

# **Team Up**

## **Software Design Report**

Version 2.0

### **Team Members:**

**Team A**

Sajad Gholamzadehrizi

Arun Ajay

Abdul Imtiaz

Ahsan Fayyaz

Team Up	Version: 2.0
Software Design Report	Date: 05/01/2020

## Revision History

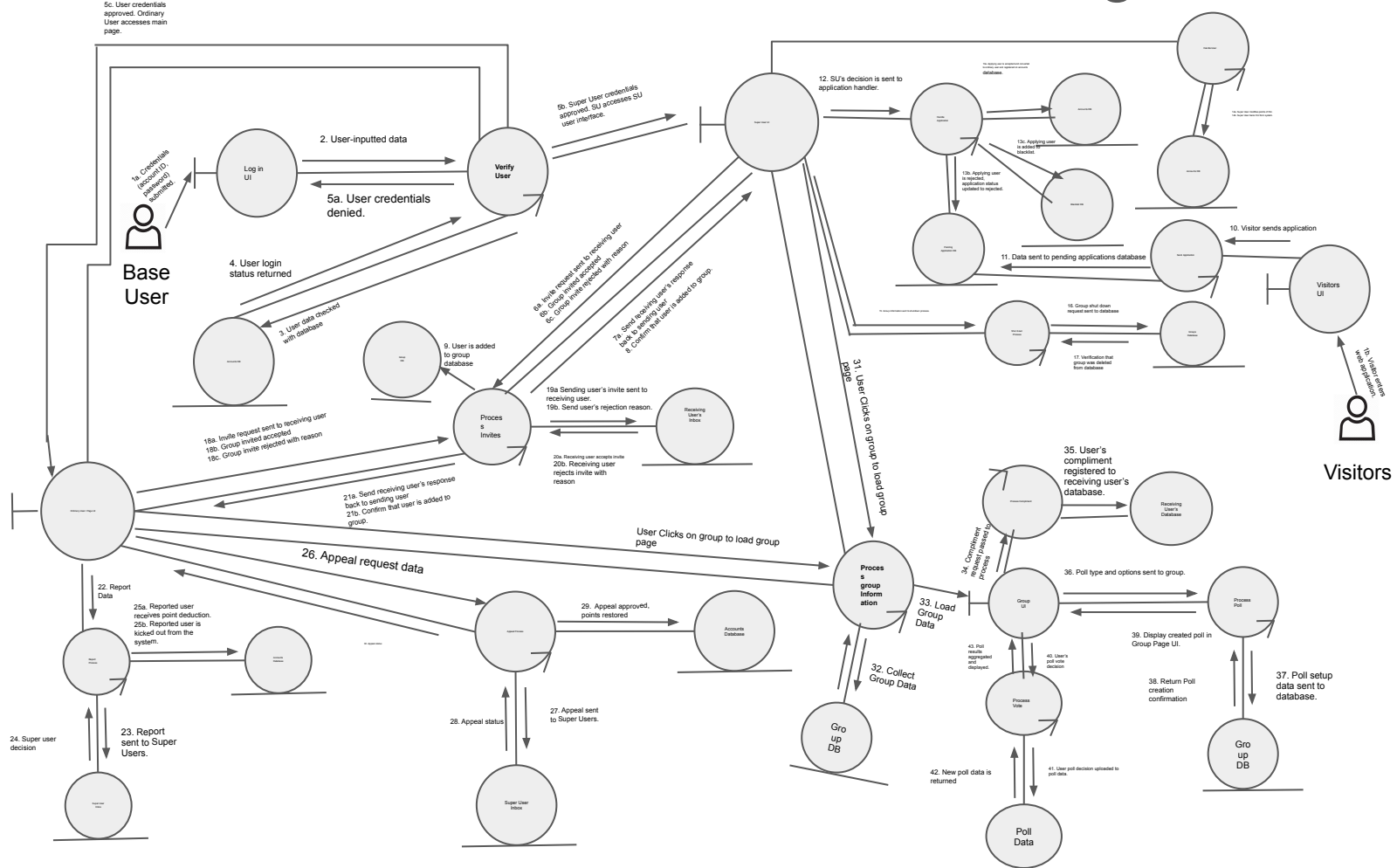
Date	Version	Description	Author
03/23/2020	1.0	Project Specifications	Team A
05/01/2020	2.0	Design Report	Team A

Team Up	Version: 2.0
Software Design Report	Date: 05/01/2020

## Table of Contents

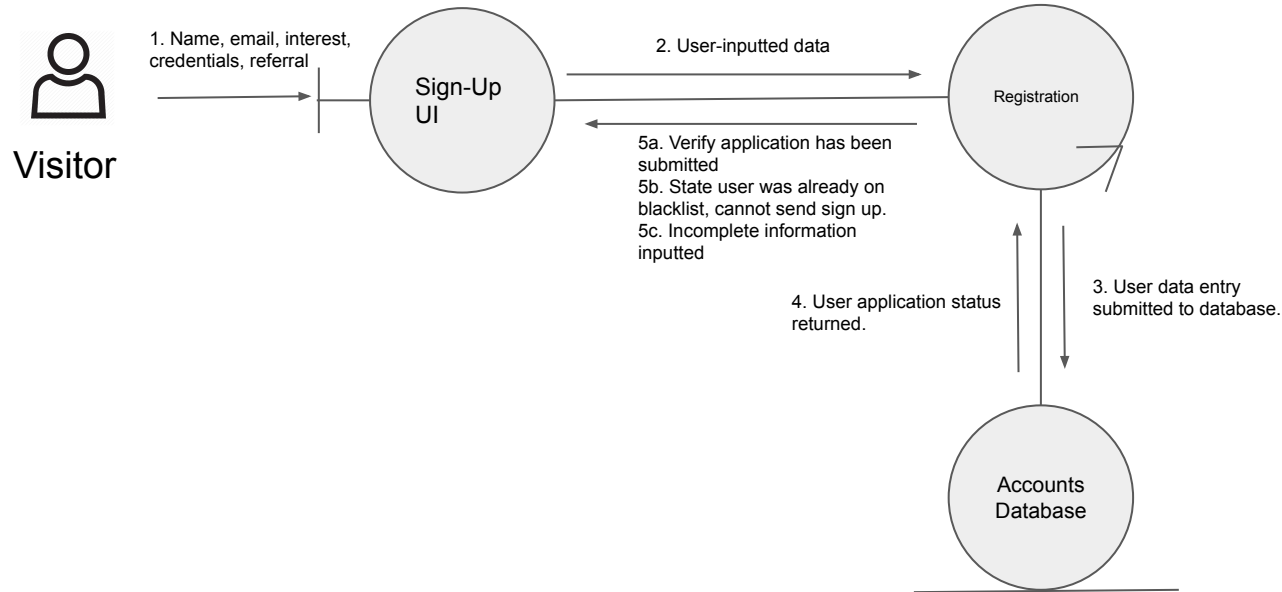
1. Introduction	4
2. Use-Cases Diagrams	5
3. Entity Relationship Diagram	24
4. Detailed Design	25
5. User Interface Screens	34
6. Meeting Minutes	46
7. GitHub Repository	46

# Overall Class Collaboration Diagram



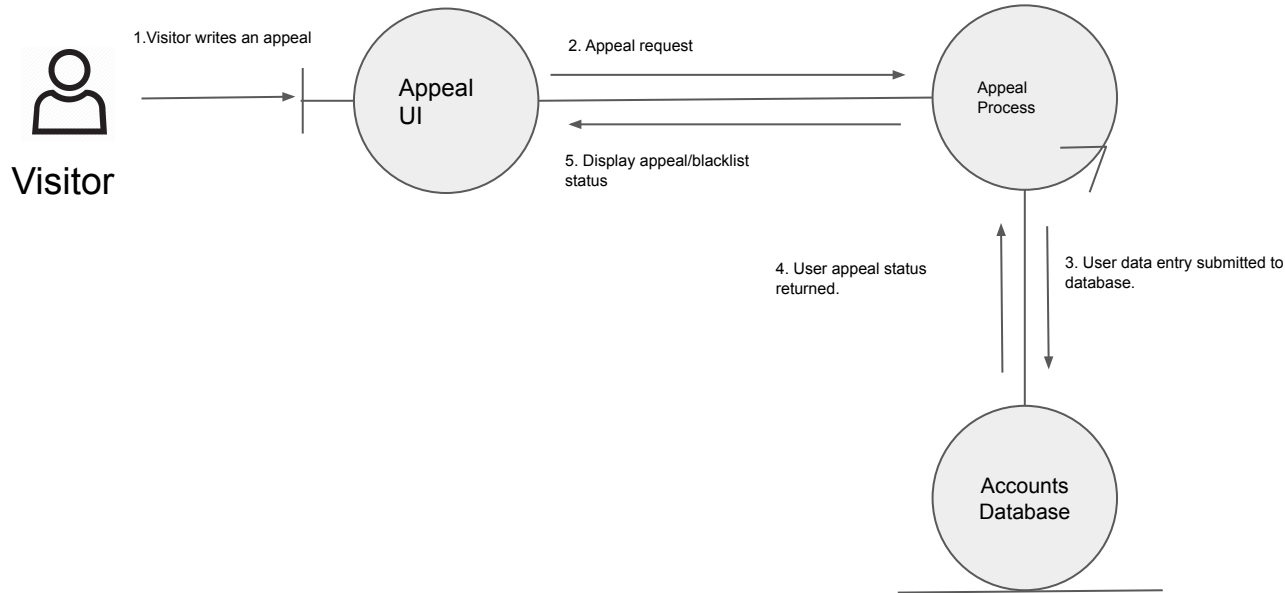
# Use Case: *Sign Up*

Visitors have the ability to request membership on Team Up. They must provide their name, email, interests/hobbies and the member who referred them (OU or VIP). If the user is in the blacklist, signup cannot proceed. The user must provide all information, or the application will not submit successfully.



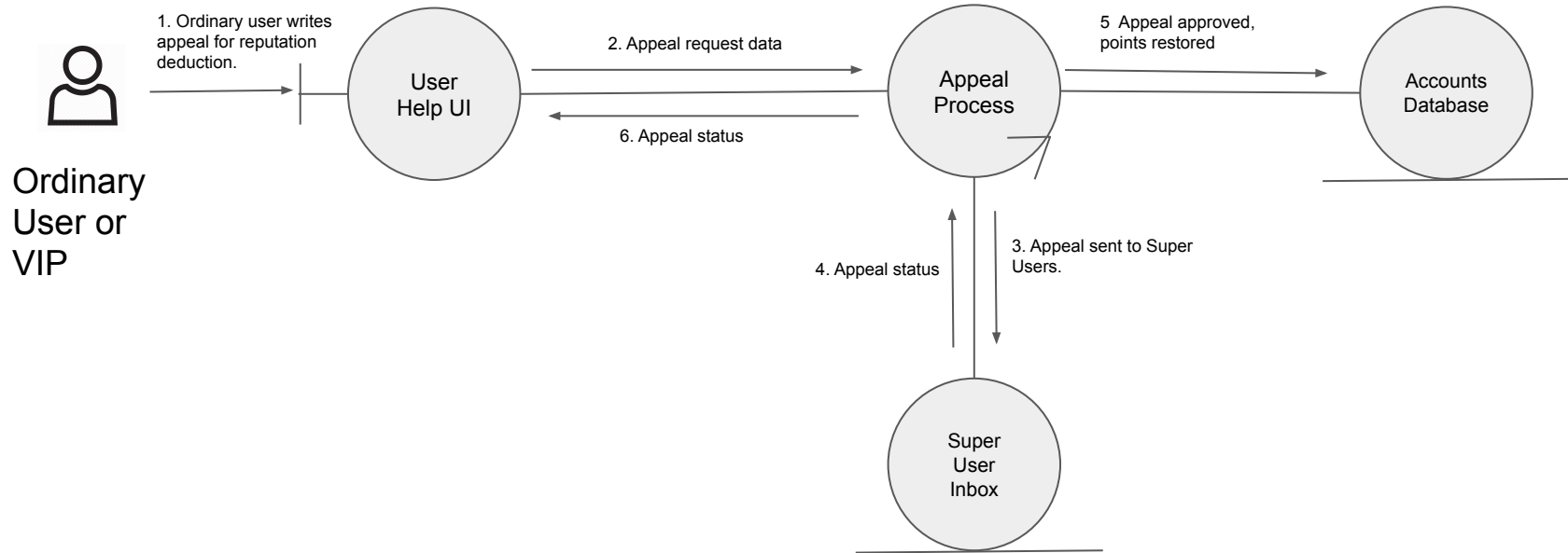
# Use Case: *Appeal Sign Up*

Visitors have the ability to appeal their application if it was rejected. However, if they are rejected again, they will be added to a blacklist and cannot utilize the Sign Up feature again.



