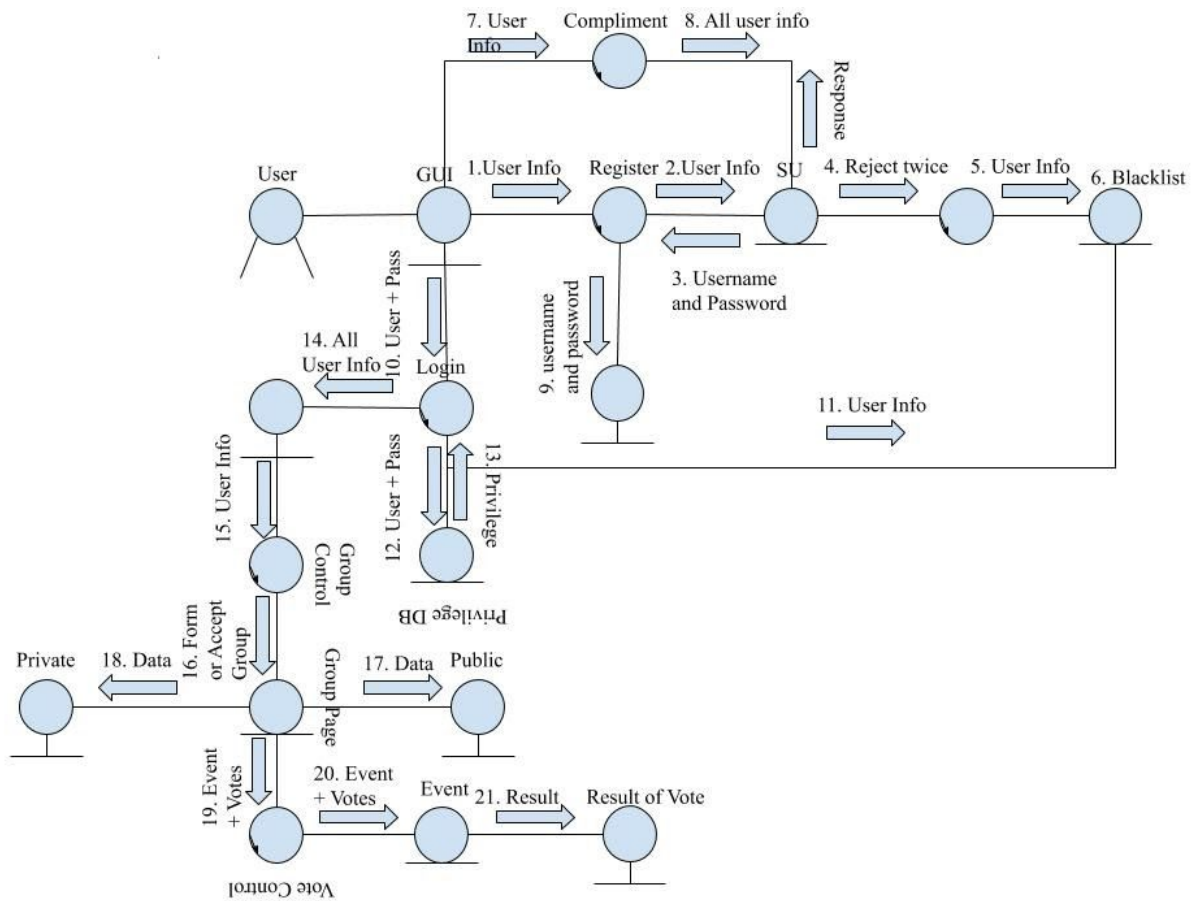
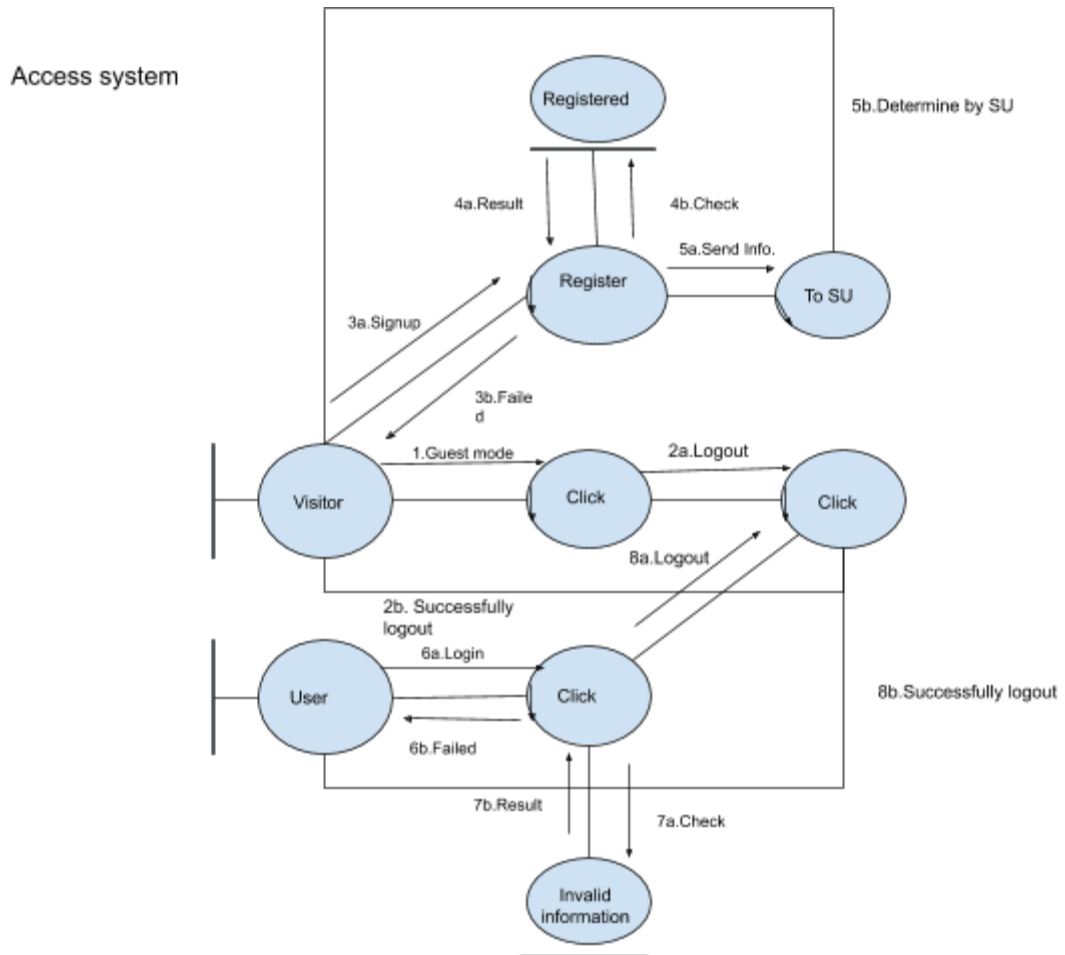
