to showcase the power of the system. A visitor can surf around to find more OUs/VIPs and projects. New Users will have the option of signing up by clicking on sign up on the login page. page, those users who are blacklisted will not not be able to sign up for the site. After signing up, the request will be sent to a SU to accept or reject the request. Once accepted the new the SU will send an email with account id and password, when the new OU first login, s/he is required to change the password. If rejected, the applicant has one chance to appeal and the SU will make a final decision to reverse the rejection: if still reject, then this visitor will be put in blacklist forever. Additionally, if password or username is wrong an error message will appear indicating that their credentials were wrong.

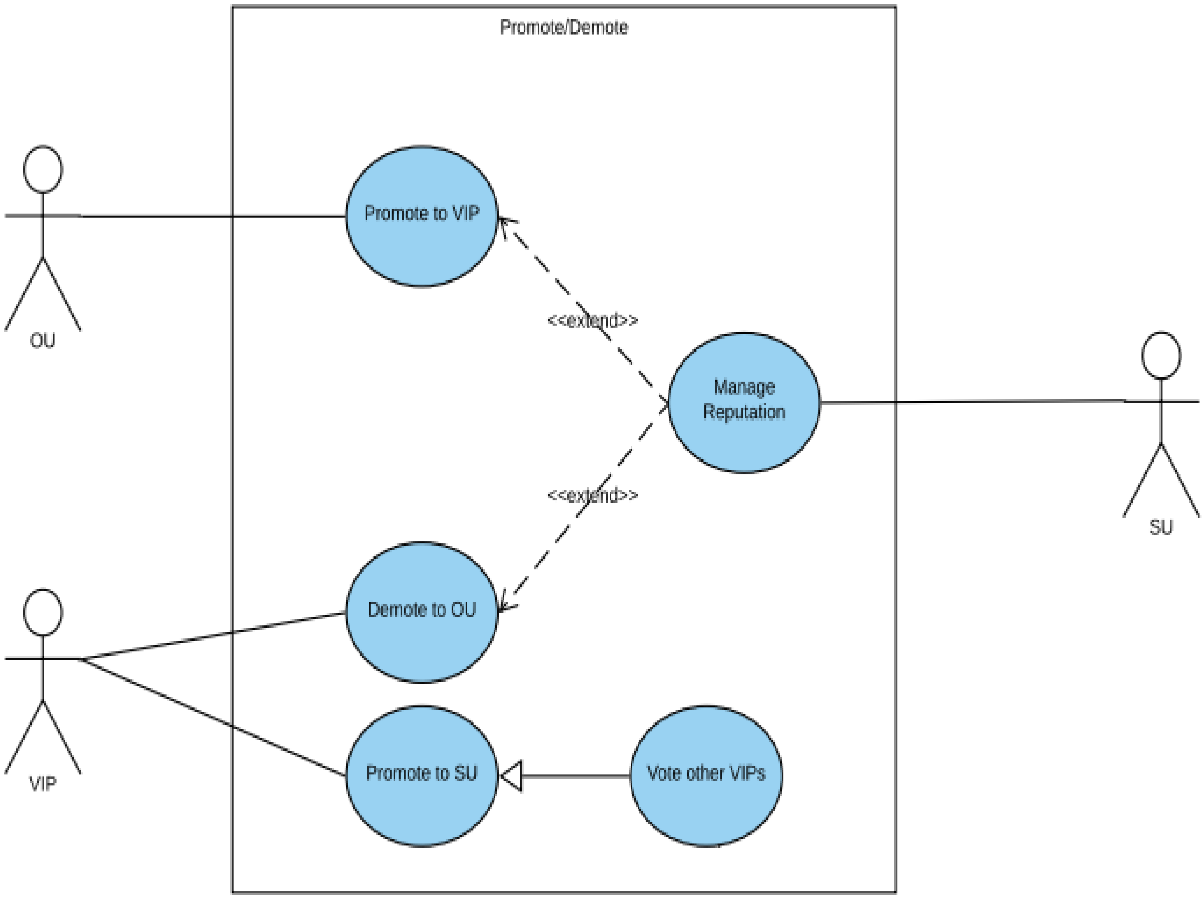

Login/Sign Up
User
Login
<<extends>>
Forgot Password
<<extends>>
Visitor
Sign Up
<<Includes>>
Account Approval
Super User

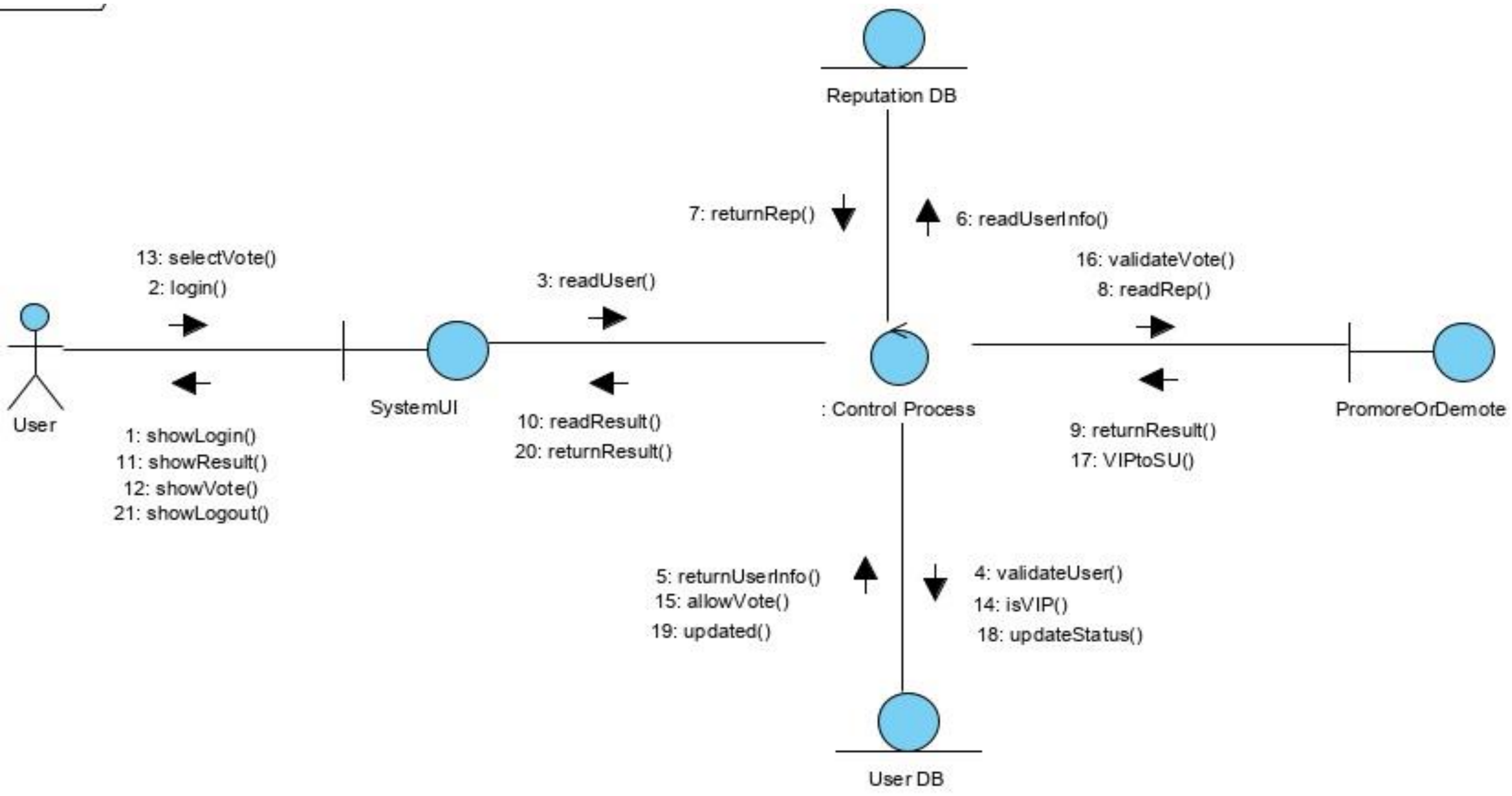

Home Page
7: returnRep()
6a: Redirect_Homepage()
User DB
2: login()
8: SignUp()
3: readUser()
9: UserSignUp()
4: validateUser()
5: returnUserInfo()
User
SystemUI
1: showLogin()
7: showMessage()
6b: InvalidUser()
15b: Approved()
: Control Process
15a: Rejected()
BlackListDB
11: UserInfo()
10: Redirect_SignUp()
14: SU_Decision
13: Input()
12: SendToSU
SignUp UI
SU

2.3 Promoting/Demoting

Common case: OU can get promoted if reputation score greater than 30 will be promoted to VIP. VIP gets demoted if reputation score falls below 25. All VIP's can vote for a VIP to be promoted to SU.

Exceptional Case: GU request can be rejected by a SU. GU can send another request if rejected again GU will be blacklisted.