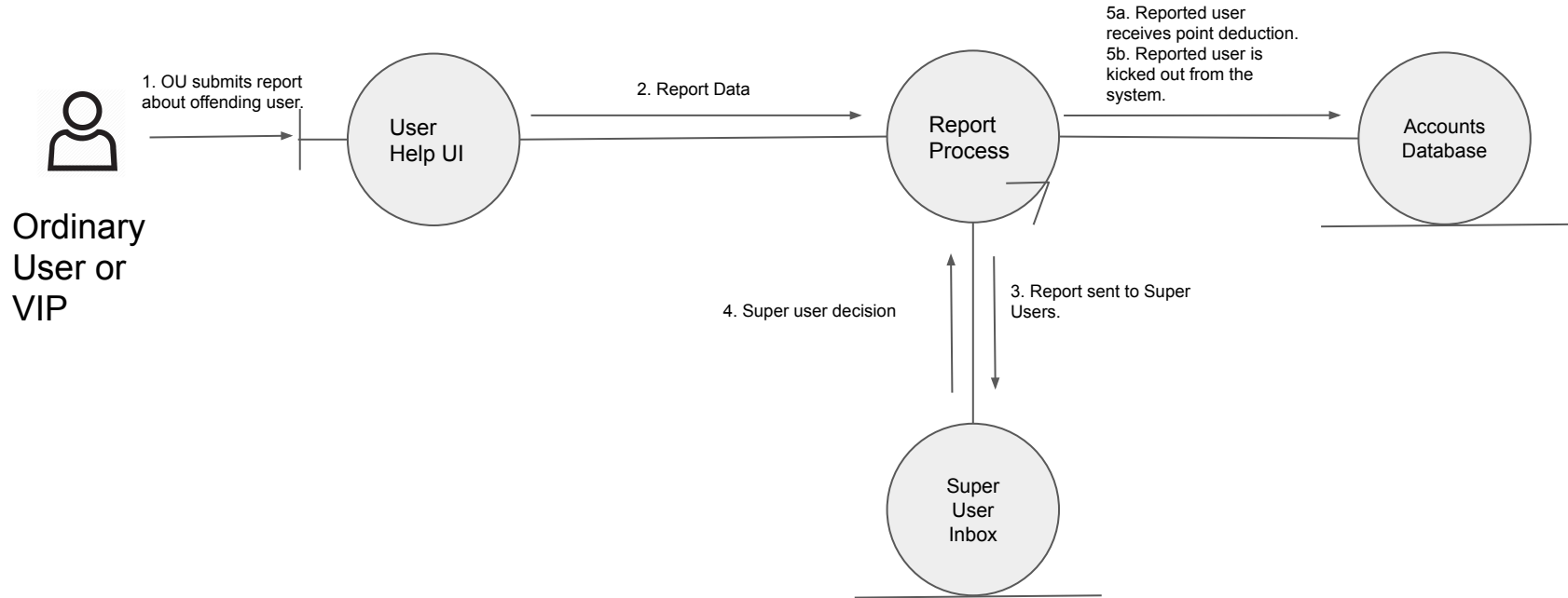
# Use Case: *Appeal Reputation Deduction*

Ordinary Users or VIP have the ability to appeal a point reduction issued by a Super User if they were kicked from a group. Their appeal may be denied, in which case the OU's points will not change, or approved, in which case the point reduction will be reversed.



# Use Case: *Report User*

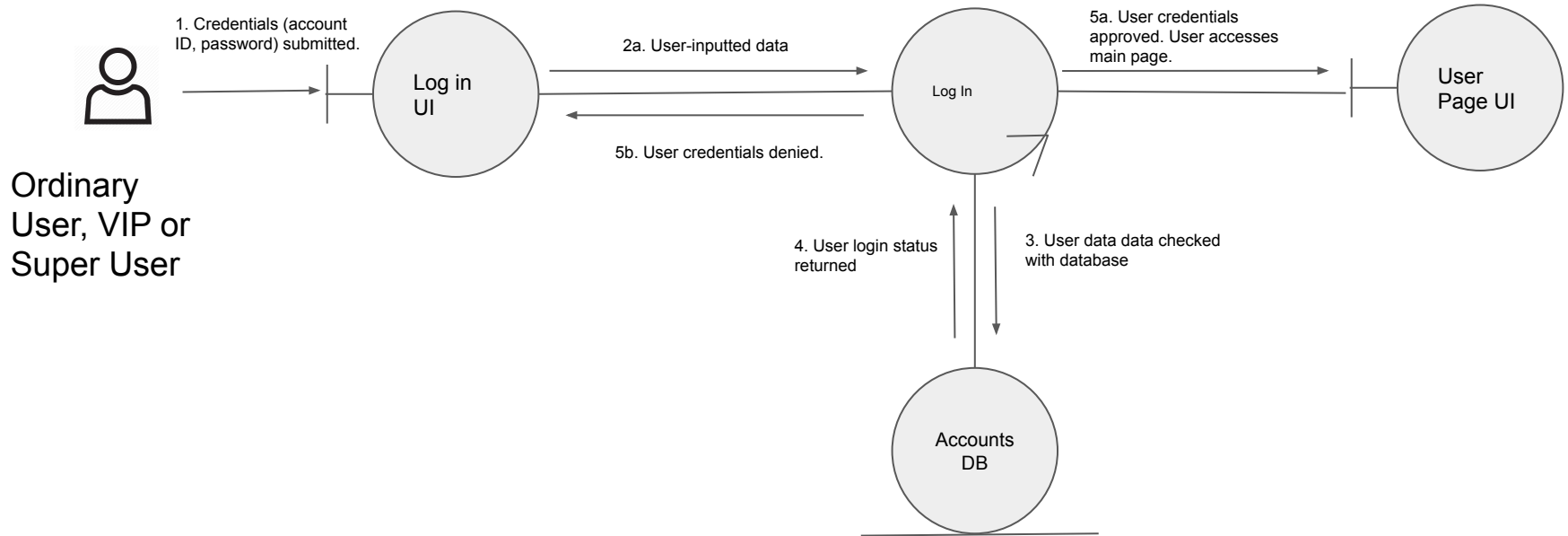
Ordinary Users, and VIP users have the ability to report a user. Super User can respond to Reports by issuing point deductions or kicking out OUs involved from the system.





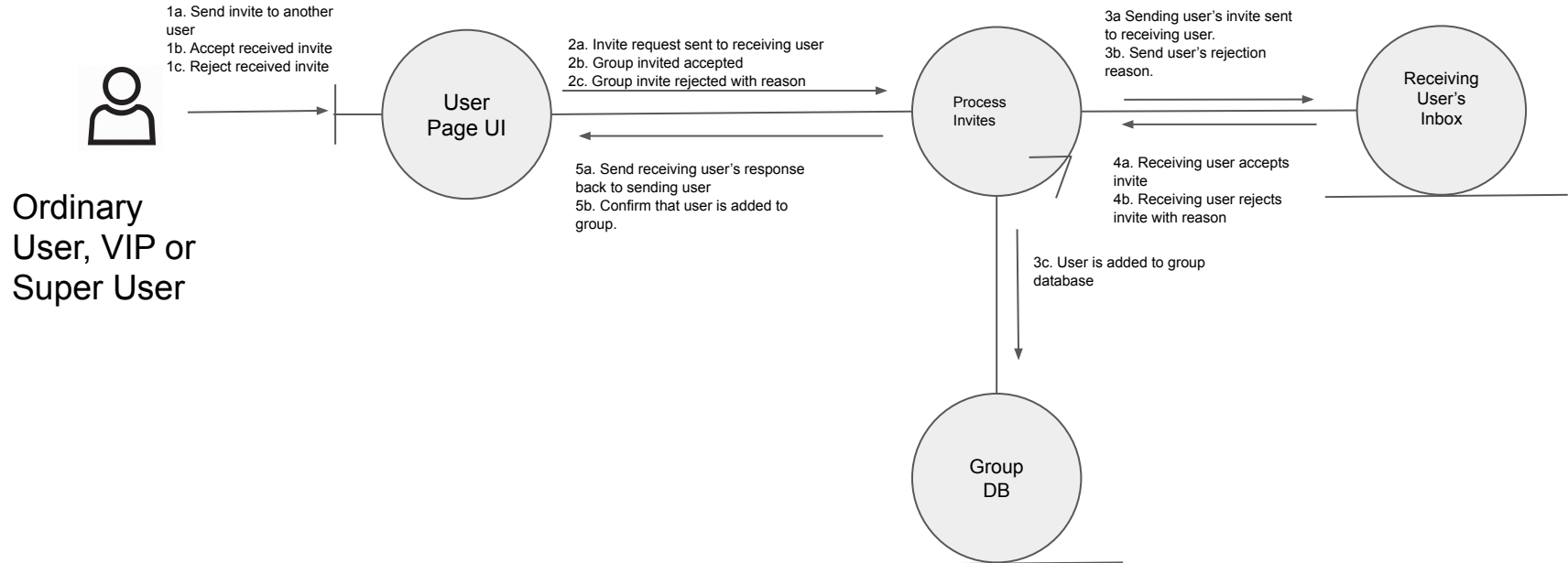
# Use Case: *User Login*

Ordinary Users, VIPs, and Super Users must log in with their credentials to access the User UI. If the user provides invalid credentials, the user will be notified and will not be permitted to access the User UI.



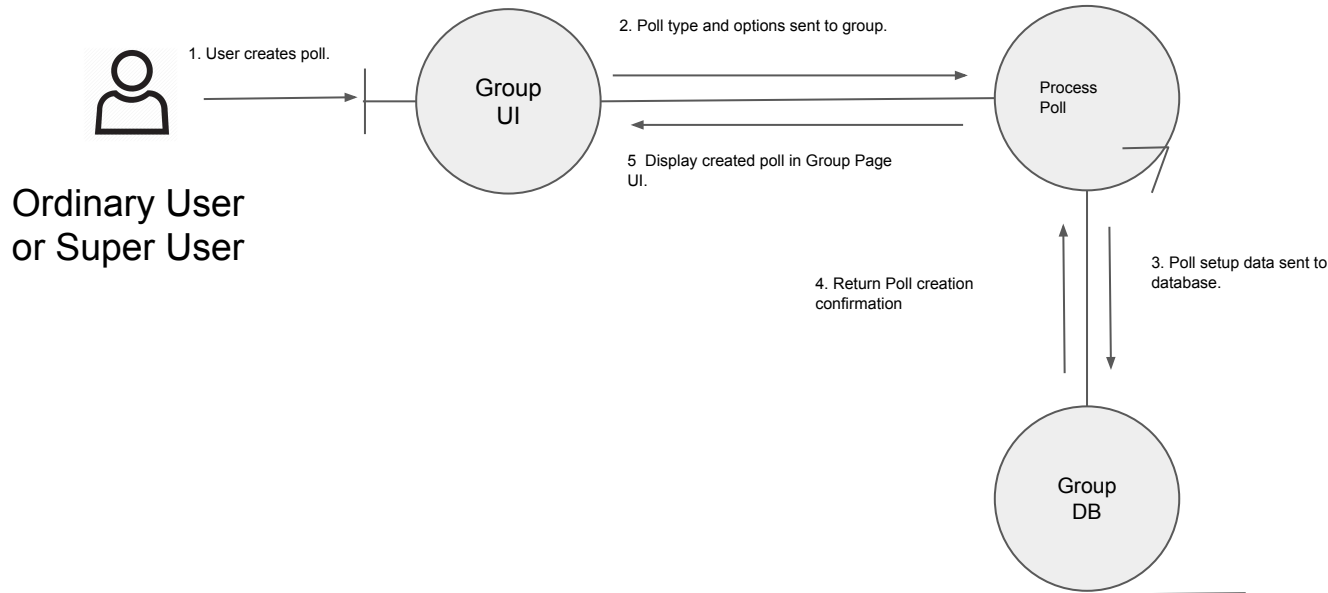
# Use Case: *Invite to Group/Handle Group Invite*

Ordinary Users, VIPs and Super Users have the ability to invite another user to a group. The invitations is sent to the receiving user's inbox. They receiving user is able to accept or reject the invitation. If they accept the invitation, they will be added to the group's database and the user who sent the invitation will receive a confirmation that they person they invited joined the group. If they reject the invitation, then they must provide a reason as to why they rejected the invitation. The sending user will also receive a response so this decision