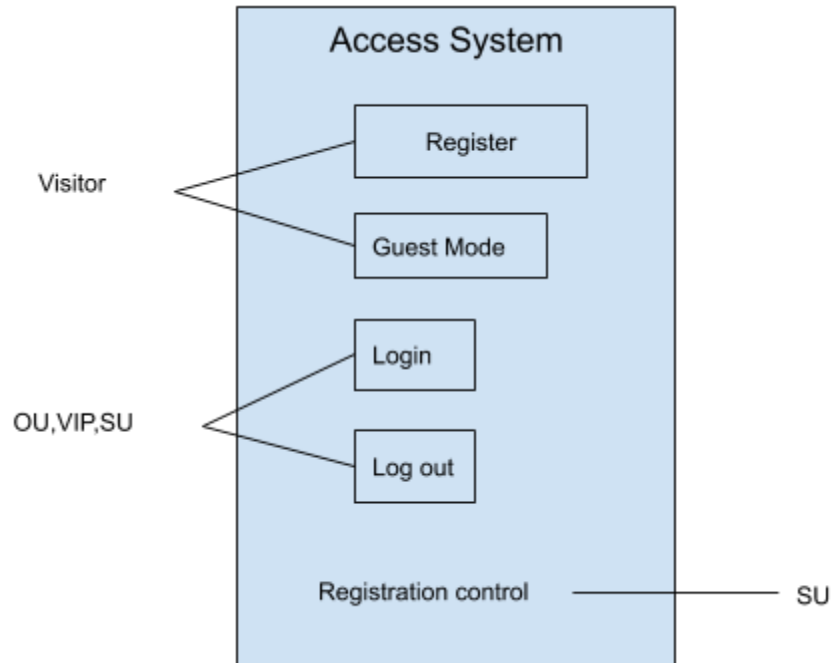


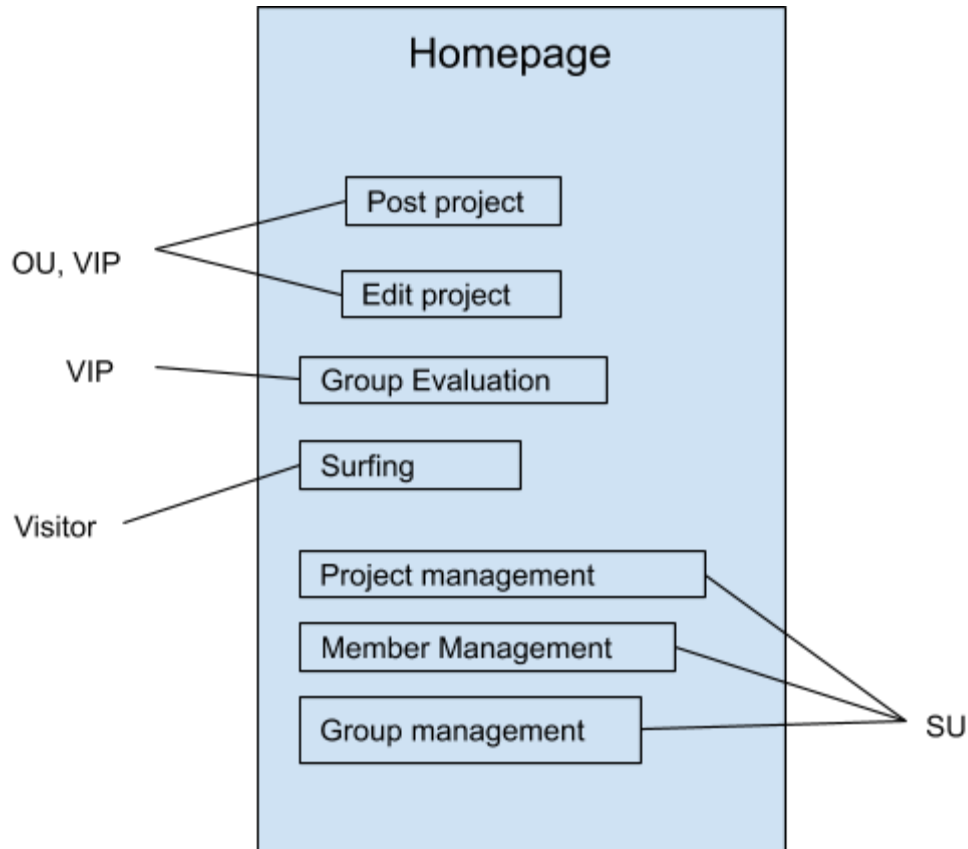