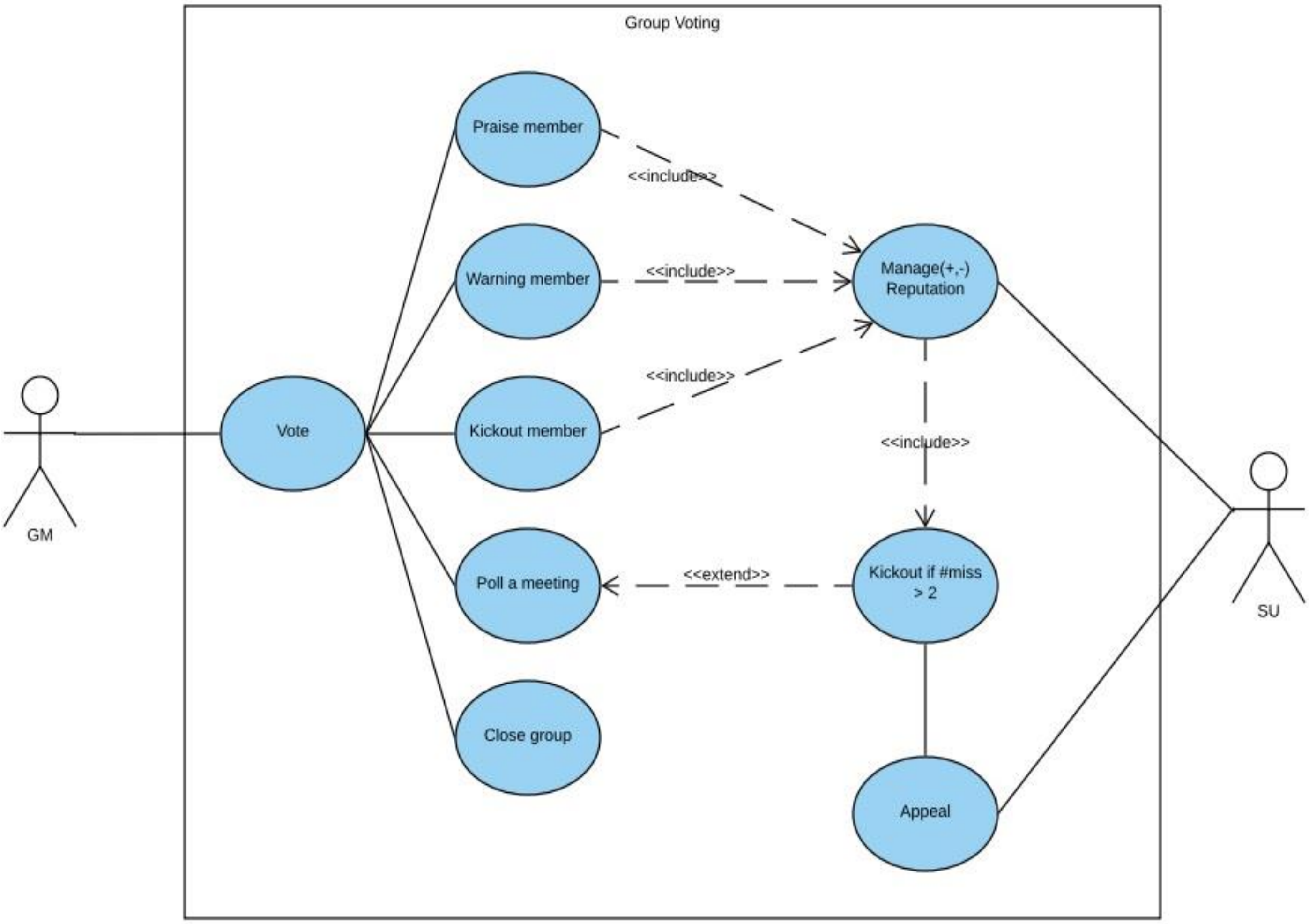


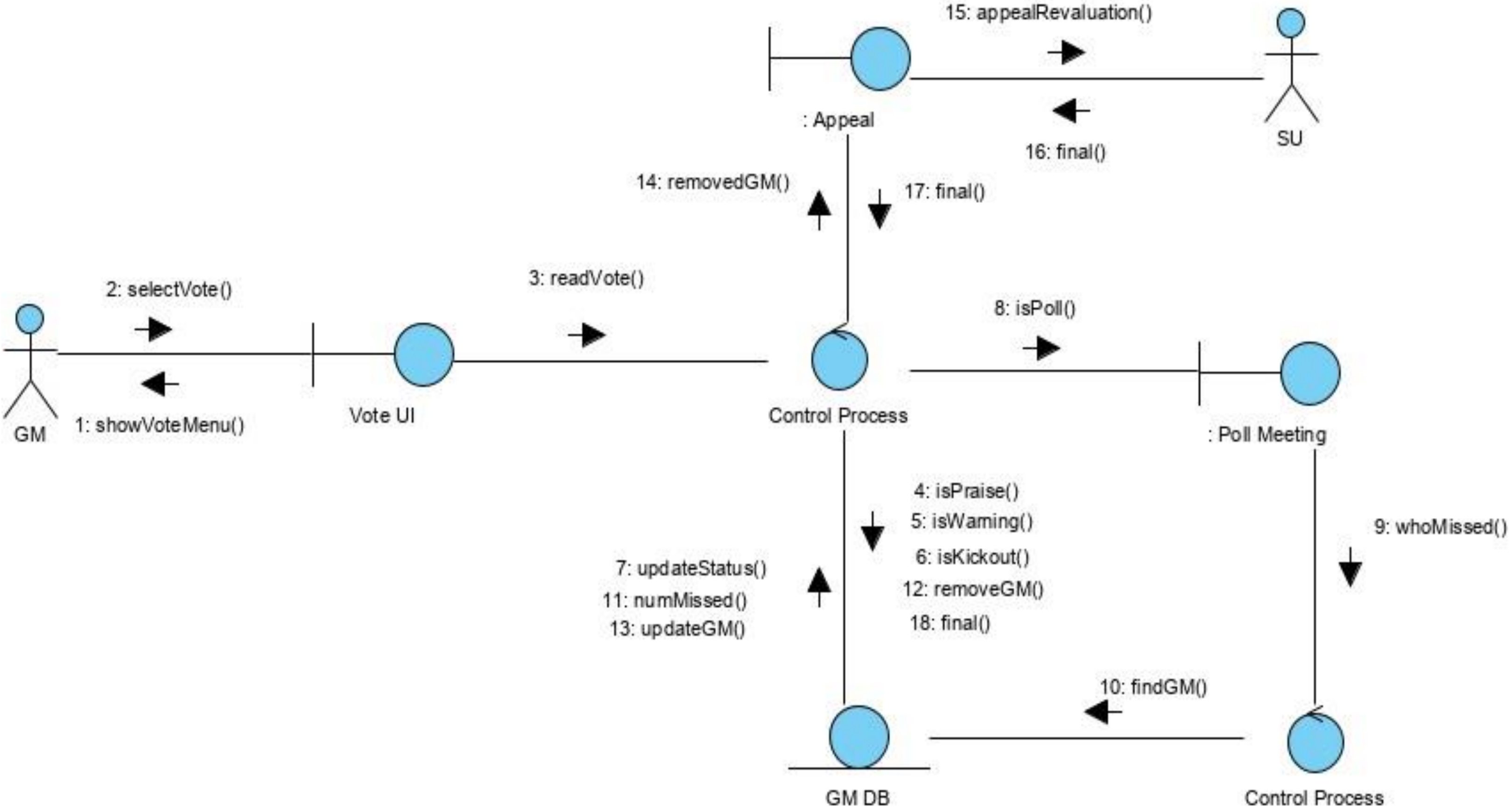

Reputation DB
7: returnRep()
6: readUserInfo()
13: selectVote()
2: login()
3: readUser()
16: validateVote()
8: readRep()
User
SystemUI
: Control Process
PromoreOrDemote
1: showLogin()
11: showResult()
12: showVote()
21: showLogout()
10: readResult()
20: returnResult()
9: returnResult()
17: VIPtoSU()
5: returnUserInfo()
15: allowVote()
19: updated()
4: validateUser()
14: isVIP()
18: updateStatus()
User DB

2.4 Group Voting:

Common case: Group members can initiate a vote for a meeting, warning a group member, kicking a group member, praise group member can close group. The votes for kicking a group member, praising a group member or warning a group member must be unanimous. Praising a group member will reward the member with reputation score. Receiving a warning will reduce the members reputation score by 5 and kicking a member will reduce reputation score by 10. Voting for a meeting will be started by any group member each group member will choose the time that best fits for them. The time slot with the most votes will be when the meeting is held. If a group member misses 2 meetings, they will be kicked out the group. That member will have the option to appeal. A SU will look at the appeal and decided if the member should get the reputation points back.

Exceptional Case: GU does not have groups voting option since they cannot form groups nor be part of them.



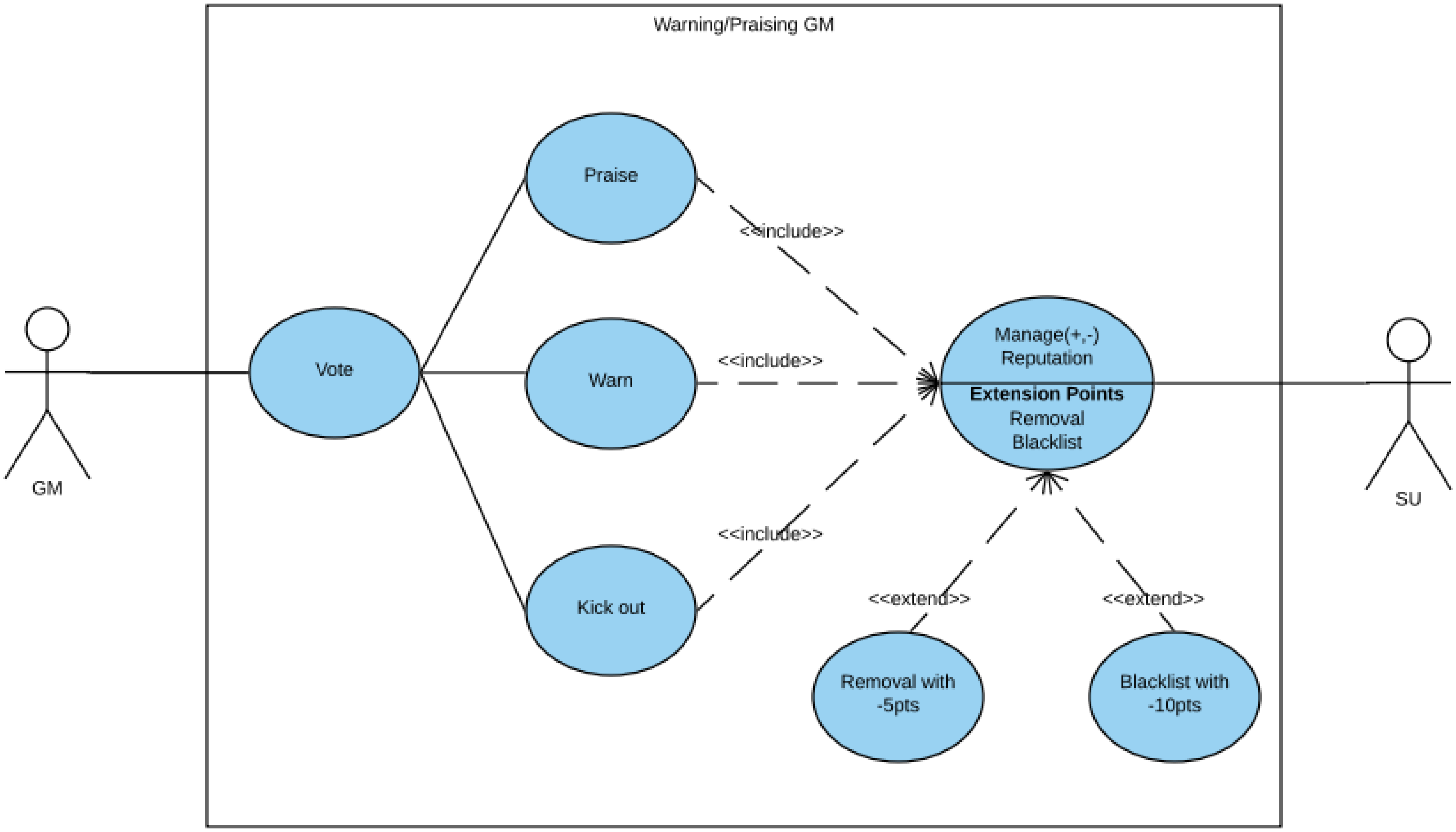

15: appealRevaluation()
16: final()
: Appeal
SU
14: removedGM()
17: final()
2: selectVote()
3: readVote()
8: isPoll()
GM
1: showVoteMenu()
Vote UI
Control Process
: Poll Meeting
4: isPraise()
5: isWarning()
6: isKickout()
12: removeGM()
18: final()
7: updateStatus()
11: numMissed()
13: updateGM()
9: whoMissed()
10: findGM()
GM DB
Control Process