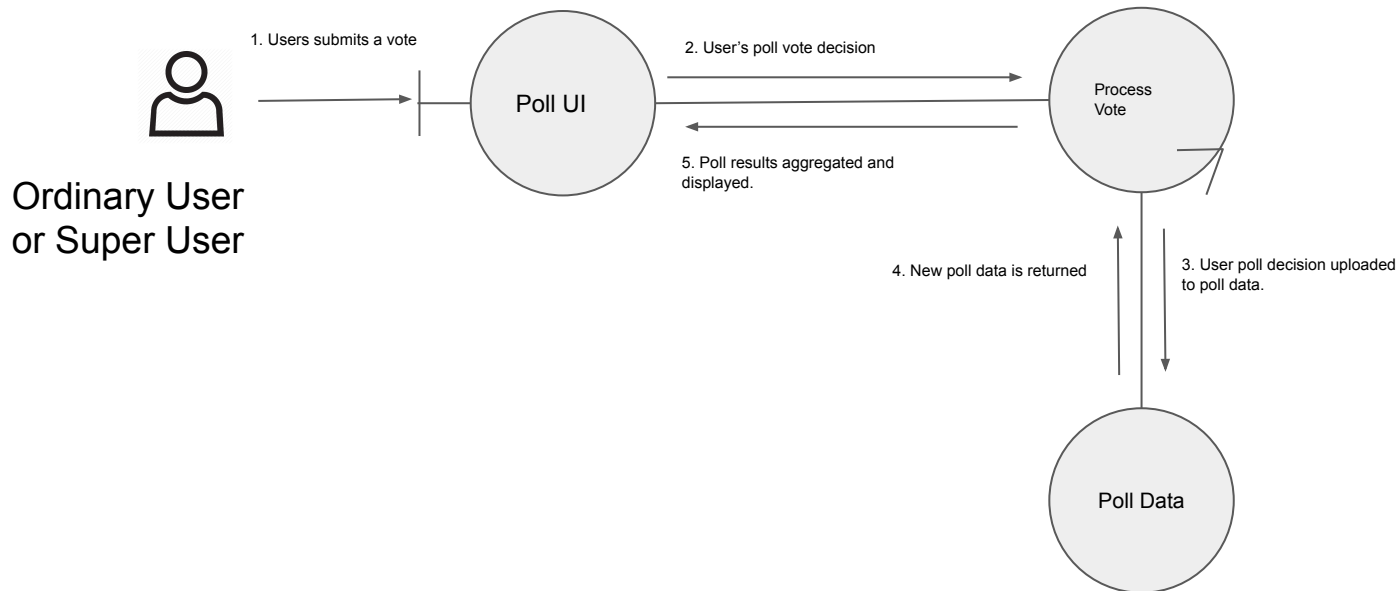
# Use Case: *Create a Meetup/Warning/Praise/KickOut/Close Group Poll*

Ordinary Users, VIPs, and Super Users have the ability to create a poll for meet-up times. The time slot with most votes gets selected. They have the ability to create a poll that determines whether an Ordinary User or VIP deserves a warning. This poll must be unanimous in order to extend a warning. They have the ability to create a poll that determines whether an Ordinary User or VIP deserves praise. This poll must be unanimous in order to extend a praise. They have the ability to create a poll that determines whether an Ordinary User or VIP deserves to be kicked out. This poll must be unanimous in order to kick out the member. They have the ability to create a poll that determines whether a group should be closed. This poll must be unanimous in order to close the group.



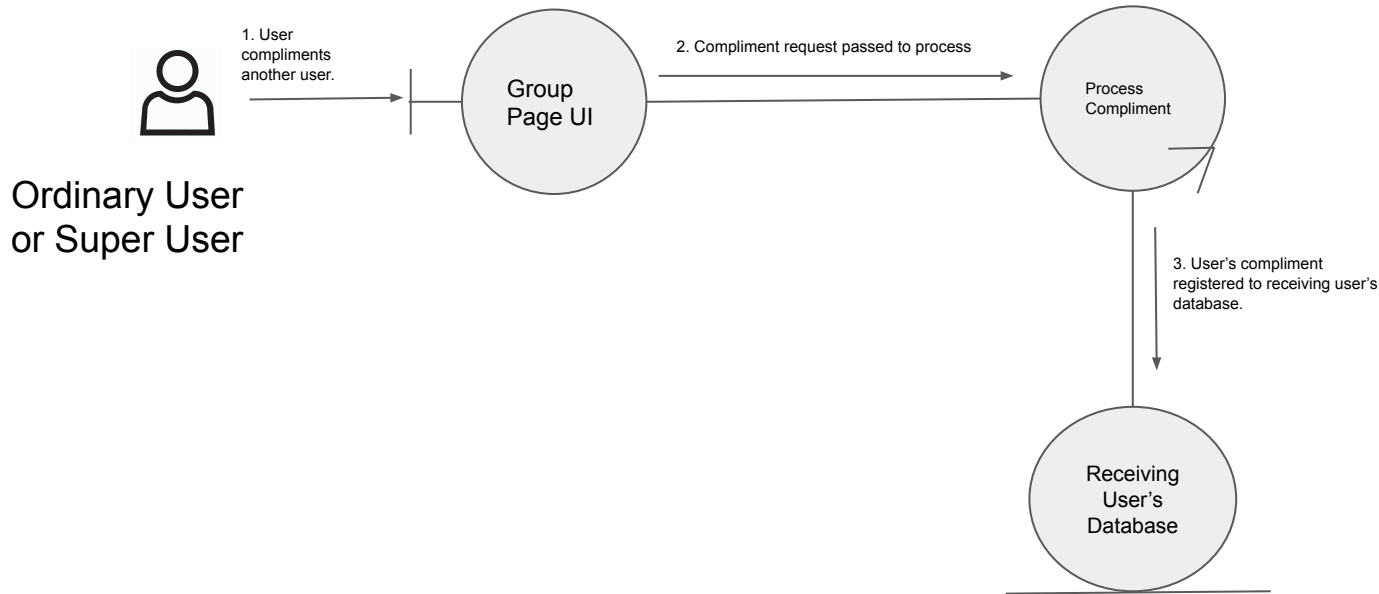
# Use Case: *Issue a Meetup/Warning/ Praise/KickOut/Close Group Poll Vote*

Ordinary Users, VIPs, and Super Users have the ability to extend a meet-up vote for a particular time-slot from a poll. They have the ability to extend a warning vote for a particular member from a poll. They have the ability to extend a praise vote for a particular member from a poll. They have the ability to extend a kick-outvote for a particular member from a poll. They have the ability to extend a kick-out vote for a particular member from a poll. Users must be in the group the poll belongs to in order to vote in it.



# Use Case: *Issue a Compliment*

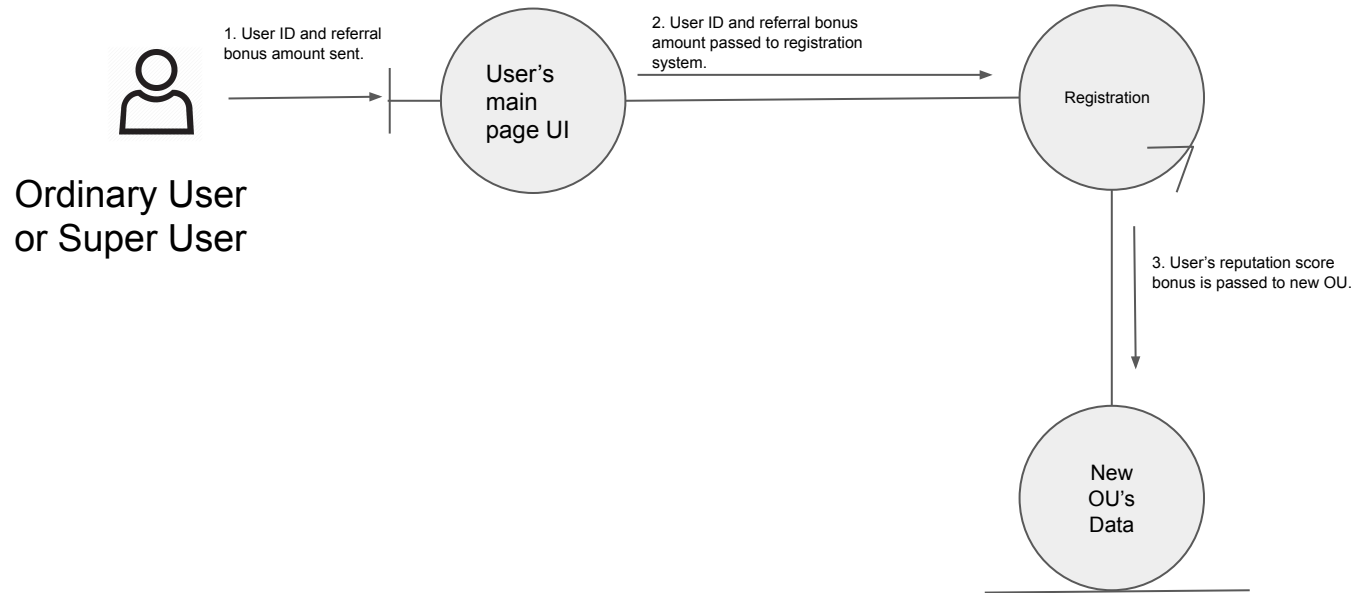
Ordinary Users, VIPs, and Super Users have the ability to extend a compliment for a particular member.



# Use Case: *Give an Initial Reputation Score*

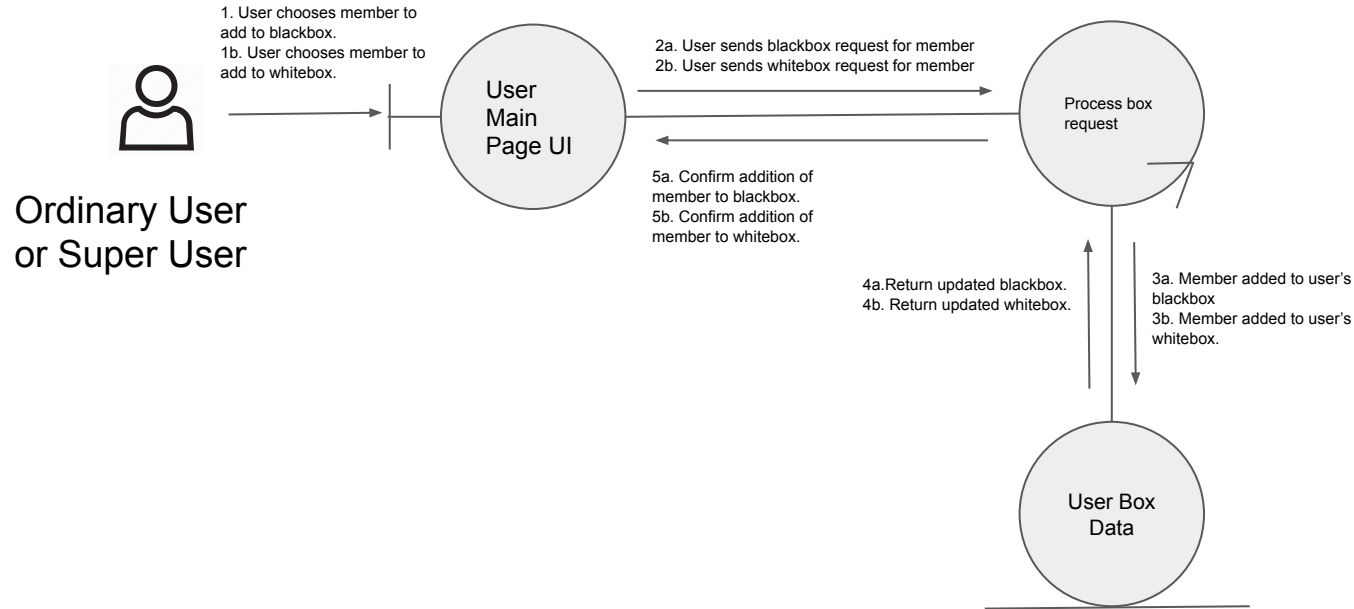
*Give an Initial Reputation Score [OU, VIP, Super User]*

- a. Ordinary Users, VIPs, and Super Users have an ability to give an initial Reputation score for any new OUs who used them as referral.
  - i. OUs can give a score between 0-10
  - ii. VIPs/Super Users can give a score between 0-20