1. Introduction



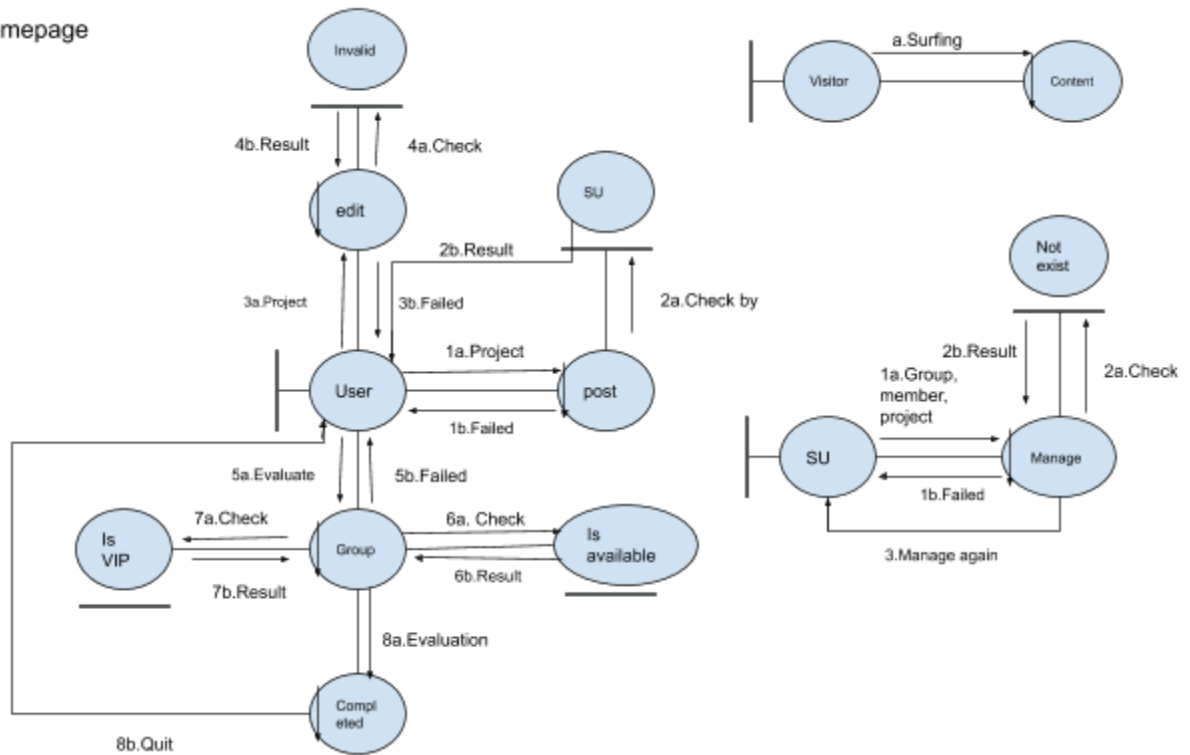