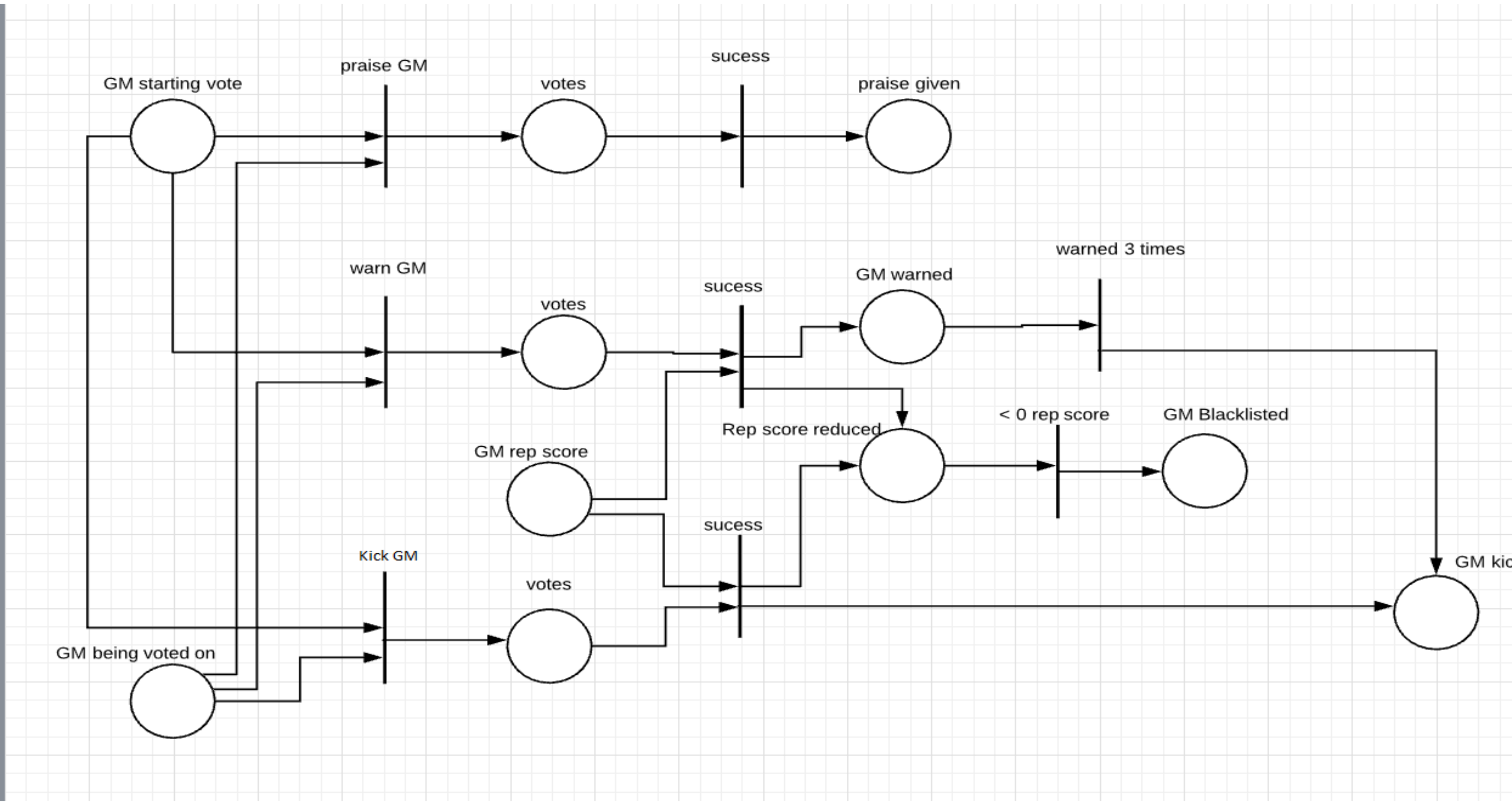
2.5 Warning/Praising Group Members

The group members (GM) can vote to warn, praise or kick-out to a group member. Either a GM gets removed from the group with deduction of 5 pts when the GM gets 3 warnings, or a GM gets kicked out with deduction of 10 pts. Any GM, who has reputation score less than zero is sent to blacklist.

Common Case: any GM could use vote options to either praise, warn or kick out other GMs, system affects all OUs in the group.

Exceptional Case: any OU, who has been kicked out, receives a message that the OU has been kicked out (automatic removal) or blacklisted after the OU login.



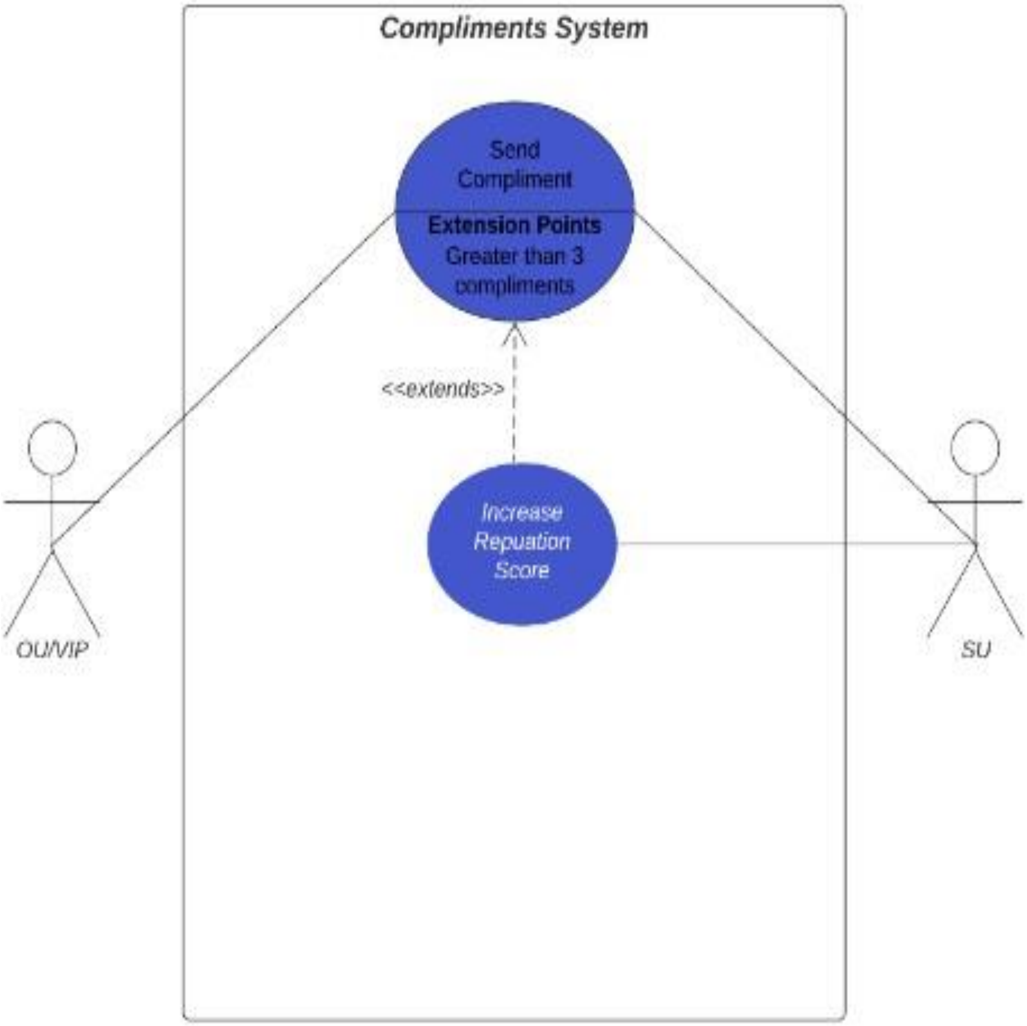

GM starting vote
praise GM
votes
sucess
praise given
warn GM
votes
sucess
GM warned
warned 3 times
GM rep score
Rep score reduced
< 0 rep score
GM Blacklisted
sucess
Kick GM
votes
GM kic
GM being voted on

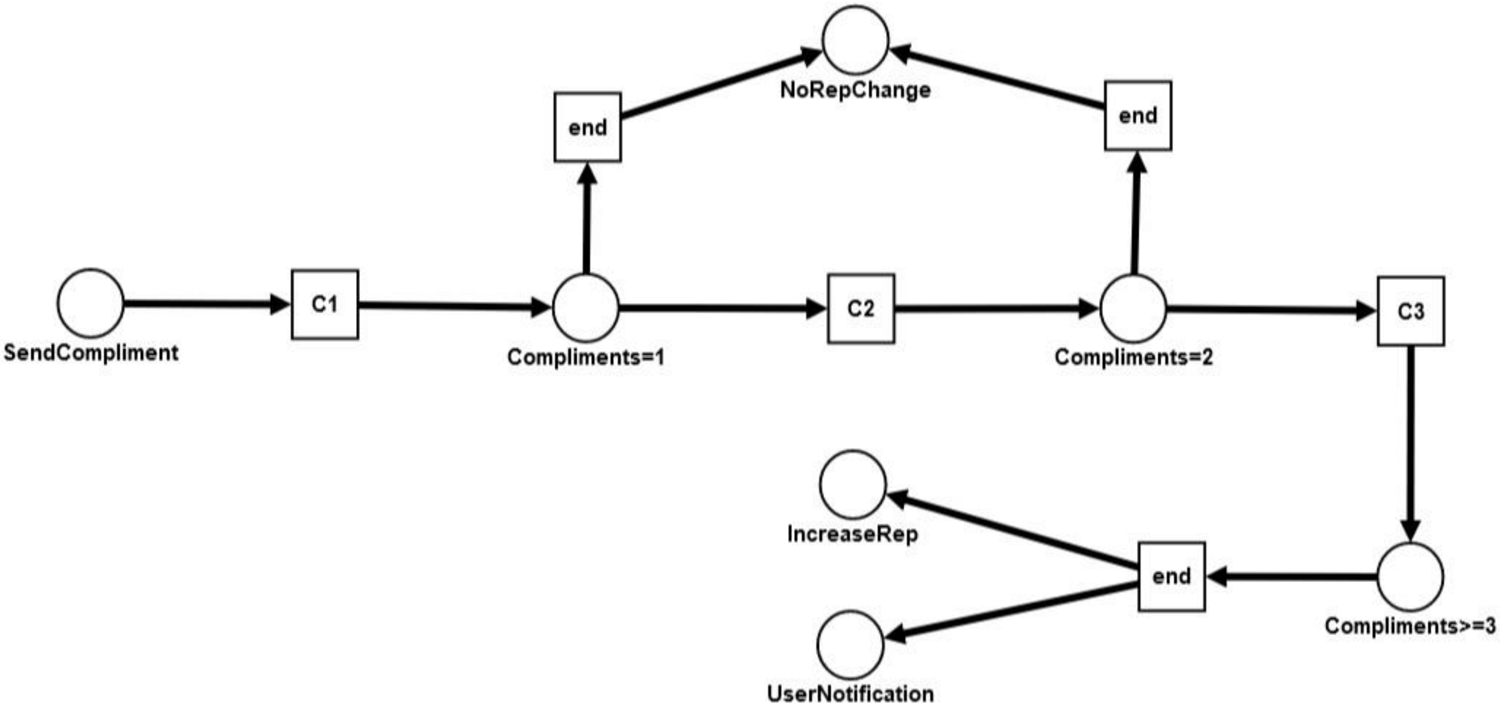
2.6 Compliment System

OU sends compliment of other OU to SU and SU keeps all compliments and counts them. SU increases reputation score of certain OU when SU receives more than three compliments of the OU.

Common Case: any OU can send compliments to SU for another OU without limitation.

Exceptional Case: any OU receives a notification saying that OU's reputation score increased due to three or more compliments from other OU's.





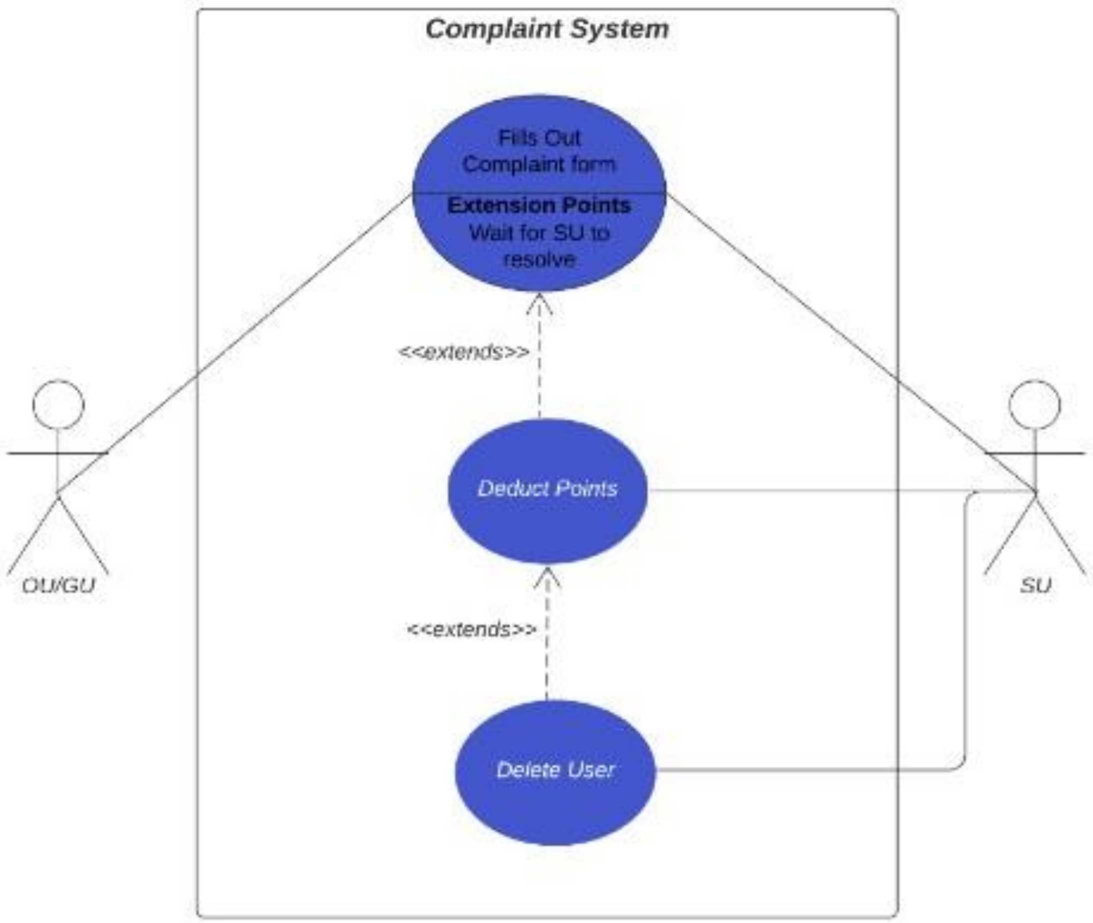
NoRepChange
end
end
SendCompliment
C1
Compliments=1
C2
Compliments=2
C3
IncreaseRep
end
UserNotification
Compliments>=3