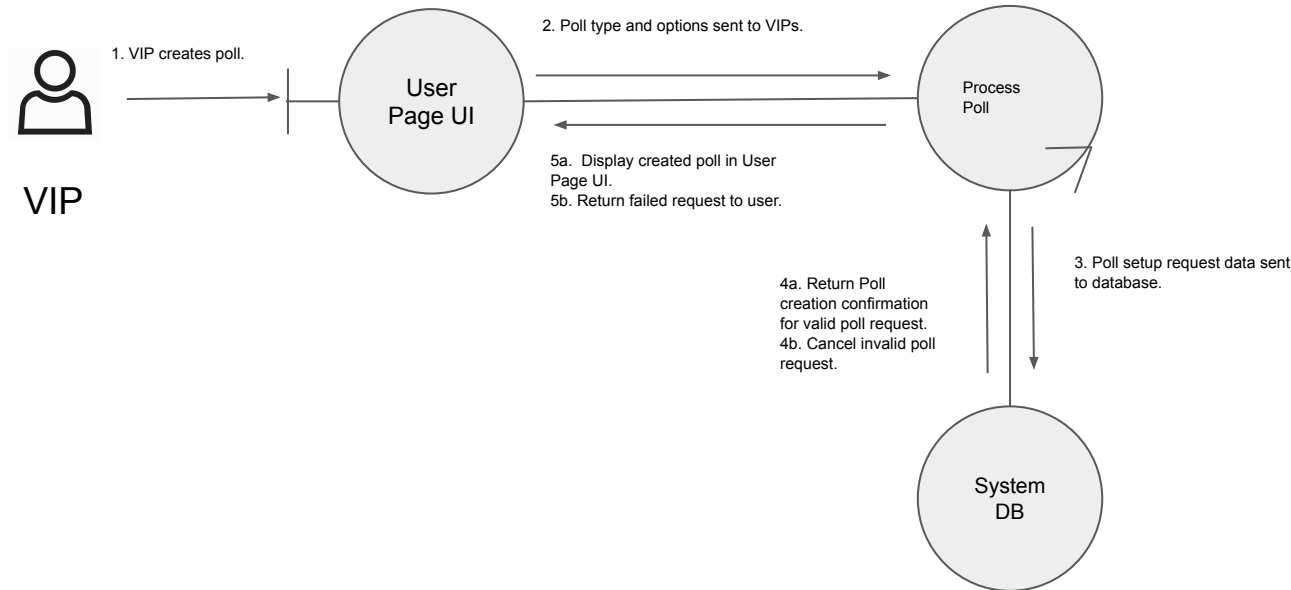
# Use Case: *Add to Blackbox/Whitebox*

Ordinary Users, VIPs, and Super Users have the ability to add a member to their blackbox and whitebox. If a



# Use Case: *Create a Democratic Super User Poll*

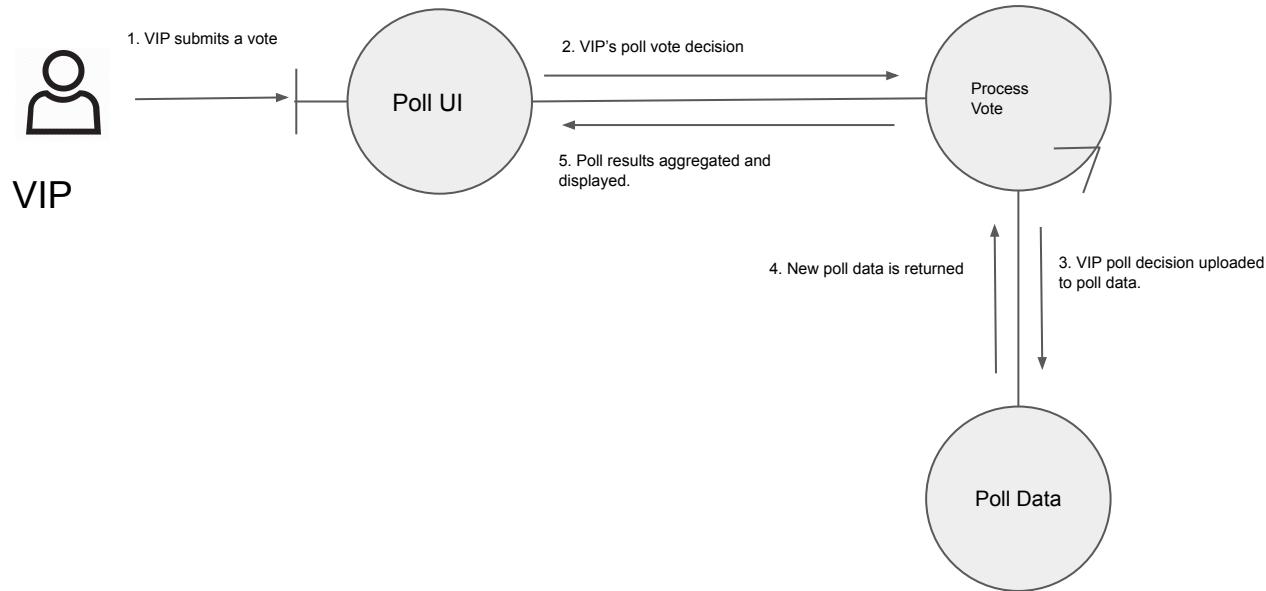
VIPs have the ability to create a Poll in order to vote one VIP as the democratic Super User. There can only be one democratic Super User, so this poll can only be created once. If any VIP tries to initialize a second democratic Super User poll, the attempt will fail and the VIP will be notified.





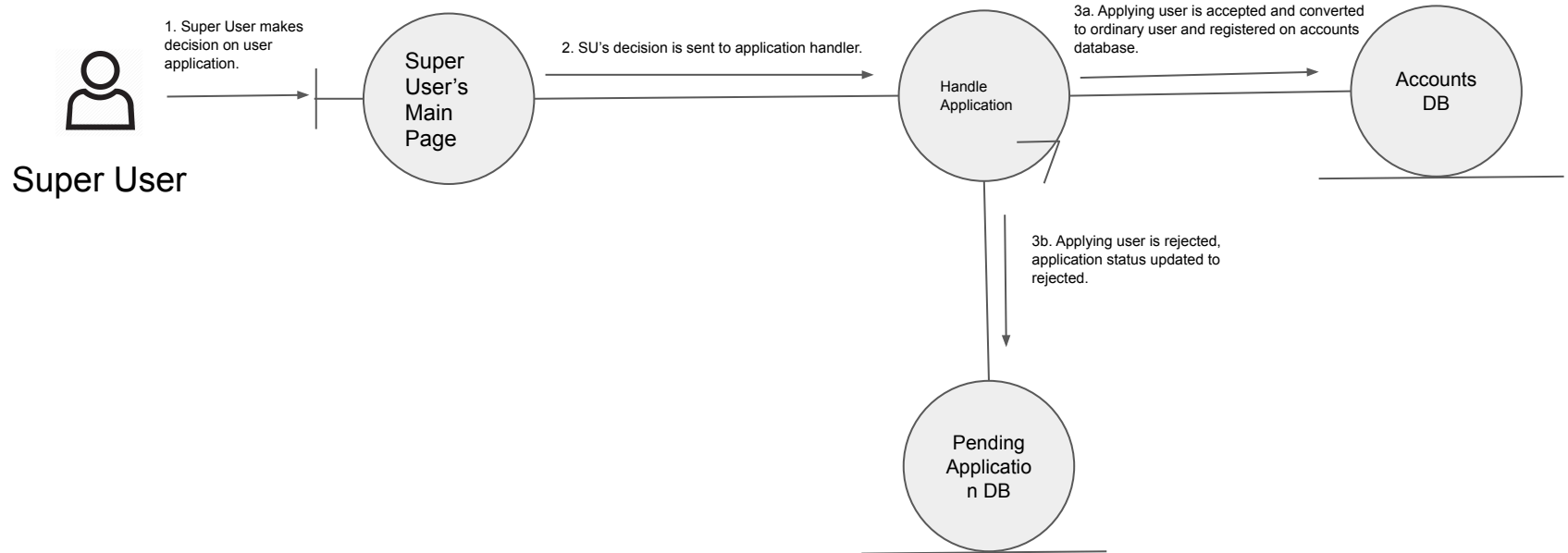
# Use Case: *Issue a Democratic Super User Vote*

VIPs have the ability to vote one VIP as the democratic Super User if a poll exists.



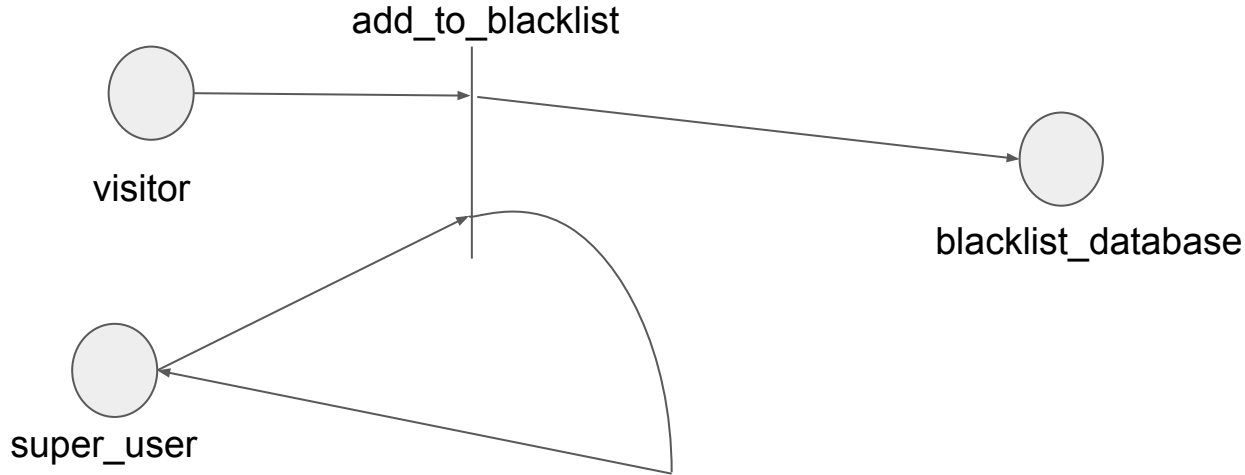
# Use Case: *Handle Signup Application*

Super User has the ability to verify the information of the visitor registering an account. If approved, an email is sent to the user which includes their credentials (account ID and password)



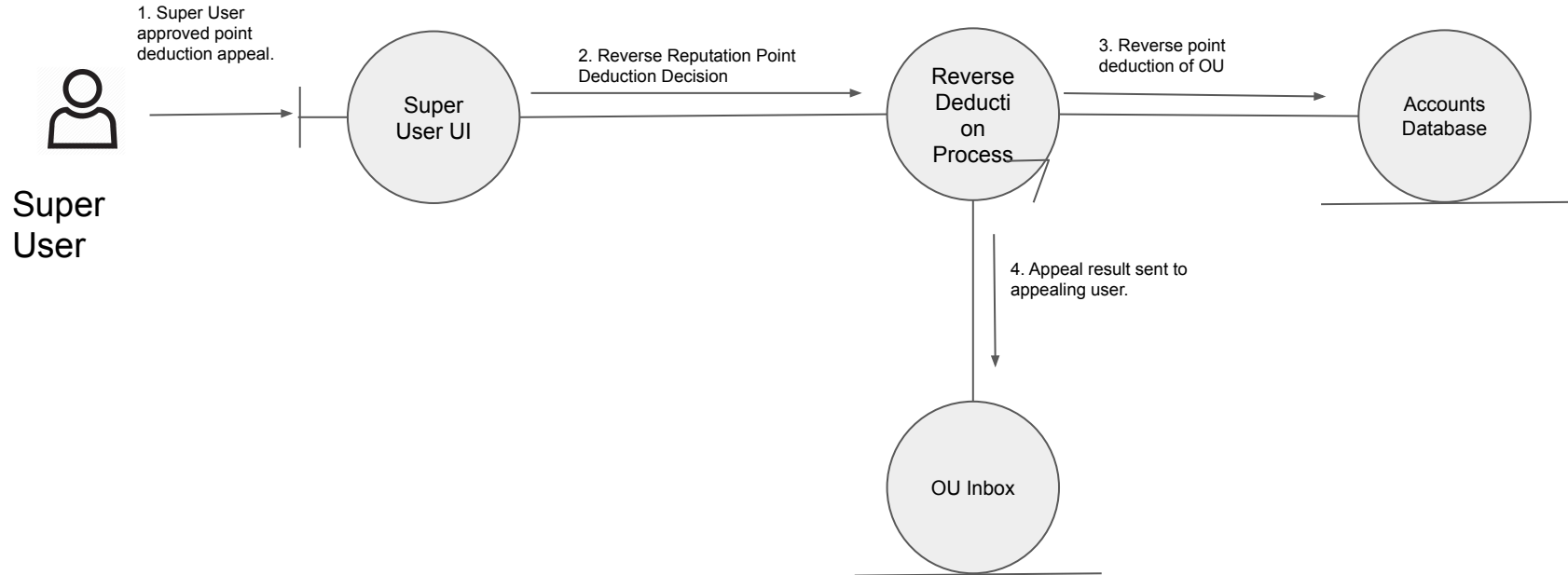
# Use Case: *Add to Server Blacklist*

Super Users have the ability to ban a visitor from Team Up by adding them to the blacklist.