2. All use cases

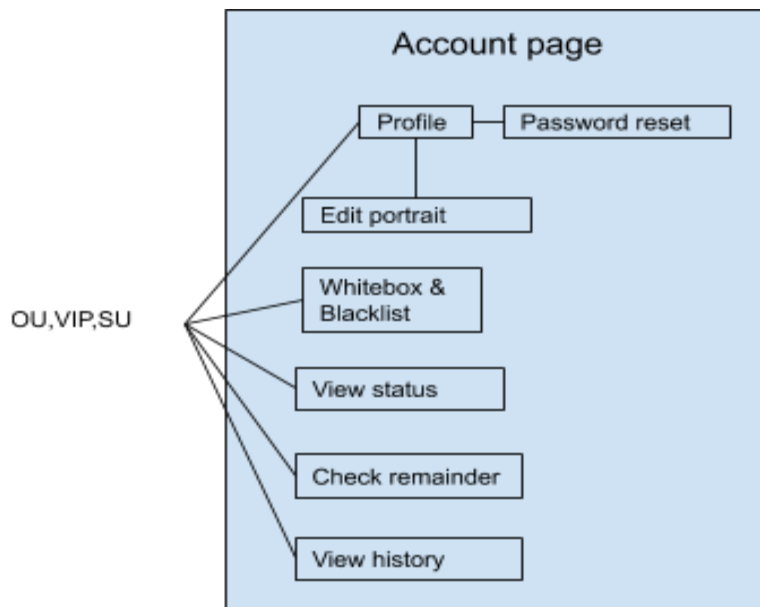
Use cases (7/7)