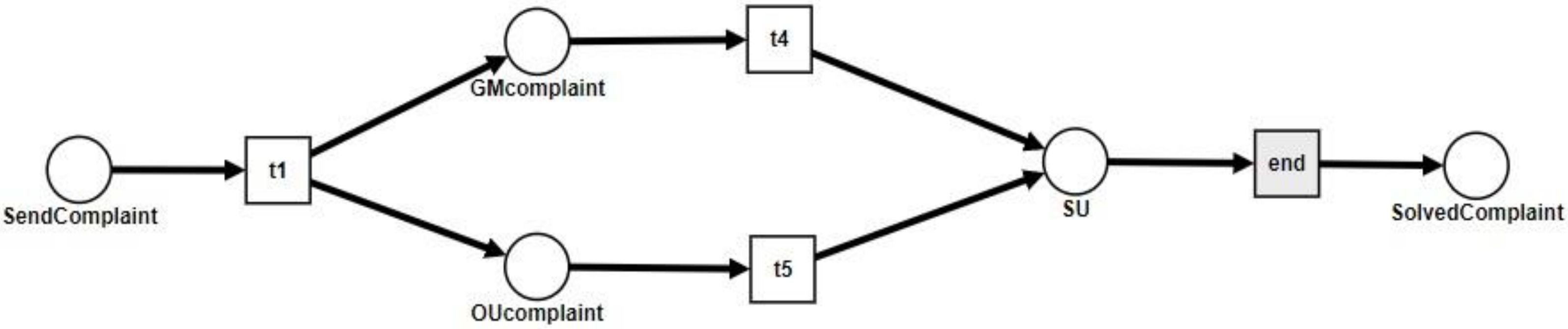
2.7 Processing complaints

Visitors and OU's can complain or voice their discontent over a group or a particular group member. The SU will review said complaint and decide if it merits an action or not.

Common Case: SU will/can deduct any amount of points that he/she deems necessary from any member of the group or particular OU.

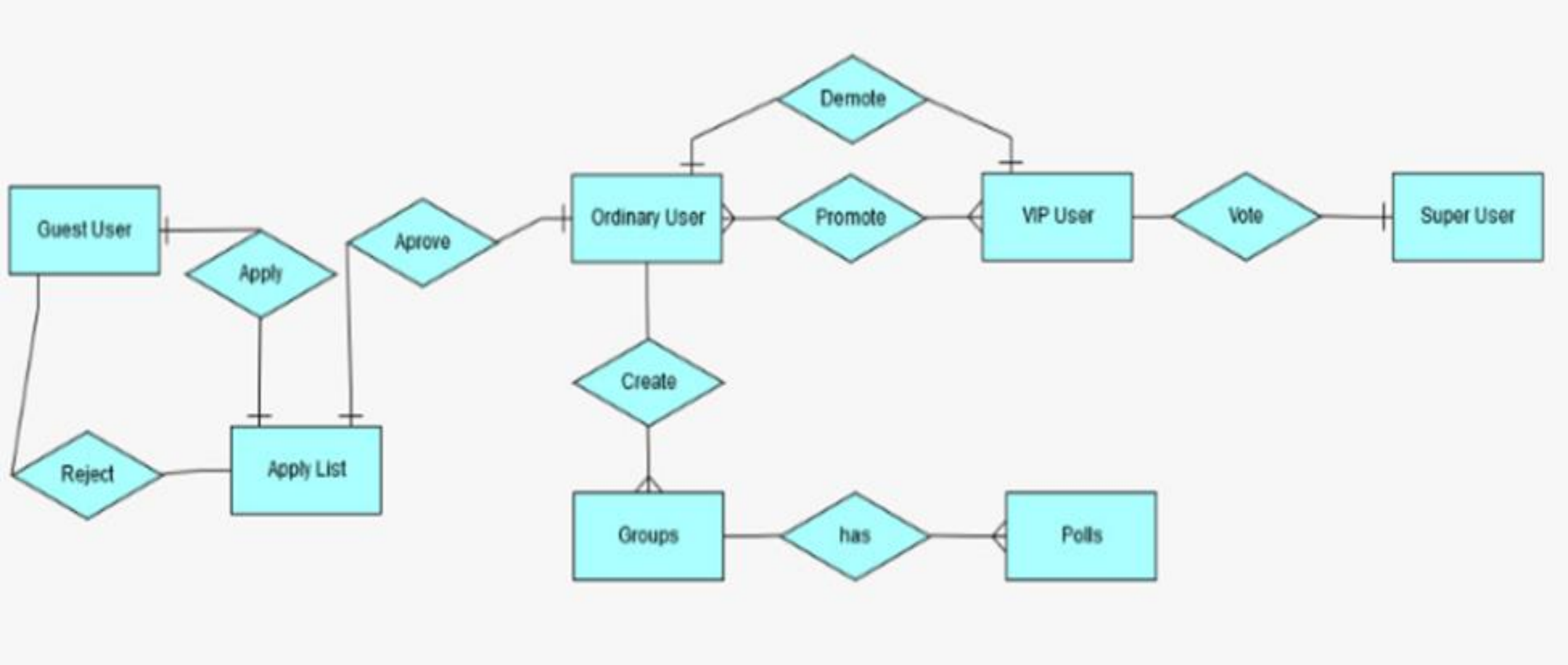
Exceptional Case: SU reserves the right to permanently kick said group member or OU from the site.




SendComplaint
t1
GMcomplaint
OUcomplaint
t4
t5
SU
end
SolvedComplaint

3. E-R Diagram

This is an Entity Relation Diagram shows an overview of our system’s database.



4. Detailed design

The following functions showcase the main functionalities of the TeamUp App (web-based application). The pseudo code shown here is more of a general concept of the methods/functions used to develop our app. However, it does provide guidelines on how we have implemented said methods.

- **Create Group**
  - def createGroup(request):

```
Triggered by create group button press
Redirect user to create group page
User fills in info
If(user clicks create)
    Info is verified
    Post request to database
If(User clicks cancel)
    Exit create group page
```

- **Sign Up**
  - def signUp(request):

```
Triggered by signup button press
Prompt the user to fill out information
If(User clicks on signup)
    Information is verified
    If(Valid format)
        Post to database
    Else
        Ask the user to input info again
Exit signup page and wait for approval
```

- **Log in**
  - def logIn(request):

```
Triggered by signin button press
If(username in database && password in database matches one in the
users table)
    Send user to home page
Else
    Print(incorrect username or pasword)
```

- **Log out**
  - ➢ def logout(request):

```
Triggered by logout button press
Send user to guess homepage
```

- **Invite to group**
  - ➢ def inviteToGroup(request):

```
On button press
Prompt user to enter in a user name
If(user exist)
    Send invite
Else
    Print(user does not exist)
```

- **Check user type**
  - ➢ checkUserType(request):

```
on login for a given username check the permissions column on the
User table
if(permission = 0 )
    OU
If(permission = 1)
    VIP
If(permission = 2)
    SU
```

- **Warn/Praise user**

  - def warnPraise(request):

```
Triggered on Vote button press from the group page
If(user selects praise)
    Prompt for which user he/she wants to praise
    Notify user being praise
    Check for how many times user has been praised
    If(User praised >=3)
        Increase rep score
    Else
        Exit
If(user selects warn):
    Prompt for which user he/she wants to praise
    Create poll excluding the user being warned
    If(Vote is unanimous)
        Warn user
        If(warnings >=3)
            Kcik user
    Else
        Exit
```

- **Leave group**

  - def leaveGroup(request):

```
Triggered by leave group button press
Prompt the user if they are sure they want to leave
If(yes)
    Update group model
If(No)
    Go back to group page
```

- **Complain**

  - def fileComplaint(request):

```
Triggered on file complaint button press
Prompt the user for the username he/she is complaining about
If(username exist)
    Prompt user to enter details
    If(press submit)
        Send complain to SU to handle
    If(press cancel)
        Go back to main page
If(username does not exist)
    Inform the user complaining that they got the wrong user name

If(user press cancel)
    Go back to main page
```

- **Handle complaint**

  - def handleComplaint():

```
the super user gets a notification and reviews the complaint
if(SU selects that the complaint is valid)
    ask the SU to type feedback for the person being complaint about
    ask the super user for a punishment
    if(cancel button press)
        go back to main page
    if(submit button press)
        POST to dabase
        notify the user complaining
        notify the user being complaint about
```

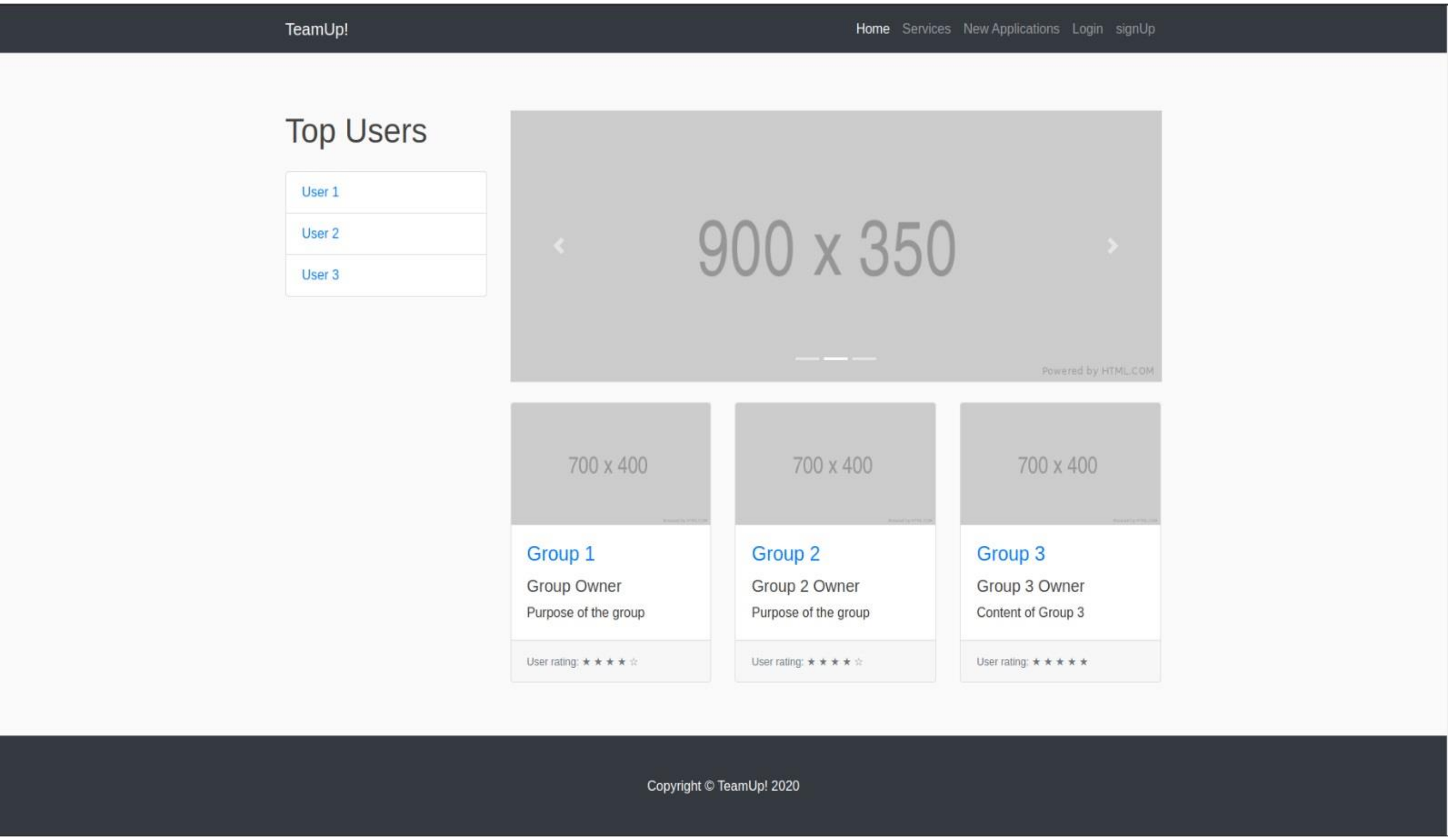
- **Check if user exist**

```
➢ def exist(request):
        search for username in user table
        If(username is foud)
            Return true
        Else
            Return false
```