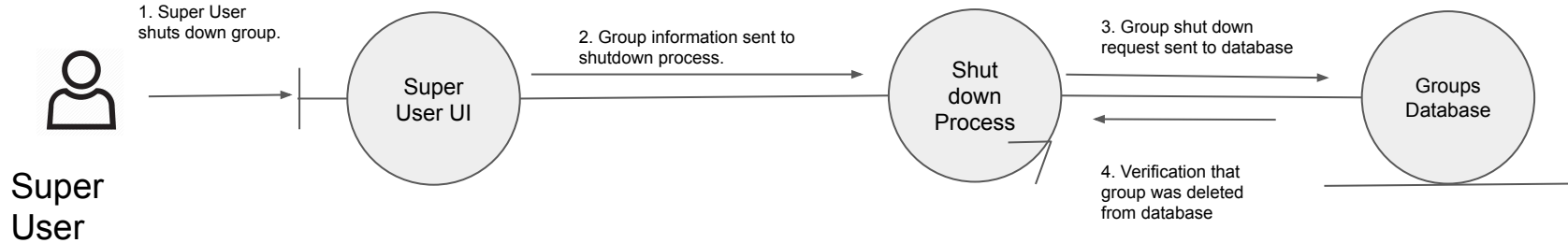
# Use Case: *Reverse Reputation Point Deduction*

Super Users have the ability to reverse a reputation point deduction.



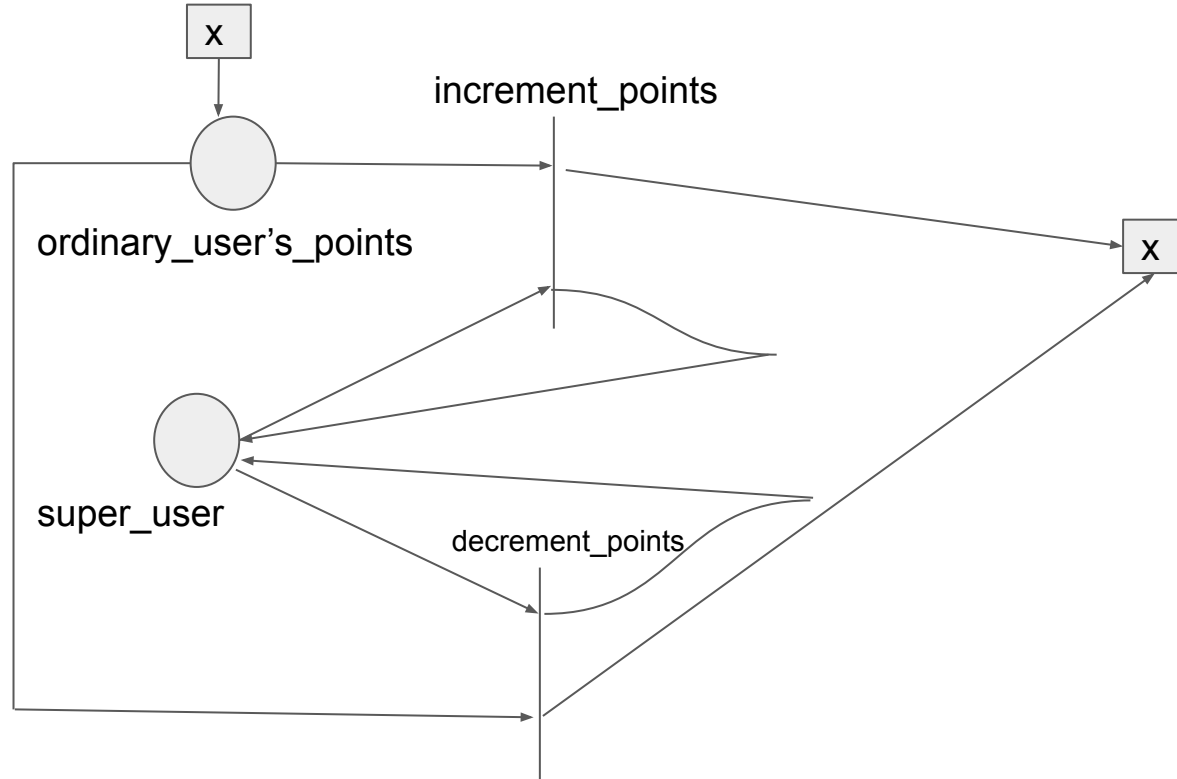
# Use Case: *Shutdown a group*

Super User can respond to the reports by shutting down a group



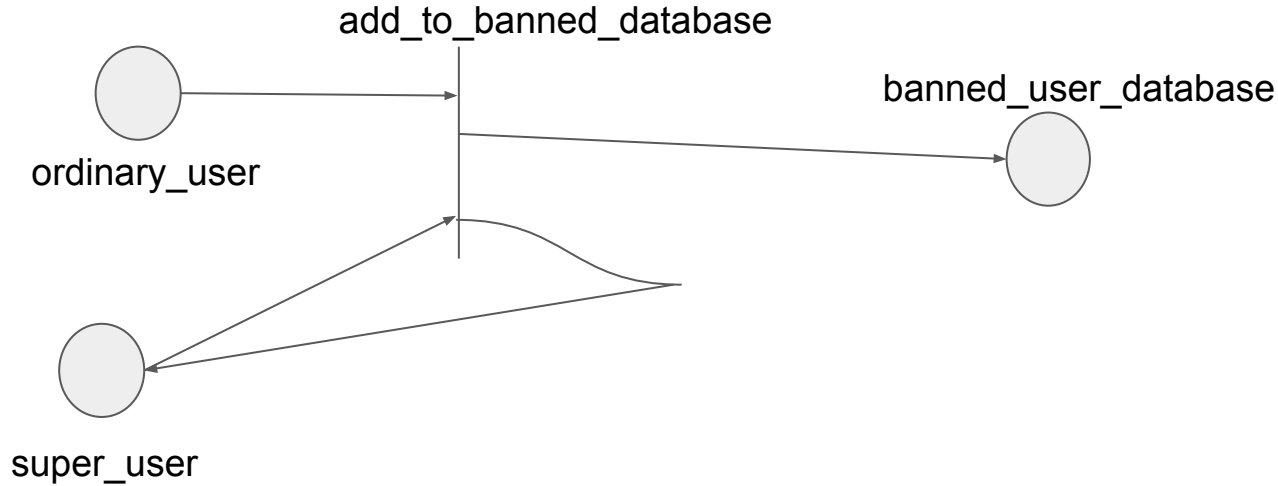
# Use Case: *Issue Point Deduction/Increment*

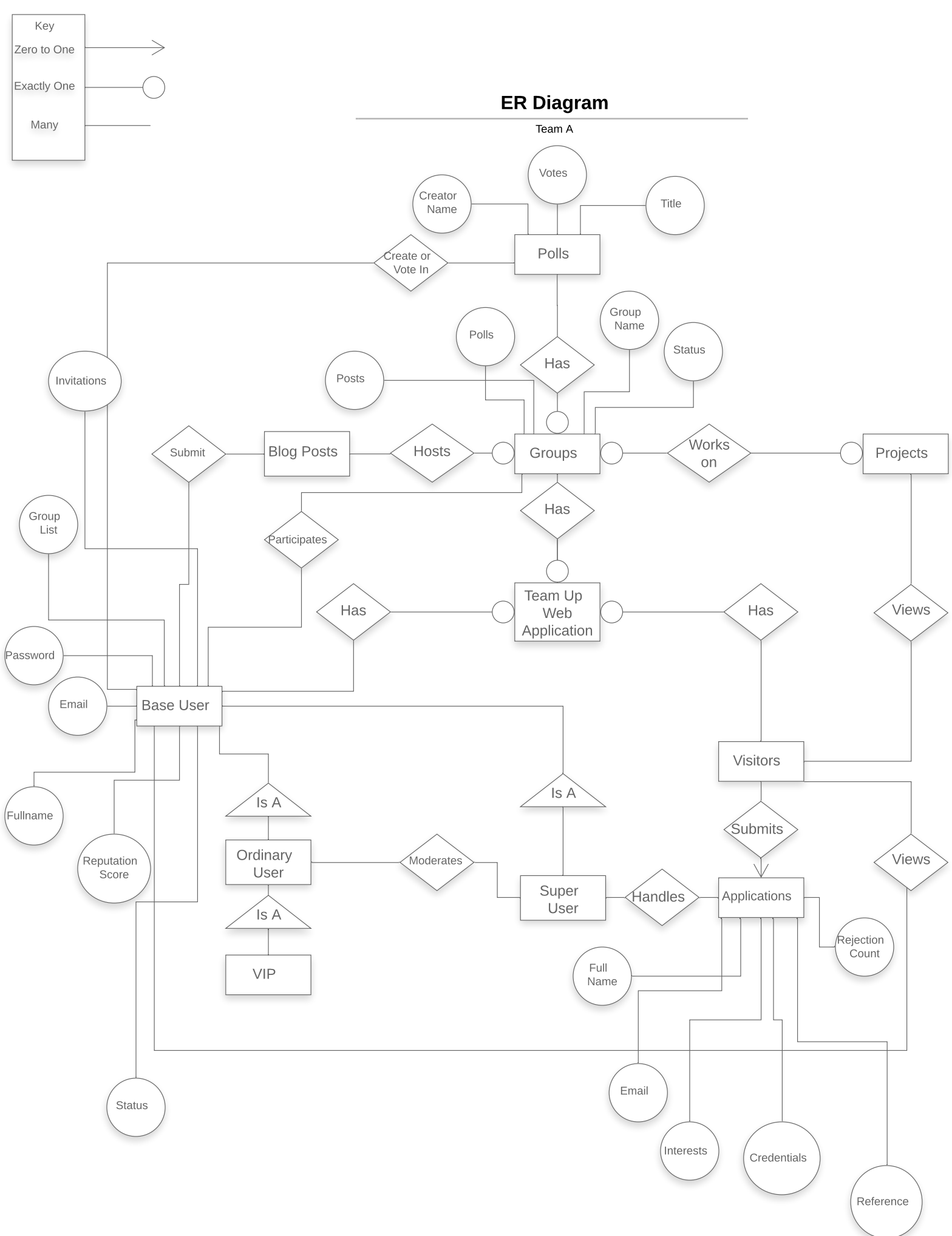
Super User can take action on users by incrementing or deducting a user's points.



# Use Case: *Ban a User*

Super User can respond to the reports by banning a user. If the user is already banned, the Super User will be notified.







Team Up	Version: 2.0
Software Design Report	Date: 05/01/2020

## 4. Detailed Design

This is a list of functions and pseudo code which we will later implement in our project. The functionality is divided into 5 modules, each containing one class.

Key: Code followed by a # represents pseudo code.