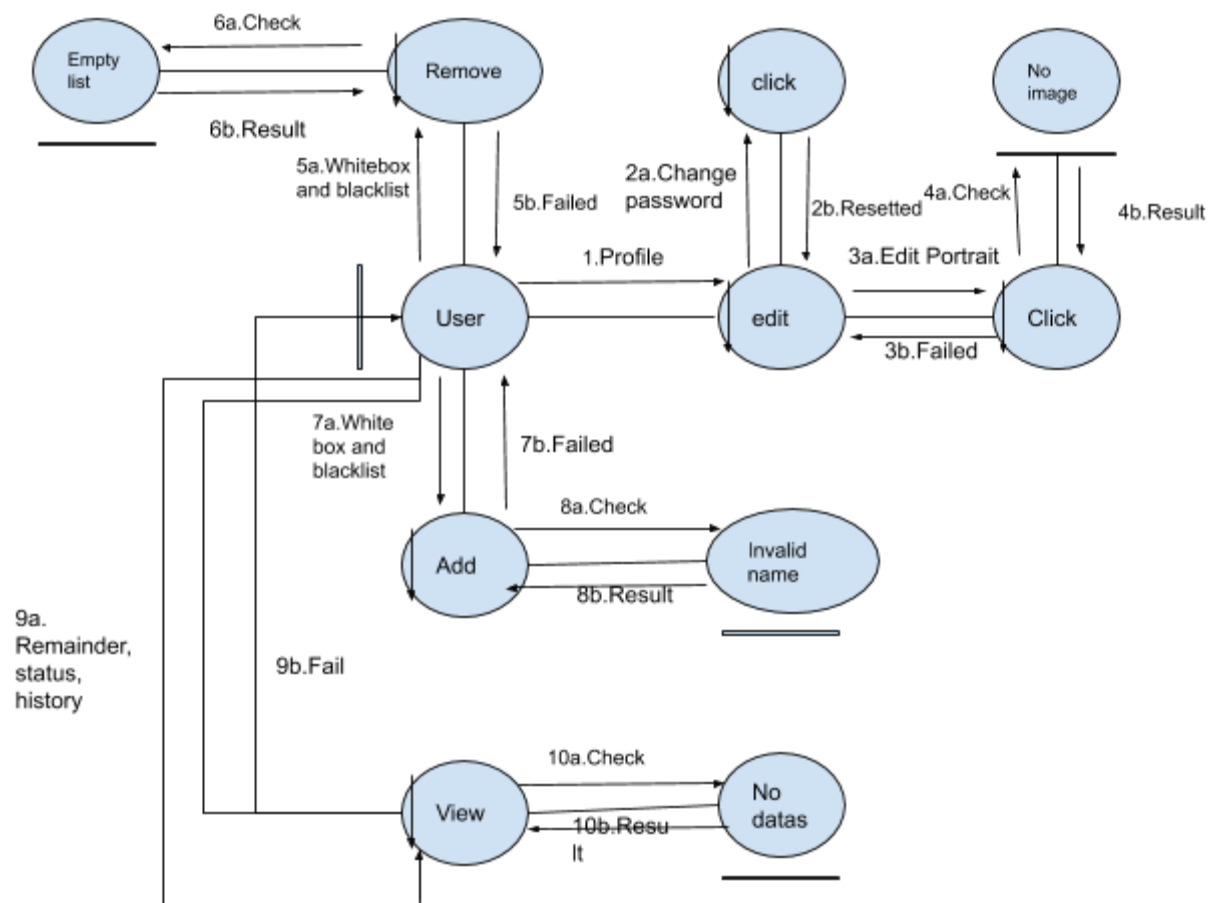


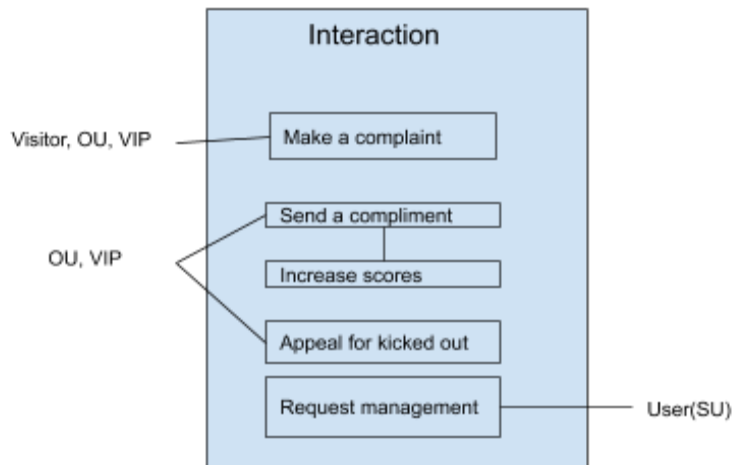