5. System Screens

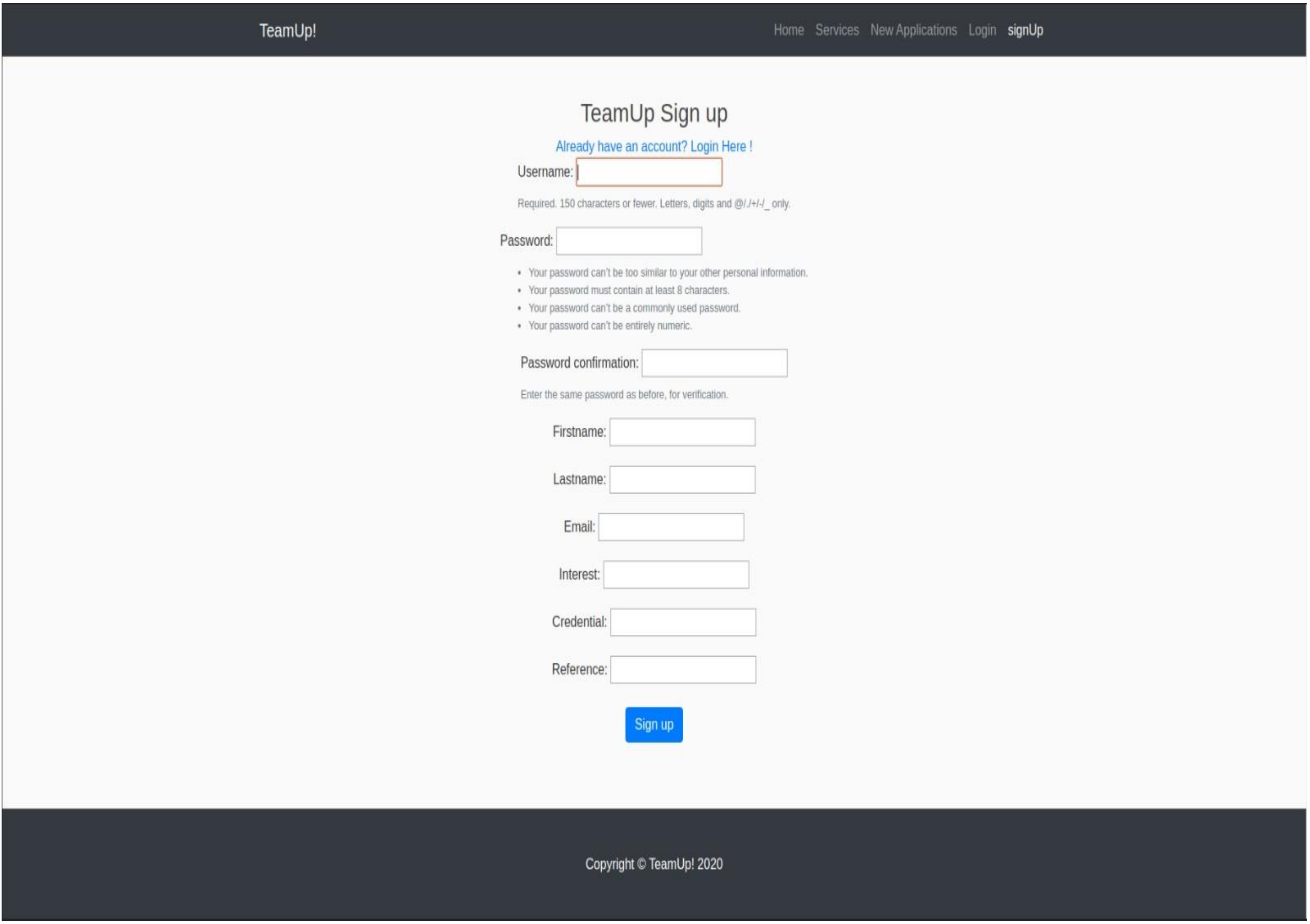
These are some of the major GUI screens in our Active TeamUp (web-base application) system. This is our initial design and definitely needs improvement, some of the images used here are placed holder just for demonstration purposes. This section will show case the (**Accepting user functionality**) as well as the sign up, log in, and group page

Home page



On this screen, a guess user can see the top rated user and the top rated groups our TeamUp(web-based application). A user can choose to log in, sign up or just browse the site from this page.

Signup page



TeamUp Sign up

Already have an account? Login Here !

Username:

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation:

Enter the same password as before, for verification.

Firstname:

Lastname:

Email:

Interest:

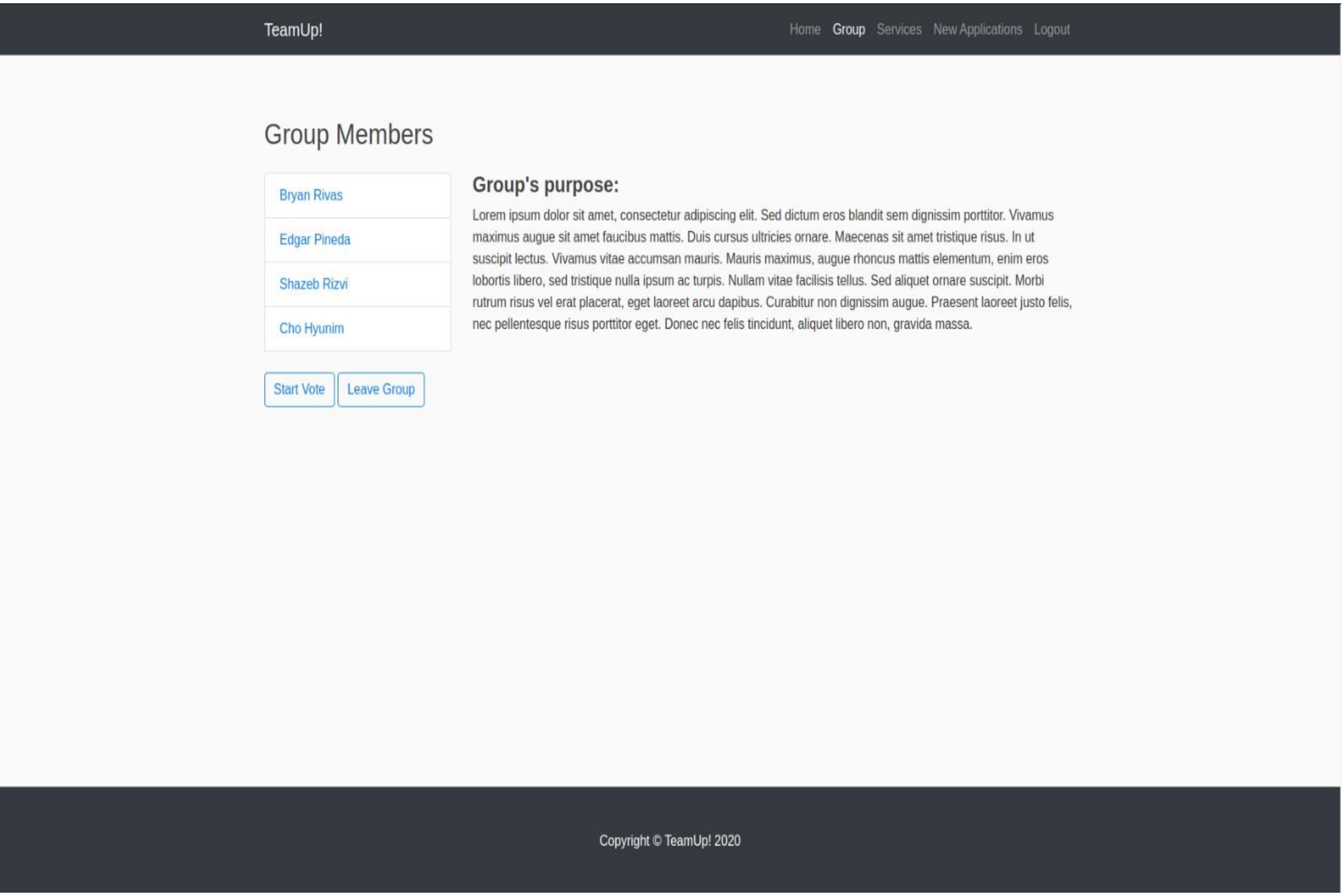
Credential:

Reference:

Copyright © TeamUp! 2020

On the sign-up page the user can fill out his or her information and if the information is within the guidelines the signup will be successful and the user will be pending for approval from the super user.