### BaseUser Class

```
class BaseUser():

    def __init__(self, name, email, username, interest, cred, rejection):
        self.name = name
        self.email = email
        self.username = username
        self.interest = interest
        self.cred = cred
        self.rejection = rejection

    def login(userID, password):
        # Input account ID, Password
        #
        # if (the account ID, password in database):
        #     approve user
        #     successful return status
        #     proceed to the User Page UI
        #
        # else:
        #     print("Your username and passwords do not match")
        #     stay on the signup page

    def inviteToGroup(senderUserID, groupName, recipientUserID):
        # if (senderUserID && recipientUserID is in User Database) &&
        #     (groupName in Group Database)
        #     row = []
        #     row.append(invite, senderUserID, groupName)
        #     recipientUser_database.table.upload(row)
        #     (backend process)
        #     print(status)

    def handleGroupMeeting(senderUserID, decision, reason, receipientUserID):
        # if (senderUserID && recipientUserID is in User Database) &&
```

Team Up	Version: 2.0
Software Design Report	Date: 05/01/2020

```

# (groupName in Group Database)
#
# row = []
# row.append(decision, senderUserID, reason, groupName,
recipientUserID)
# recipientUser_database.table.upload(row)
#
# print(status)

def createMeetupPoll(creatorUserID, pollName, pollType, optionsList):
# if (UserID is in User Database)
#
# if (pollName already exists in the Poll Database):
#     print("Poll already exists")
#
# else:
#     row = []
#     row.append(creatorUserID, pollName, optionsList, pollType,
recipientUserID)
#     Polldatabase.table.upload(row)
#
#     print("Poll added to the database")

def createWarningPoll(pollName, pollType, optionsList, targetUserID):
# if (UserID is in User Database)
#
# if (pollName already exists in the Poll Database):
#     print("Poll already exists")
#
# else:
#     row = []
#     row.append(targetUserID, pollName, optionsList, pollType)
#     Polldatabase.table.upload(row)
#     print("Poll added to the database")

def createPraisePoll(pollName, pollType, optionsList, targetUserID):
# if (UserID is in User Database)
#
# if (pollName already exists in the Poll Database):
#     print("Poll already exists")
#
# else:
#     row = []
#     row.append(targetUserID, pollName, optionsList, pollType)
#     Polldatabase.table.upload(row)
#

```

Team Up	Version: 2.0
Software Design Report	Date: 05/01/2020

```

#         print("Poll added to the database")

def createKickPoll(pollName, pollType, optionsList, targetUserID):
# if (UserID is in User Database)
#
#     if (pollName already exists in the Poll Database):
#         print("Poll already exists")
#
#     else:
#         row = []
#         row.append(targetUserID, pollName, optionsList, pollType)
#         Polldatabase.table.upload(row)
#
#         print("Poll added to the database")

def createCloseGroupPoll(pollName, pollType, optionsList, targetGroupID):
# if (GroupID is in Group Database)
#
#     if (pollName already exists in the Poll Database):
#         print("Poll already exists")
#
#     else:
#         row = []
#         row.append(targetUserID, pollName, optionsList, pollType)
#         Polldatabase.table.upload(row)
#         print("Poll added to the database")

def issueMeetupVote(pollName, UserID, decision):
# if (UserID is in vip User Database)
#
#     if (vipUserID has not voted yet):
#         row = []
#         row.append(UserID, pollName, decison)
#         Polldatabase.table.upload(row)
#         print("Your decision has been submitted")

def issueWarningVote(pollName, UserID, decision):
# if (userID is in User Database && pollName in Poll Database)
#
#     if (userID has not voted yet):
#         row = []
#         row.append(UserID, pollName, decision)
#         Polldatabase.table.upload(row)
#         print("Your decision has been submitted")
#
#     else:
#         print("You have already submitted your response for this poll")

def issuePraiseVote(pollName, UserID, decision):

```

Team Up	Version: 2.0
Software Design Report	Date: 05/01/2020

```

        # if (userID is in User Database && pollName in Poll Database)
        #
        #     if (userID has not voted yet):
        #         row = []
        #         row.append(userID, pollName, decision)
        #         Polldatabase.table.upload(row)
        #         print("Poll added to the database")
        #
        #         print("Your decision has been submitted")
        #     else:
        #         print("You have already submitted your response for this poll")

def issueKickVote(pollName, UserID, decision):
    # if (userID is in User Database && pollName in Poll Database)
    #
    #     if (userID has not voted yet):
    #         row = []
    #         row.append(userID, pollName, decision)
    #         Polldatabase.table.upload(row)
    #         print("Your decision has been submitted")
    #     else:
    #         print("You have already submitted your response for this poll")

def issueCompliment(UserId, complimentComment):
    #if (userID exists in the user database):
    #    row = []
    #    row.append(UserID, complimentComment)
    #    Userdatabase.table.upload(row)

    #    return success status
    #
    #else:
    #    print("The user you are trying to issue a compliment to, doesnt't
exist")
    pass

def addToWhiteBox(UserID):
    #if (userID exists in the user database):
    #    if (userID exists in self.whitebox database):
    #        print("User already added to whitebox")
    #
    #    else:
    #        row = []
    #        row.append(UserID)
    #        Whitelistdatabase.table.upload(row)
    #        print("User added to your whitebox")
    #
    #    else:

```

Team Up	Version: 2.0
Software Design Report	Date: 05/01/2020

```

#     print("The user you are trying to whitelist doesn't exist")

def addToBlackBox(userID):
    #if (userID exists in the user database):
    #     if (userID exists in self.blackbox database):
    #         print("User already to blackbox")
    #     else:
    #         row = []
    #         row.append(userID)
    #         blacklistdatabase.table.upload(row)
    #         print("User added to your blackbox")
    #         print("User banned")
    #     else:
    #         print("The user you are trying to ban doesn't exist")

```

### **OrdinaryUser Class**

```

class OrdinaryUser(BaseUser):
    def __init__(self):
        BaseUser.__init__(self)

    def appealReputation(appealMessage):
        # input appealMessage
        # row = []
        # row.append(appealMessage)
        # SuperUser_inbox_database.table.upload(row)
        #
        # if (append to SuperUser Inbox database status == "success"):
        #     print("Appeal submitted successfully")

    def reportUser(userID, reportMessage):
        # input UserID and reportMessage
        # row = []
        # row.append(userID, reportMessage)
        # SuperUser_inbox_database.table.upload(row)
        #
        # if (append to SuperUser Inbox database status == "success"):
        #     print("Report submitted successssfully")

    def reportGroup(groupName, reportMessage):
        # input groupName and reportMessage
        #
        # if (groupName is in Group Database)

```

Team Up	Version: 2.0
Software Design Report	Date: 05/01/2020

```

# row = []
# row.append(groupName, reportMessage)
# SuperUser_inbox_database.table.upload(row)
#
# if (append to SuperUser Inbox databae status == "success"):
#     print("Report submitted successssfully")

def referenceReputation(UserID, reputationPoints):
# input UserID and reputationPoints that will be given
#
# if (UserID is in User Database):
#     currentUser.scorepoints += reputationPoints
#     print("Initial reputation score has been successfully added")

```

### **SuperUser Class**

```

class SuperUser(BaseUser):
    def __init__(self):
        BaseUser.__init__(self)

    def handleApplication(UserID, decision):
# input UserID
# if (UserID exists in the pending application database):
#     if (decision == accept):
#         remove the user from PendingUser Database
#         row = []
#         row.append(UserID, decision)
#         database.table.upload(row)
#         print("User Registered Successfully")
#
#     elif (decision == deny):
#         if (number_of_times_denied > 1):
#             add to blacklist
#
#         number_of_times_denied += 1
#         print("User Registration denied")

    def blacklistFromServer(UserID):
# input UserID
# if (UserID exists in the User Database):
#     if (User already not exists in BlackList database):
#         row = []
#         row.append(UserID)
#         BlackList_Database.table.upload(row)

```

Team Up	Version: 2.0
Software Design Report	Date: 05/01/2020

```

def reverseReputationDeduction(UserID, decision):
    # input UserID
    # if (UserID exists in the UserDatabase):
    #     databasetable.currentUser.reputationpoint += 5
    #     print("Reputation Deduction is reversed")

def shutDownGroup(groupName):
    # input groupName
    # if(groupName exist in the Groups Database):
    #     GroupsDatabase.remove(groupName)
    #     print("Group was shutdown successfully")
    #

def issuePointDeduction(UserID):
    # if (userID exists in the user database):
    #     currentUser.reputationpoint -= 5
    #     print("Reputation Deduction done")

def issuePointIncrement(UserID, ):
    # if (userID exists in the user database):
    #     currentUser.reputationpoint += 5
    #     print("Reputation Increment done")

def banUser(UserID):
    # if (userID exists in the user database):
    #     if (userID exists in blacklist database):
    #         print("User already banned")
    #     else:
    #         row = []
    #         row.append(UserID)
    #         BlackList_Database.table.upload(row)
    #     else:
    #         print("The user you are trying to ban doesn't exist")
    #

```

### **VipUser Class**

```

class VipUser(OrdinaryUser):
    def __init__(self):
        OrdinaryUser.__init__(self)

```

Team Up	Version: 2.0
Software Design Report	Date: 05/01/2020

```

def createDemocraticSuperUserPoll(vipUserID, pollName, pollType,
optionsList):
    # if (vipUserID is in vip User Database)
    #
    #   if (pollName already exists in the Poll Database):
    #       print("Poll already exists")
    #
    #   else:
    #       row = []
    #       row.append(vipUserID, pollName, pollType, optionsList)
    #       database.table.upload(row)

    #   return status
    #

def issueDemocraticSuperUserVote(pollName, vipUserID, decision):
    # if (vipUserID is in vip User Database && pollName in Poll Database)
    #
    #   if (vipUserID has not voted yet):
    #       row = []
    #       row.append(pollName, vipUserID, decision)
    #       database.table.upload(row)
    #       print("Your decision has been submitted")

```

## Visitor Class

```

class Visitor():
    def __init__(self, name, email, password, referredBy, topicOfInterest):
        self.name = name
        self.email = email
        self.password = password
        self.referredBy = referredBy
        self.topicOfInterest = topicOfInterest

    def signUp(self):
        # Take in the following info from visitor: name, email, password,
        # credentials, referral
        #
        #   if (user already exists in the PendingUsers database):
        #       print("Please wait while your current application status is under
        # review")
        #
        #   if (user already exists in the User Database):
        #       print("Account already exists, Sign in")

```



Team Up	Version: 2.0
Software Design Report	Date: 05/01/2020

```

#
# if (the user exists in the blacklist database):
#     print("You are banned, signup denied")
#
# else:
#     row = []
#     row.append(name, email, password, credentials, referral)
#     Userdatabase.table.upload(row)
#     user application status returned
#     verify that application has been submitted

def appealRejection(userID, message):
    # Visitor inputs an appeal message
    #
    # if (the visitor userID exists in the PendingUser Database):
    #     fetch the User data
    #     row = []
    #     row.append(userID, message)
    #     SuperUser.Database.table.upload(row) #send the user's message and
appeal request to SuperUser
    #
    #
    # if (the appeal request submitted to the User database successfully):
    #     return the application status
    # else:
    #     return failed status

```

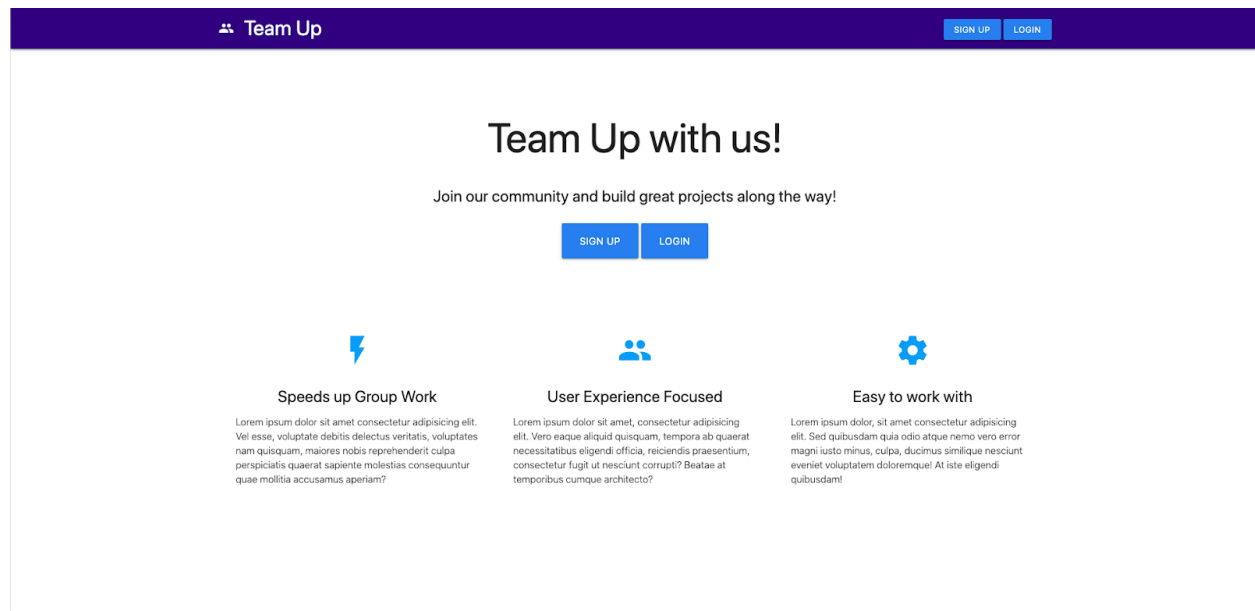
## 5. User Interface Screens

Team Up	Version: 2.0
Software Design Report	Date: 05/01/2020

We created the user interface prototype for each user of the web application (Visitor, Ordinary User, VIP, Super User). These prototypes are used as a starting point to help developers have a better understanding of the overall User Interface of the web application as perspectives of visitors, OUs, VIPs, and SUs. These drafts will be developed upon along the way and are used to consolidate the overall ideas on pages and components of the web app. Technologies that will be used include HTML CSS JavaScript AOS animation jQuery and Materialize library.