Homepage



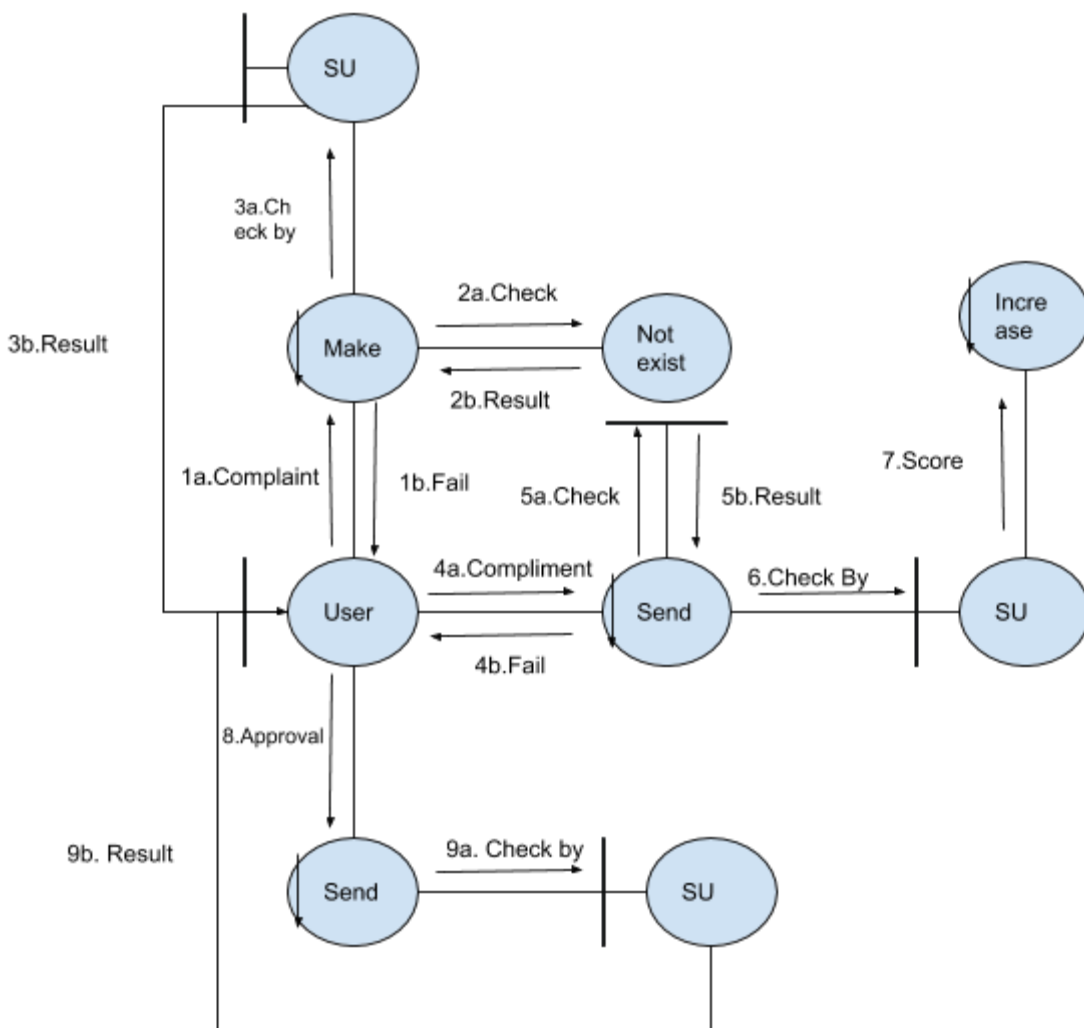


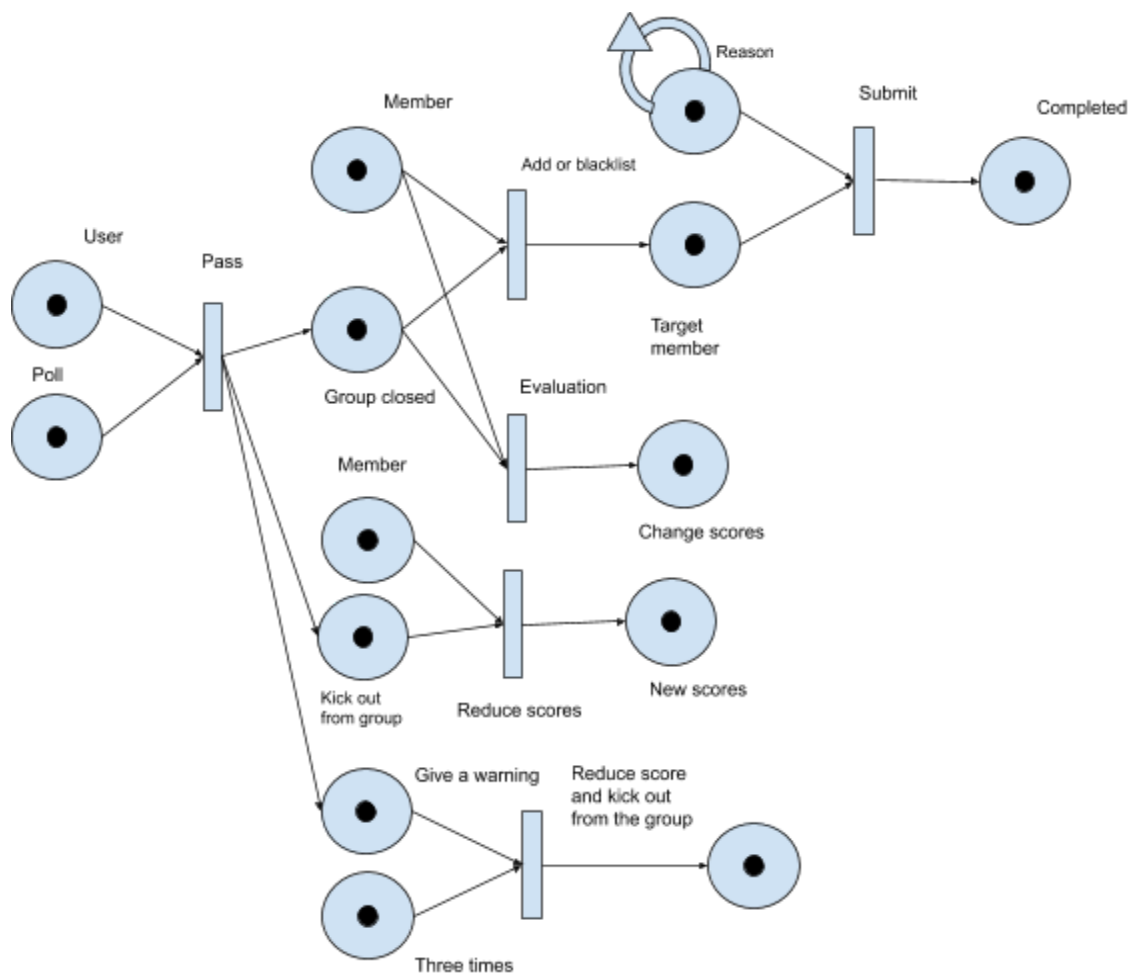