Group Page



Group Members

- Bryan Rivas
- Edgar Pineda
- Shazeb Rizvi
- Cho Hyunim

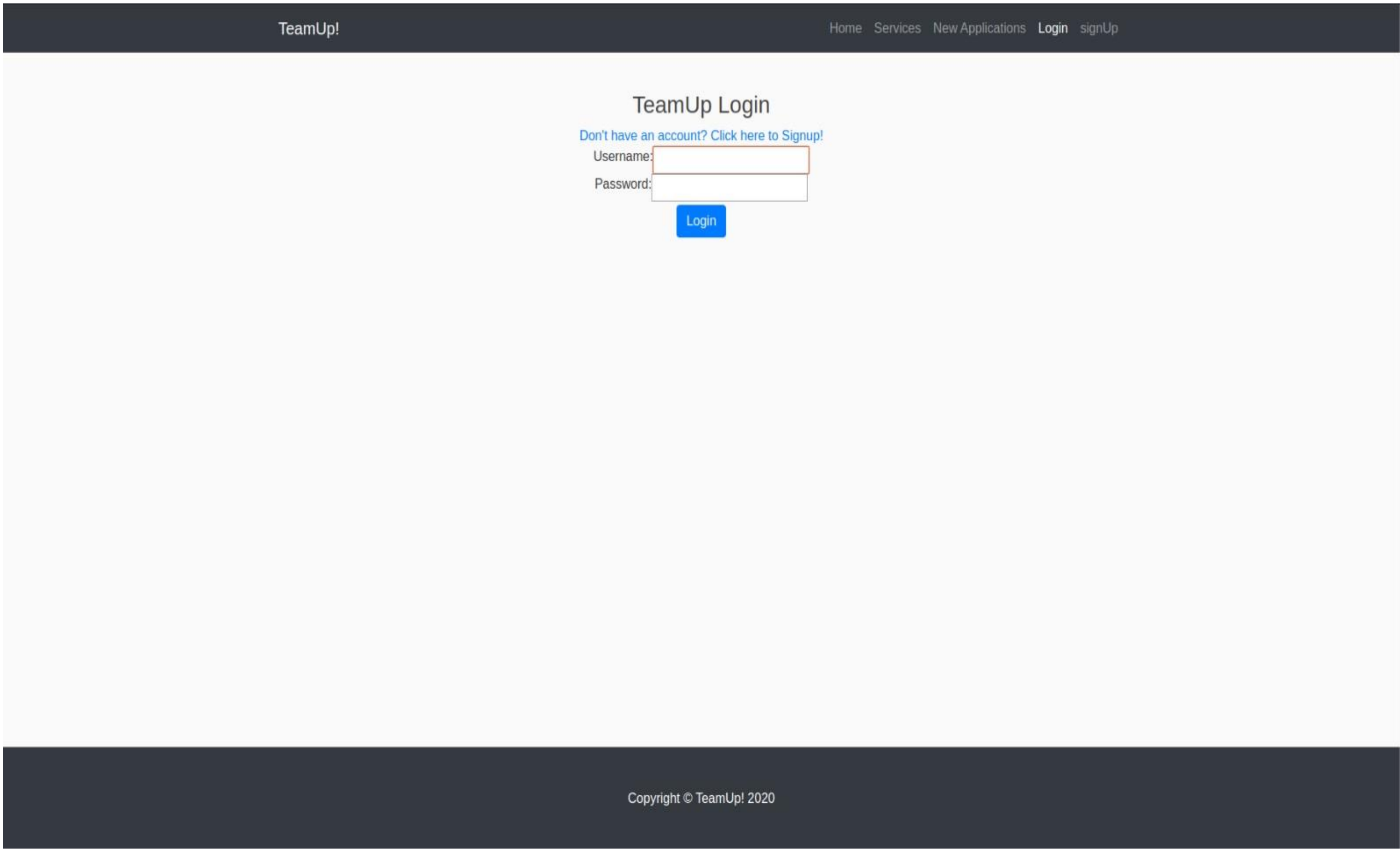
Group's purpose:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed dictum eros blandit sem dignissim porttitor. Vivamus maximus augue sit amet faucibus mattis. Duis cursus ultricies ornare. Maecenas sit amet tristique risus. In ut suscipit lectus. Vivamus vitae accumsan mauris. Mauris maximus, augue rhoncus mattis elementum, enim eros lobortis libero, sed tristique nulla ipsum ac turpis. Nullam vitae facilisis tellus. Sed aliquet ornare suscipit. Morbi rutrum risus vel erat placerat, eget laoreet arcu dapibus. Curabitur non dignissim augue. Praesent laoreet justo felis, nec pellentesque risus porttitor eget. Donec nec felis tincidunt, aliquet libero non, gravida massa.

Copyright © TeamUp! 2020

On the group page, a user can see his/her group members, start a vote, leave the group and see a brief description of the group's purpose.

Login page



TeamUp Login

Don't have an account? Click here to Signup!

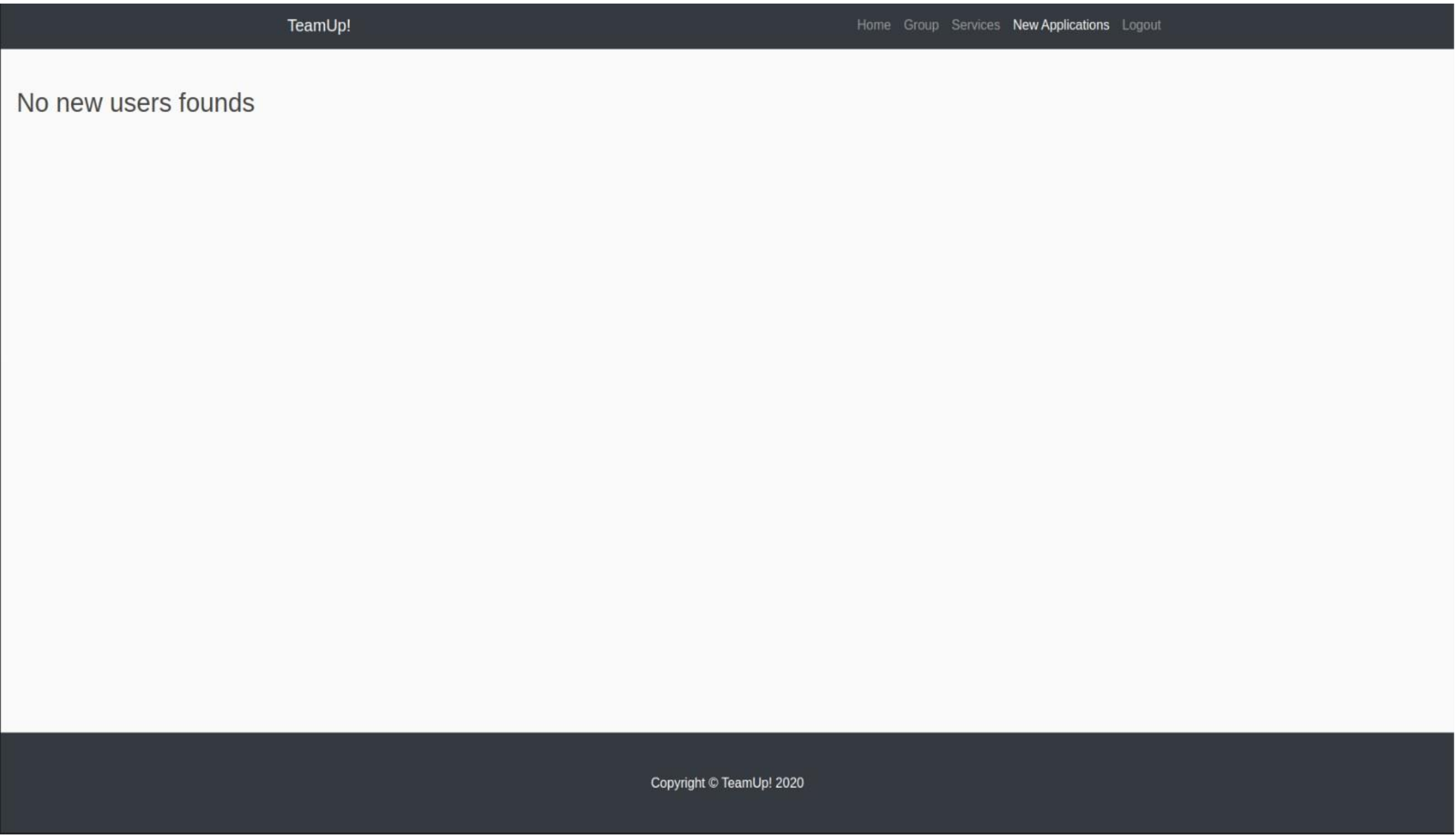
Username:

Password:

Copyright © TeamUp! 2020

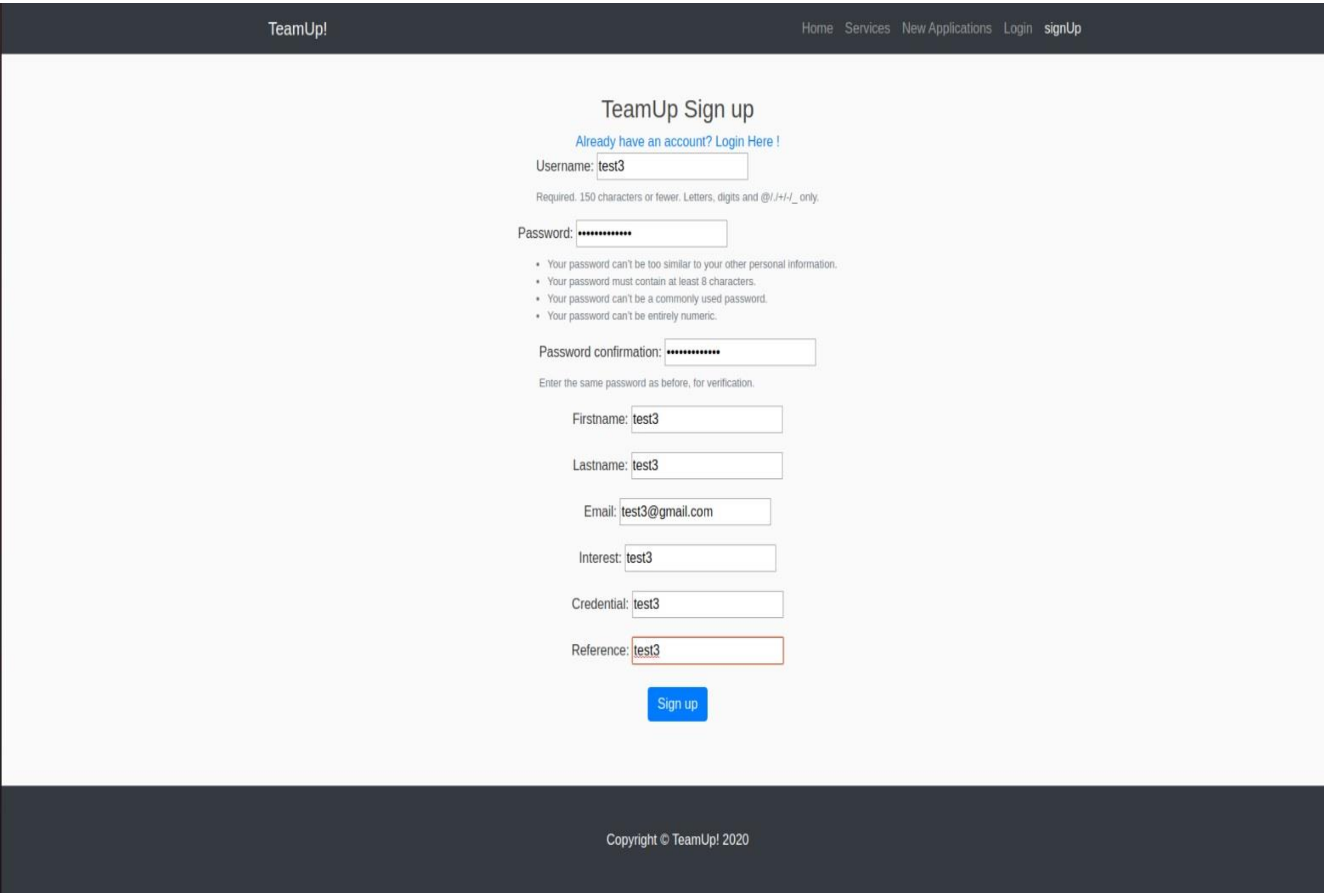
From the login page we can either enter our credential or click on the "Don't have account?" link to be taken to the sign-up page

New applications page: (Show case of accepting new user functionality)



No new users founds

Here no new user has registered on our app yet. Let’s do a test signup



TeamUp Sign up

Already have an account? Login Here !

Username: test3

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Password: •••••••••••••

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation: •••••••••••••

Enter the same password as before, for verification.