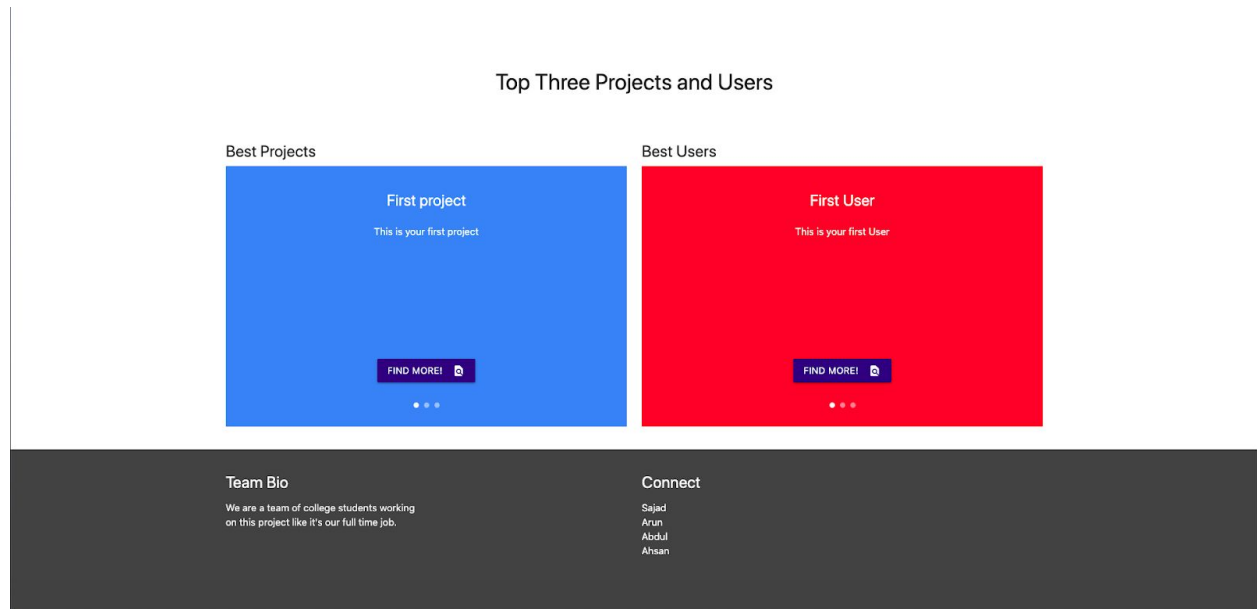
## Landing Page

This is the very first page any user interacts with. They can scroll down to see more, or just signup or login.



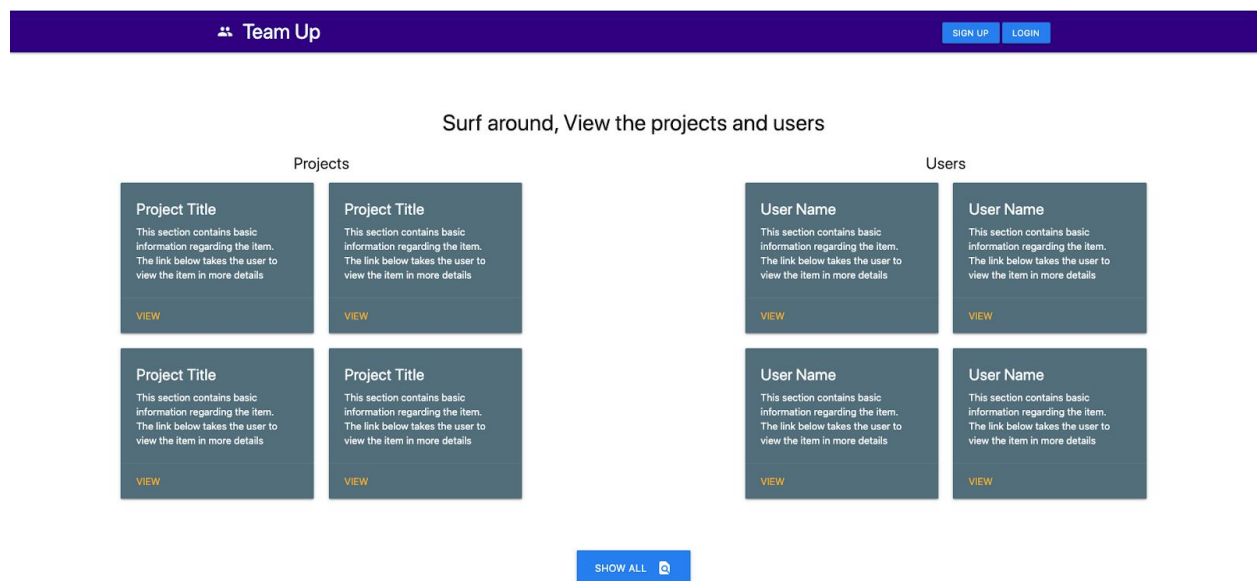
When the user scrolls down, they'd see three of the top projects and users. They can also click on Find More! To be taken to the Surfing page.

Team Up	Version: 2.0
Software Design Report	Date: 05/01/2020



## Surfing Page

This page is visible to all users including visitors. Once the Find More button is clicked, they'll be routed to this page to surf around for more projects and users.



Team Up	Version: 2.0
Software Design Report	Date: 05/01/2020

# Signup and Login Pages

Used for registration of new users or logging in the members of the web application

Team Up

SIGN UP

LOGIN

Sign up for a free account

First Name

Last Name

Email

Password

Interests

Enter an interest

Referring Member

SIGN UP ✓

Team Bio

We are a team of college students working on this project like it's our full time job.

Connect

Sajad

Arun

Abdul

Team Up

SIGN UP

LOGIN

Login

Email or Username

Password

LOGIN ✓

Team Bio

We are a team of college students working on this project like it's our full time job.

Connect

Sajad

Arun

Abdul

Ahsan

Team Up	Version: 2.0
Software Design Report	Date: 05/01/2020

## Ordinary User Page

Main page for the OU user when they log in, will contain information and features available for them.

Team Up

LOGOUT

GROUPS

INVITATIONS

WHITE LIST

BLACKLIST

CREATE A GROUP

PENDING SCORES

Ordinary User name

Your Current Groups

Group Title

This section contains basic information regarding the item.  
The link below takes the user to view the item in more details

VIEW

Group Title

This section contains basic information regarding the item.  
The link below takes the user to view the item in more details

VIEW

Group Title

This section contains basic information regarding the item.  
The link below takes the user to view the item in more details

VIEW

Your Invitations

Group Title

This section contains basic information regarding the item.  
The link below takes the user to view the item in more details

ACCEPTDECLINE

Group Title

This section contains basic information regarding the item.  
The link below takes the user to view the item in more details

ACCEPTDECLINE

Group Title

This section contains basic information regarding the item.  
The link below takes the user to view the item in more details

ACCEPTDECLINE

Your White List

Your White List

ADD TO WHITE LIST

User name

This section contains basic information regarding the item.  
The link below takes the user to view the item in more details

REMOVE

User name

This section contains basic information regarding the item.  
The link below takes the user to view the item in more details

REMOVE

User name

This section contains basic information regarding the item.  
The link below takes the user to view the item in more details

REMOVE

User name

This section contains basic information regarding the item.  
The link below takes the user to view the item in more details

REMOVE

Your Black List

Your Black List

ADD TO BLACK LIST

User name

This section contains basic information regarding the item.  
The link below takes the user to view the item in more details

REMOVE

User name

This section contains basic information regarding the item.  
The link below takes the user to view the item in more details

REMOVE

User name

This section contains basic information regarding the item.  
The link below takes the user to view the item in more details

REMOVE

User name

This section contains basic information regarding the item.  
The link below takes the user to view the item in more details

REMOVE

37

Team Up	Version: 2.0
Software Design Report	Date: 05/01/2020

### Create a Group

Group Name

Invites' usernames

Enter a username

SUBMIT ✓

### Pending Initial Scores

There's currently no item here

## VIP User Page

Main page for VIP Users when they log in.

Team Up

LOGOUT

GROUPS

INVITATIONS

WHITE LIST

BLACKLIST

CREATE A GROUP OR VOTE

PENDING SCORES

VIP User name

### Your Current Groups

Group Title

This section contains basic information regarding the item.  
The link below takes the user to view the item in more details

VIEW

Group Title

This section contains basic information regarding the item.  
The link below takes the user to view the item in more details

VIEW

Group Title

This section contains basic information regarding the item.  
The link below takes the user to view the item in more details

VIEW

### Your Invitations

Group Title

This section contains basic information regarding the item.  
The link below takes the user to view the item in more details

ACCEPT DECLINE

Group Title

This section contains basic information regarding the item.  
The link below takes the user to view the item in more details

ACCEPT DECLINE

Group Title

This section contains basic information regarding the item.  
The link below takes the user to view the item in more details

ACCEPT DECLINE

Team Up	Version: 2.0
Software Design Report	Date: 05/01/2020

#### Your White List

ADD TO WHITE LIST

User name

This section contains basic information regarding the item. The link below takes the user to view the item in more details

REMOVE

User name

This section contains basic information regarding the item. The link below takes the user to view the item in more details

REMOVE

User name

This section contains basic information regarding the item. The link below takes the user to view the item in more details

REMOVE

User name

This section contains basic information regarding the item. The link below takes the user to view the item in more details

REMOVE

#### Your Black List

ADD TO BLACK LIST

User name

This section contains basic information regarding the item. The link below takes the user to view the item in more details

REMOVE

User name

This section contains basic information regarding the item. The link below takes the user to view the item in more details

REMOVE

User name

This section contains basic information regarding the item. The link below takes the user to view the item in more details

REMOVE

User name

This section contains basic information regarding the item. The link below takes the user to view the item in more details

REMOVE

#### Create a Group

Group Name

Invitees' usernames

Enter a username

SUBMIT ✓

#### Vote for a Democratic Super User

It's not the time to vote yet!

#### Pending Reputation Scores

There's currently no item here

#### Pending Initial Scores

There's currently no item here

## Super User Page

Main page for Super Users when they log in.

Team Up	Version: 2.0
Software Design Report	Date: 05/01/2020

Team Up

LOGOUT

GROUPS
INVITATIONS
WHITE LIST
BLACKLIST
CREATE GROUP
COMPLIMENTS COMPLAINS
CLOSED GROUPS

Super User name

### New Registration Reuquests

New User Name
This section contains basic information regarding the user.

ACCEPTDECLINE

New User Name
This section contains basic information regarding the item.

ACCEPTDECLINE

New User Name
This section contains basic information regarding the item.