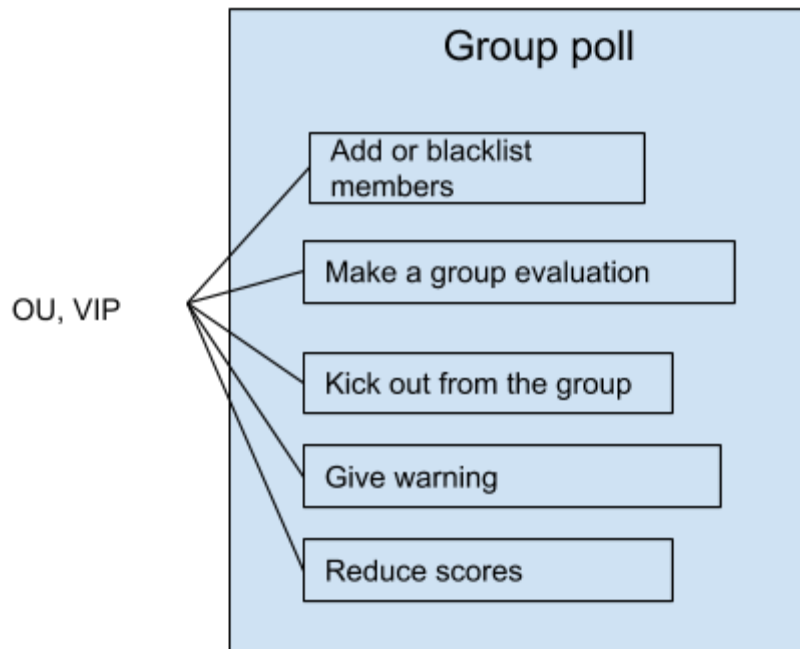
Account page