Firstname: test3

Lastname: test3

Email: test3@gmail.com

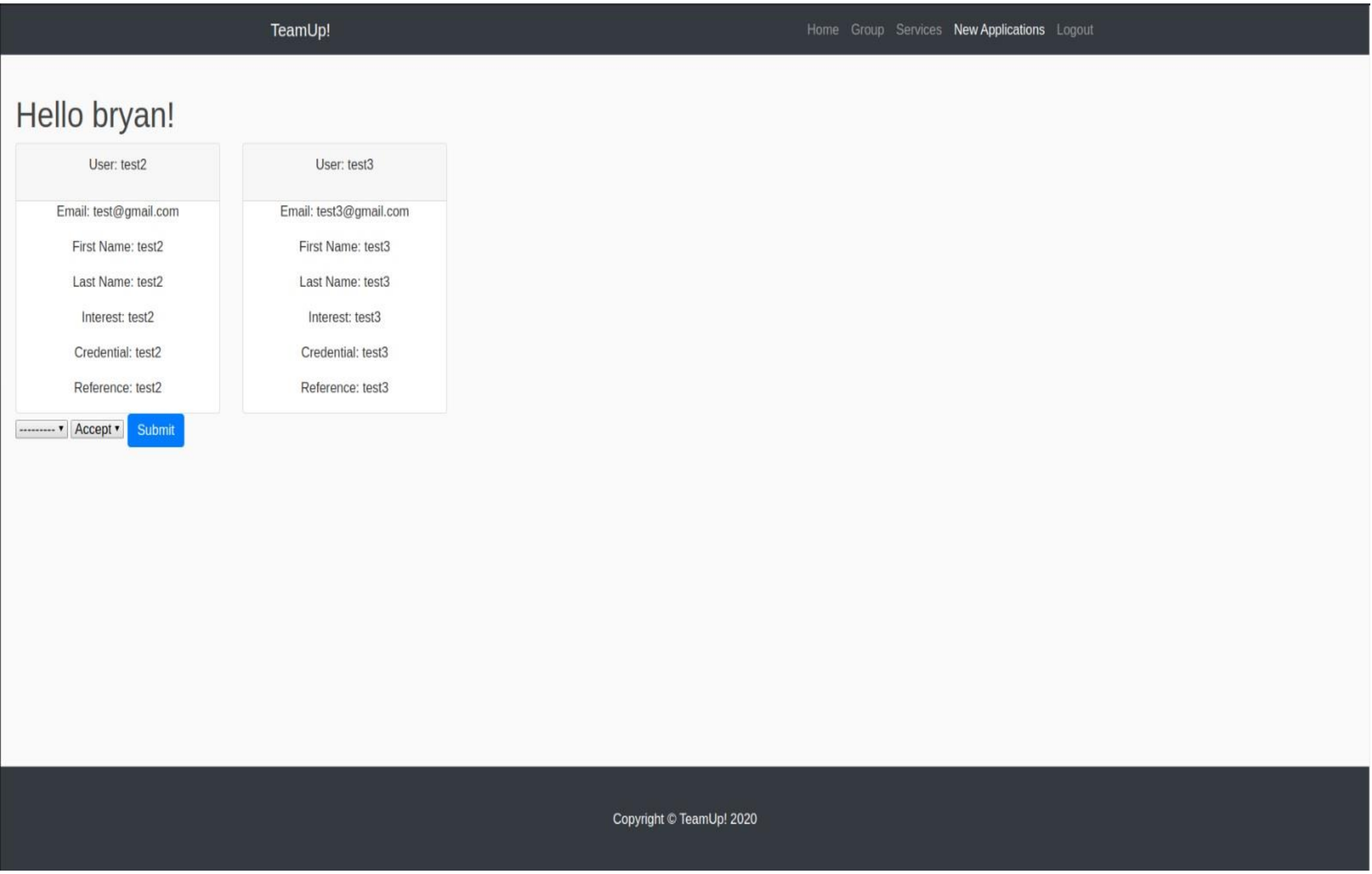
Interest: test3

Credential: test3

Reference: test3

Copyright © TeamUp! 2020

Here we can see one of the two test signups that we performed to test our functionality



Hello bryan!

User: test2

Email: test@gmail.com

First Name: test2

Last Name: test2

Interest: test2

Credential: test2

Reference: test2

User: test3

Email: test3@gmail.com

First Name: test3

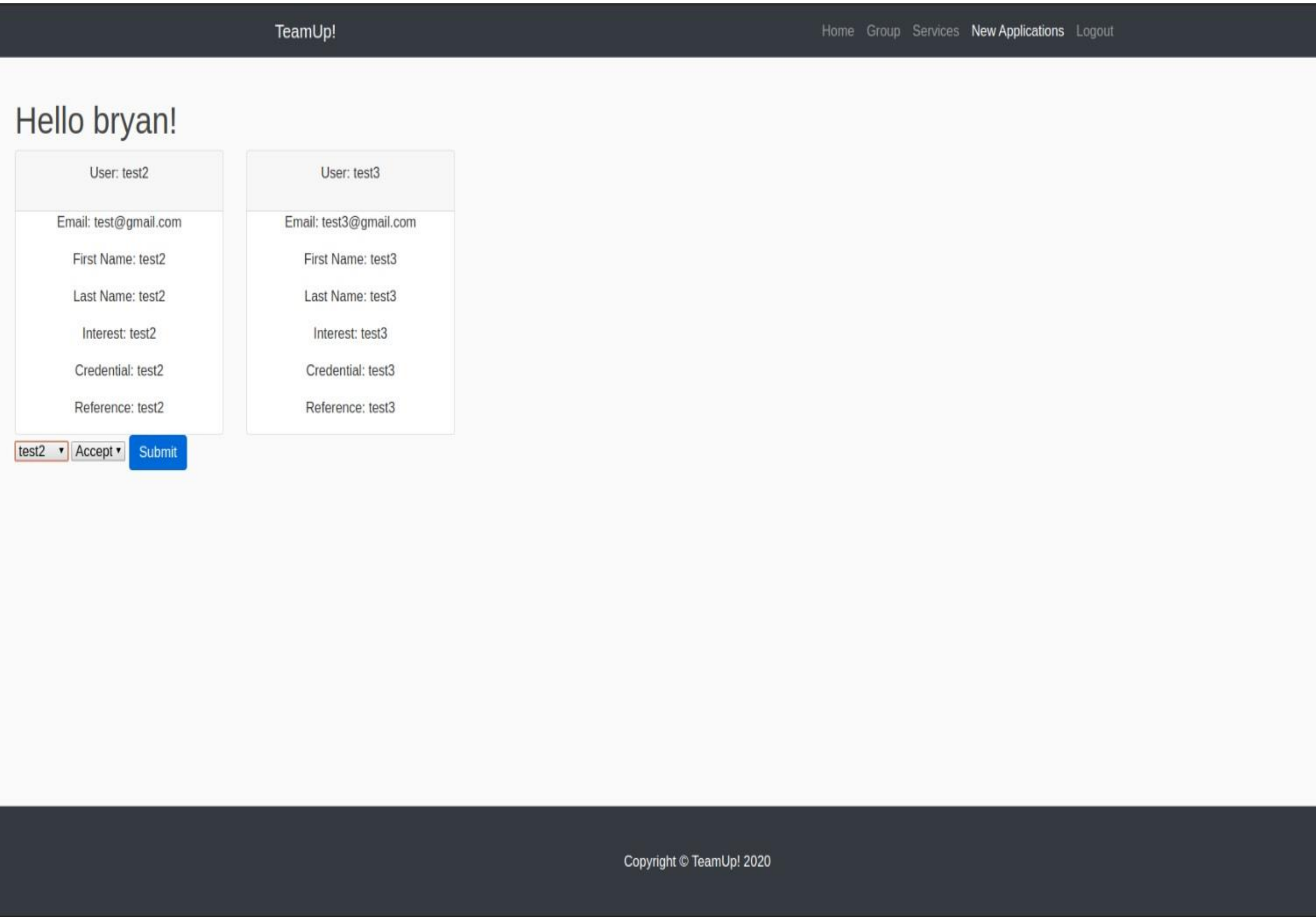
Last Name: test3

Interest: test3

Credential: test3

Reference: test3

Here we can see the test users with their information



Hello bryan!

User: test2

Email: test@gmail.com

First Name: test2

Last Name: test2

Interest: test2

Credential: test2

Reference: test2

User: test3

Email: test3@gmail.com

First Name: test3

Last Name: test3

Interest: test3

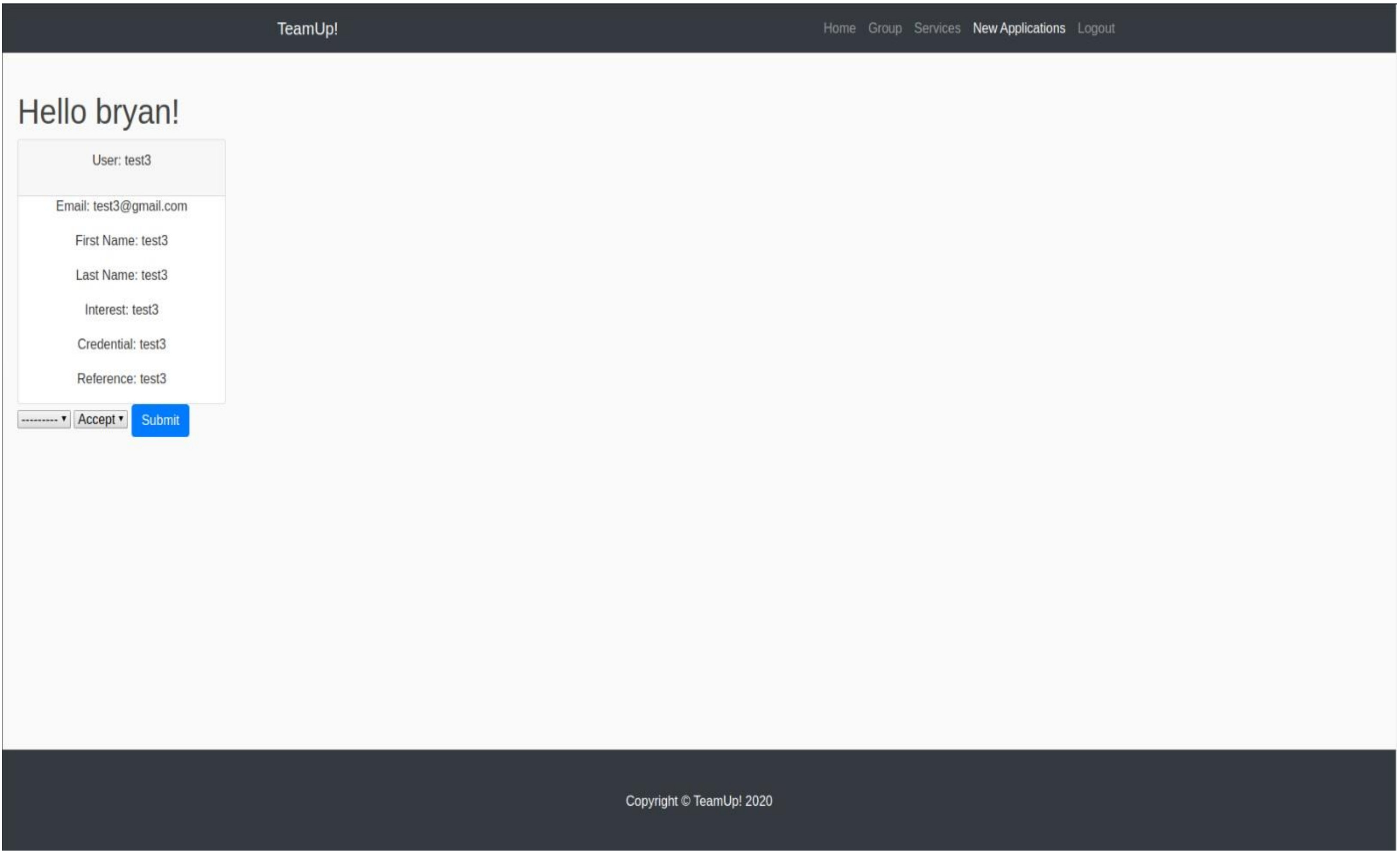
Credential: test3

Reference: test3

test2 ▾ Accept ▾

Copyright © TeamUp! 2020

Here we select test2 as the user to be accepted and perform the accepting action



Hello bryan!