ACCEPTDECLINE

### Your Current Groups

Group Title
This section contains basic information regarding the item. The link below takes the user to view the item in more details

VIEW

Group Title
This section contains basic information regarding the item. The link below takes the user to view the item in more details

VIEW

Group Title
This section contains basic information regarding the item. The link below takes the user to view the item in more details

VIEW

### Your Invitations

### Your Invitations

Group Title
This section contains basic information regarding the item. The link below takes the user to view the item in more details

ACCEPTDECLINE

Group Title
This section contains basic information regarding the item. The link below takes the user to view the item in more details

ACCEPTDECLINE

Group Title
This section contains basic information regarding the item. The link below takes the user to view the item in more details

ACCEPTDECLINE

### Your White List

ADD TO WHITE LIST

User name
This section contains basic information regarding the item. The link below takes the user to view the item in more details

REMOVE

User name
This section contains basic information regarding the item. The link below takes the user to view the item in more details

REMOVE

User name
This section contains basic information regarding the item. The link below takes the user to view the item in more details

REMOVE

User name
This section contains basic information regarding the item. The link below takes the user to view the item in more details

REMOVE



Team Up	Version: 2.0
Software Design Report	Date: 05/01/2020

#### Your Black List

ADD TO BLACK LIST

User name  
This section contains basic information regarding the item.  
The link below takes the user to view the item in more details  
REMOVE

User name  
This section contains basic information regarding the item.  
The link below takes the user to view the item in more details  
REMOVE

User name  
This section contains basic information regarding the item.  
The link below takes the user to view the item in more details  
REMOVE

User name  
This section contains basic information regarding the item.  
The link below takes the user to view the item in more details  
REMOVE

#### Create a Group

Group Name

Invitees' usernames

Enter a username

SUBMIT ✓

#### New Compliments and Complains

Compliment  
Username Mike33 received 3 compliments  
This section contains basic information regarding the item.  
RESPOND

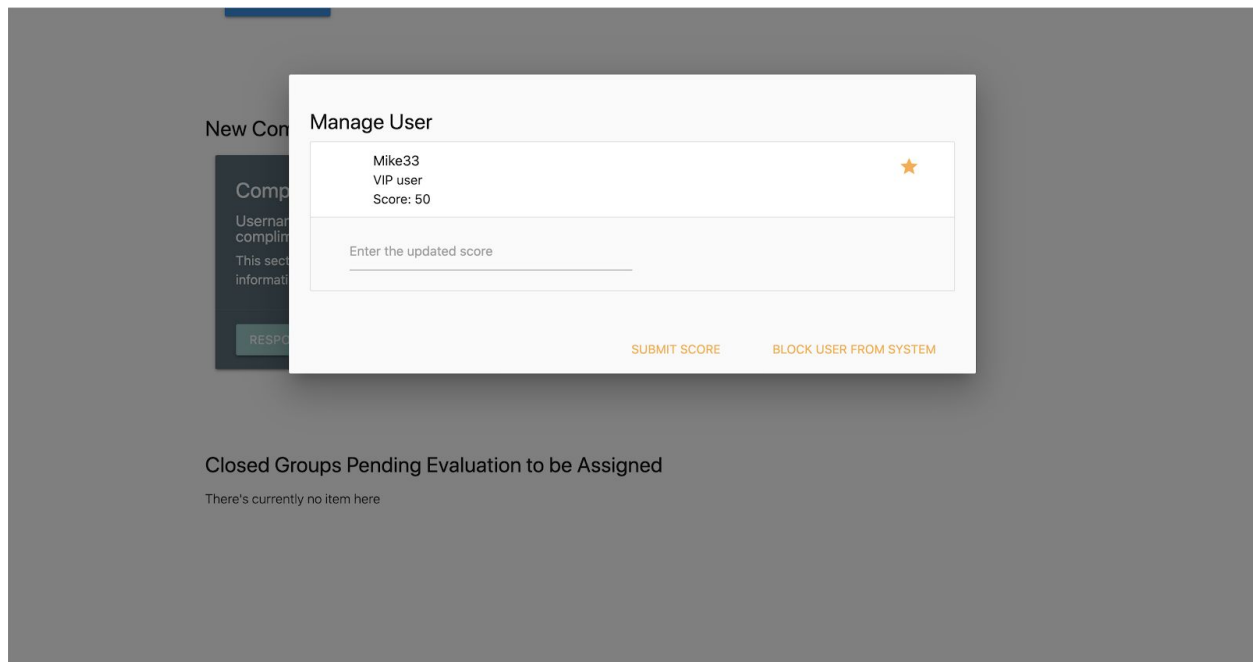
Complain  
Username Mike33 received a compliments  
This section contains basic information regarding the item.  
RESPOND

#### Closed Groups Pending Evaluation to be Assigned

There's currently no item here

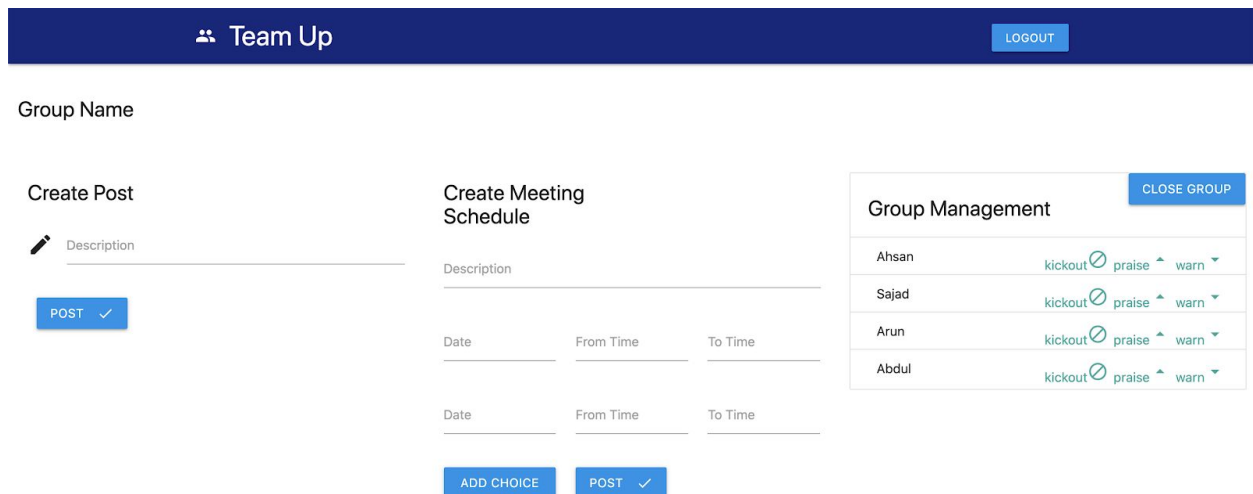
As one of the responsibilities of the super user, they will receive compliments of complaints they need to be taken care of. Here's a rough example of how a complaint can be responded by a super user. The user can be punished by a deduction of scores of being blocked from the system.

Team Up	Version: 2.0
Software Design Report	Date: 05/01/2020



## Group Page

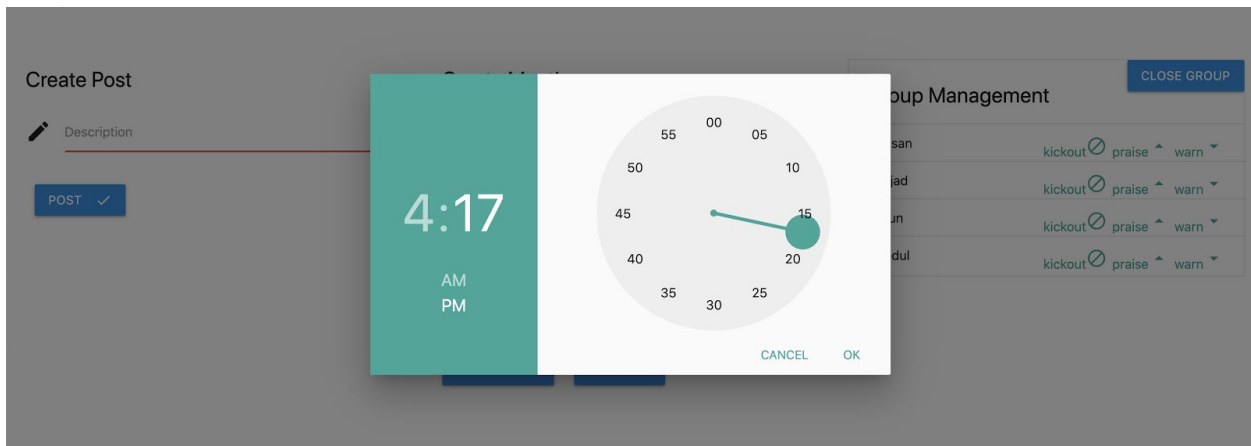
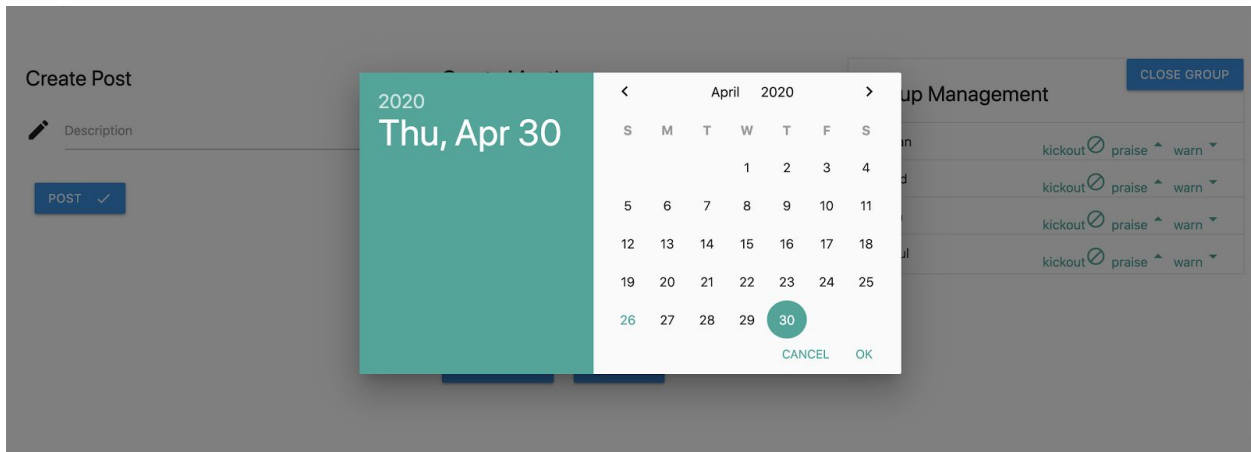
Group Page for scheduling meetups, polling and posting updates and also reporting compliments or complains.



Creating a Meeting Schedule with several choices to vote for.

Team Up	Version: 2.0
Software Design Report	Date: 05/01/2020

The user will fill in the form. Write the description of the meeting, choose the dates and times of the meeting as follows.



Now two choices have been added.

Team Up	Version: 2.0
Software Design Report	Date: 05/01/2020

Create Post

Description

POST ✓

Create Meeting Schedule

Description

This is going to be a great meeting!

Date

From Time

To Time

04/30/2020

04:17 PM

05:15 PM

Date

From Time

To Time

04/30/2020

05:17 PM

06:30 PM

ADD CHOICE

POST ✓

Group Management

CLOSE GROUP

Ahsan	kickout	praise	warn
Sajad	kickout	praise	warn
Arun	kickout	praise	warn
Abdul	kickout	praise	warn

They may add more choices for the meeting, by clicking on Add Choice. And when ready, “Post” the poll.

Group Name

Create Post

Description

POST ✓

Create Meeting Schedule

Description

This is going to be a great meeting!

Date

From Time

To Time

04/30/2020

04:17 PM

05:15 PM

Date

From Time

To Time

04/30/2020

05:17 PM

06:30 PM

Date

From Time

To Time

04/30/2020

06:35 PM

07:40 PM

ADD CHOICE

POST ✓

Group Management

CLOSE GROUP

Ahsan	kickout	praise	warn
Sajad	kickout	praise	warn
Arun	kickout	praise	warn
Abdul	kickout	praise	warn

The polls and posts will appear below. Users are allowed to vote for the time that works best for them.

Team Up	Version: 2.0
Software Design Report	Date: 05/01/2020

#### Recent Posts and Schedules

Sajad posted a poll

Hey guys, Meeting is to be scheduled to work on the software engineering group project.

Please pick the best time and vote for one of the choices below

- ☐ 05/01/2020 From 10:00 AM to 12:00 AM
- ☐ 05/01/2020 From 10:00 AM to 12:00 AM
- ☐ 05/01/2020 From 12:00 AM to 2:00 PM
- ☐ 05/01/2020 From 02:00 PM to 4:00 PM

SUBMIT ✓

Ahsan posted an update

Hey guys, please try to get here on time.

The meeting will start at the exact time.

## 6. Meeting Minutes

Dates	Duration	Agenda
04/16/2020	90 mins	<ul style="list-style-type: none"> <li>- Planned out and divided up different parts of the report.</li> <li>- Designated specific tasks to each member of the team.</li> <li>- Discussed the prototypes and the general approach for the design of the project and the diagrams.</li> </ul>
04/20/2020	69 mins	<ul style="list-style-type: none"> <li>- Discussed backend side of the project               <ul style="list-style-type: none"> <li>- Covered database usage for storing and requesting user data</li> <li>- Discussed the pros and cons of hosting the backend on firebase or AWS.</li> <li>- Weeded out repeated functionality by organizing users into classes with inheritance.</li> </ul> </li> </ul>

Team Up	Version: 2.0
Software Design Report	Date: 05/01/2020

04/25/2020	112 mins	<ul style="list-style-type: none"> <li>- Discussed the pseudocode and functionality</li> <li>- Reviewed pseudocode implementations of functions</li> <li>- Worked on and reviewed collaboration diagrams and ER diagrams</li> </ul>
05/01/2020	70 mins	<ul style="list-style-type: none"> <li>- Finalized the report</li> <li>- Final Feedback from Team Members</li> <li>- Fixed Errors</li> </ul>

## 7. GitHub Repository

You may also find the source code and *phase I and II report* on our [GitHub repository](https://github.com/sajadgzd/softwareEngineeringProject)

<https://github.com/sajadgzd/softwareEngineeringProject>