User: test3

Email: test3@gmail.com

First Name: test3

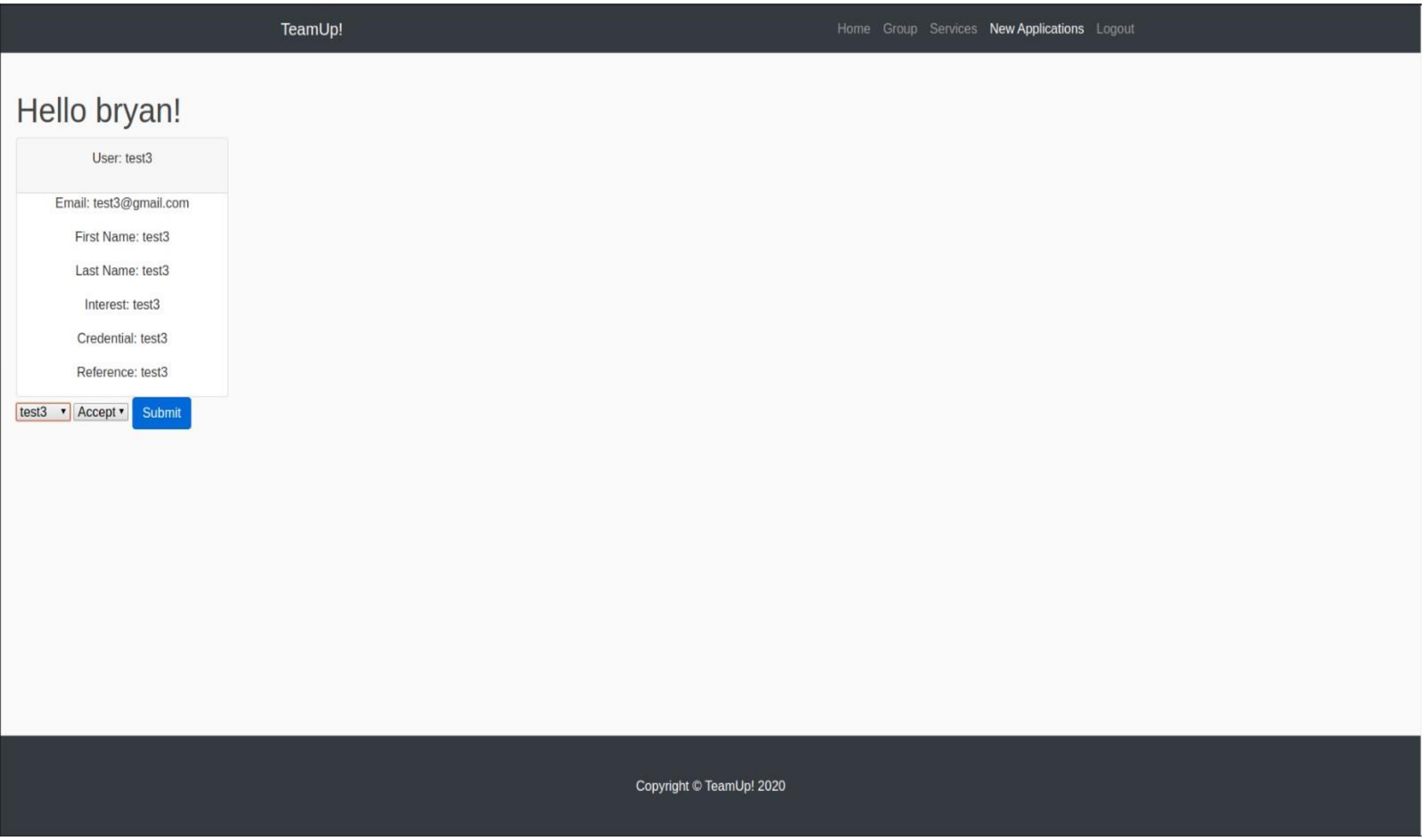
Last Name: test3

Interest: test3

Credential: test3

Reference: test3

The page immediately updates after accepting the user



Hello bryan!

User: test3

Email: test3@gmail.com

First Name: test3

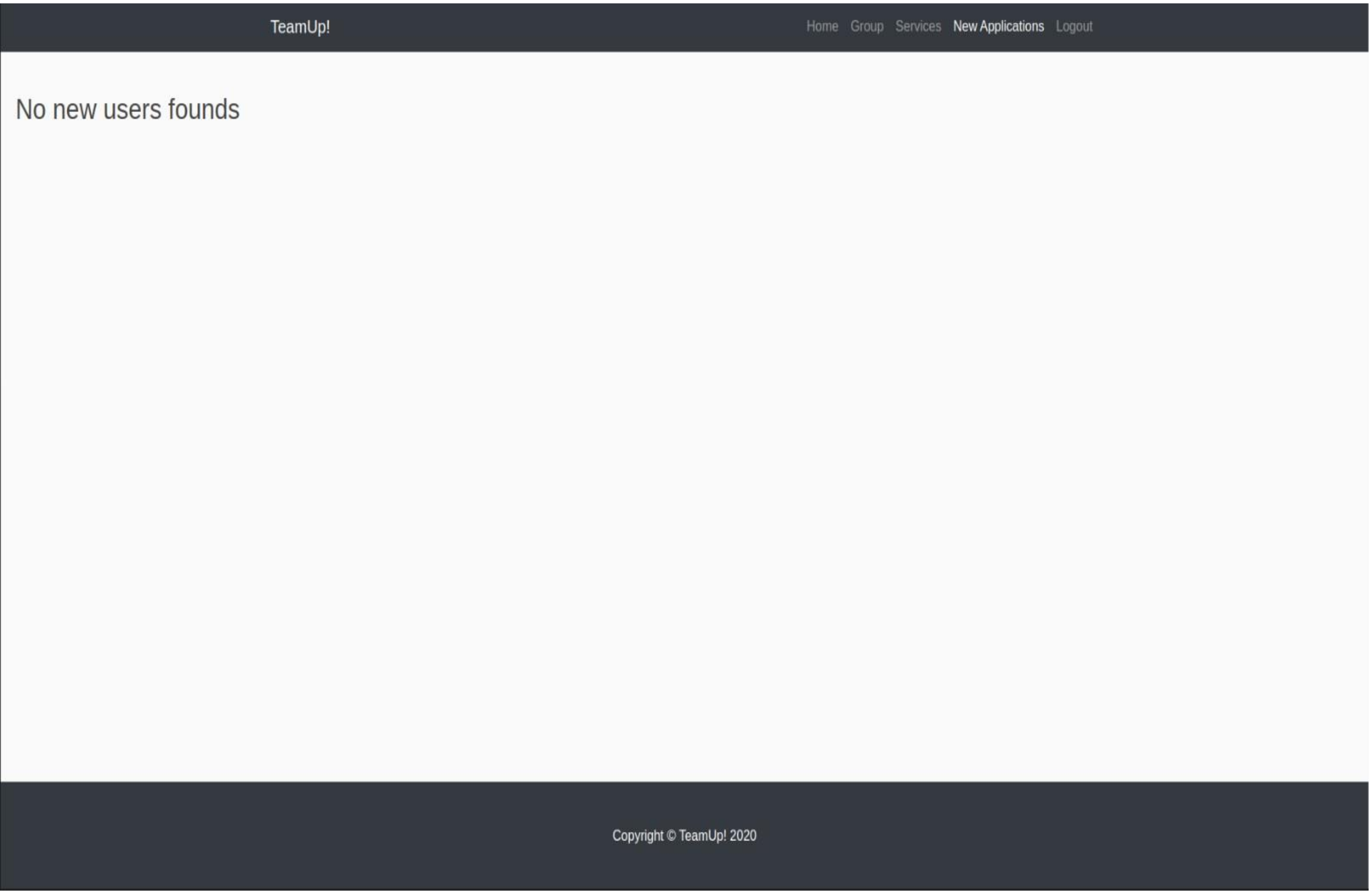
Last Name: test3

Interest: test3

Credential: test3

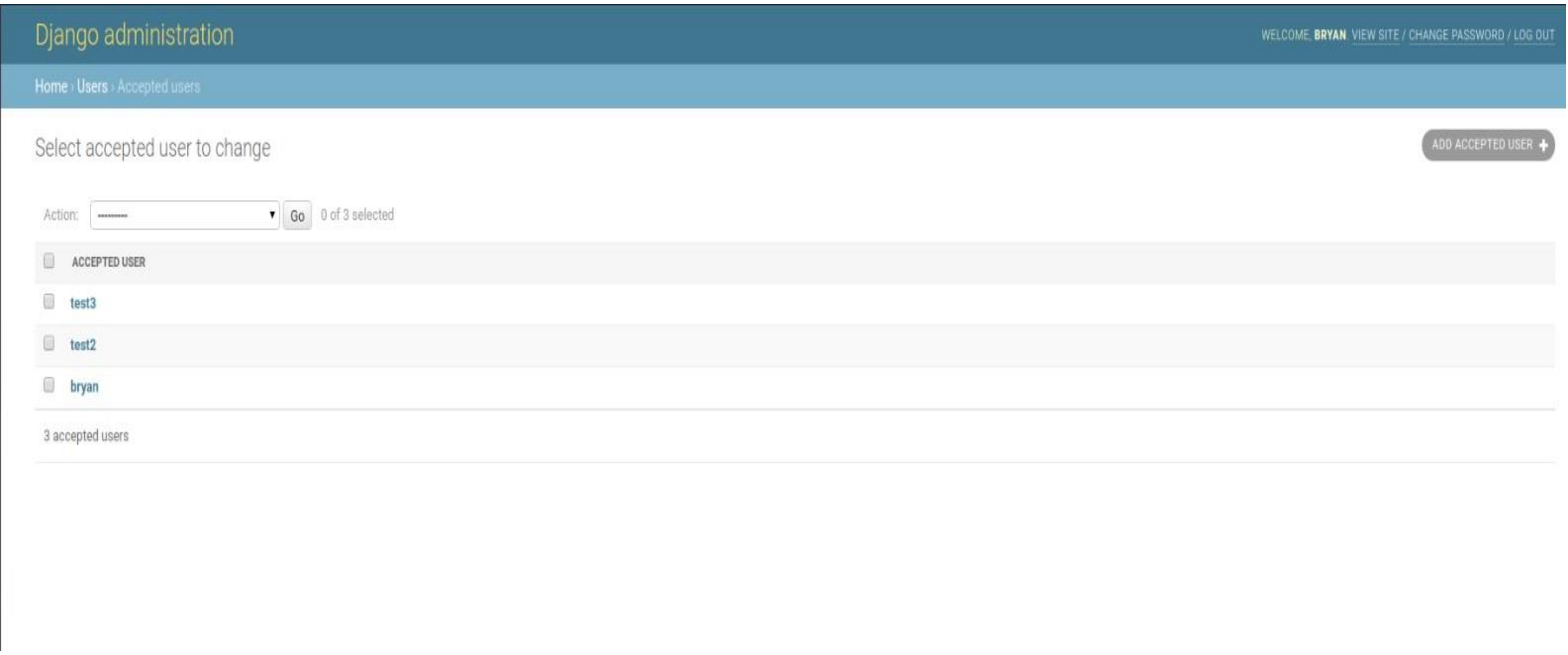
Reference: test3

Here we select to accept test user 3 and submit it



No new users founds

Here we can see the app immediately updating after accepting the last user


Django administration
WELCOME, BRYAN. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Users › Accepted users
ADD ACCEPTED USER +
Action:

Go
0 of 3 selected

Select accepted user to change

□	ACCEPTED USER
□	test3
□	test2
□	bryan

3 accepted users

Here we can see the changes reflected in our Database on the accepted users table

6. Minutes of Group meetings.

In-person Meeting #	**Discussion**
1	Discussed possible approaches, frameworks and overall tech stack to use while developing the app
2	Laid out most of the app functionalities with and designed most of the database models to keep track of how we would implement the functionalities

Online Meeting#	**Discussion**
1	Discussed GUI ideas for the web application and how the users would navigate through the app
2	Finished most of the GUI for the web application and finished the signup, login, and accepting user functionality

7. Github

Github Repo Link: https://github.com/edgarp2017/SEproject