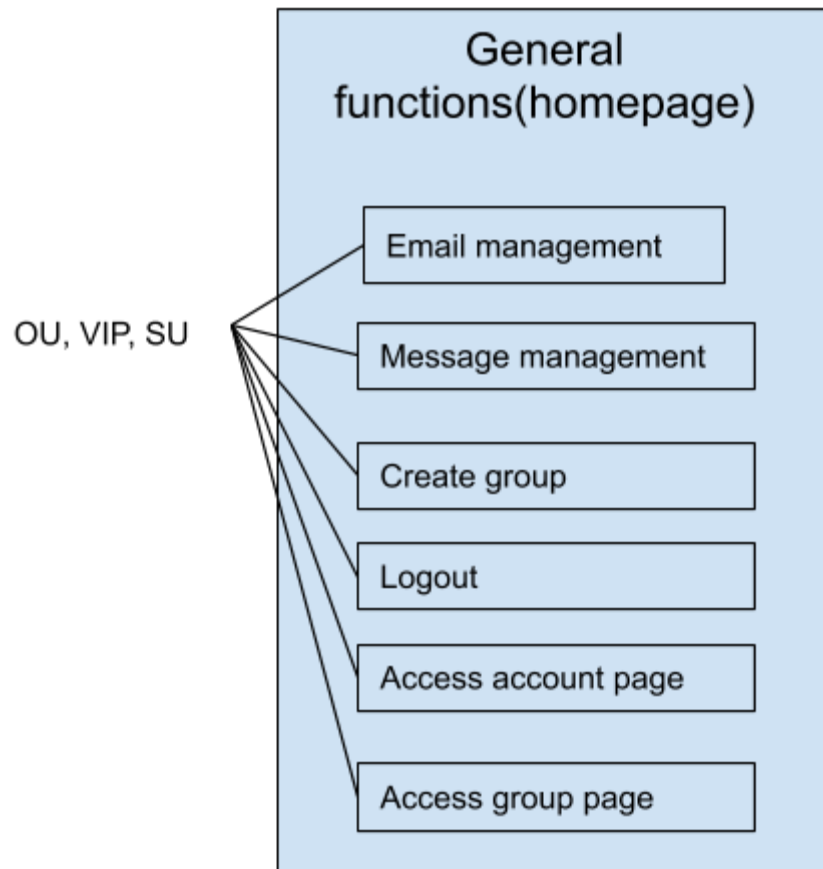


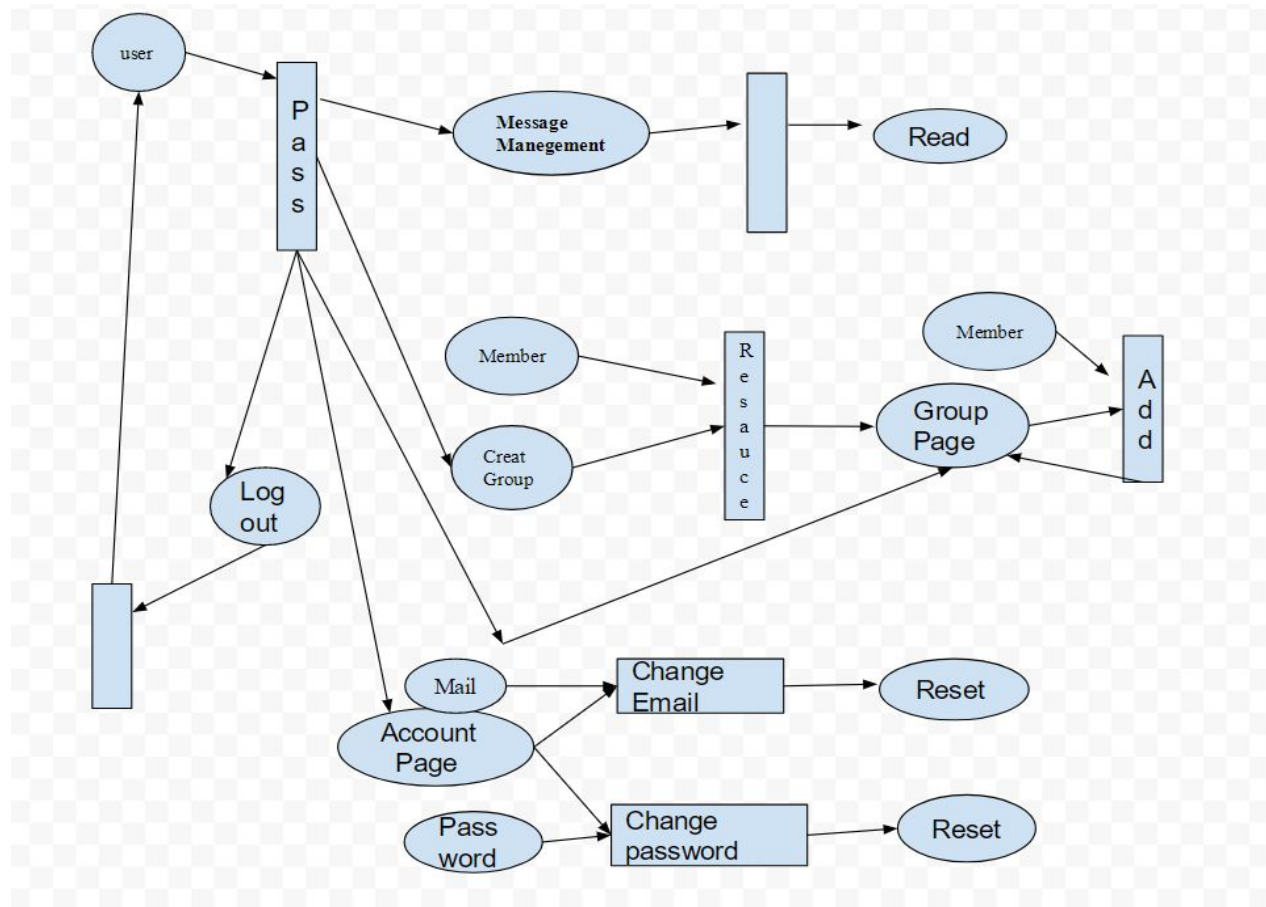


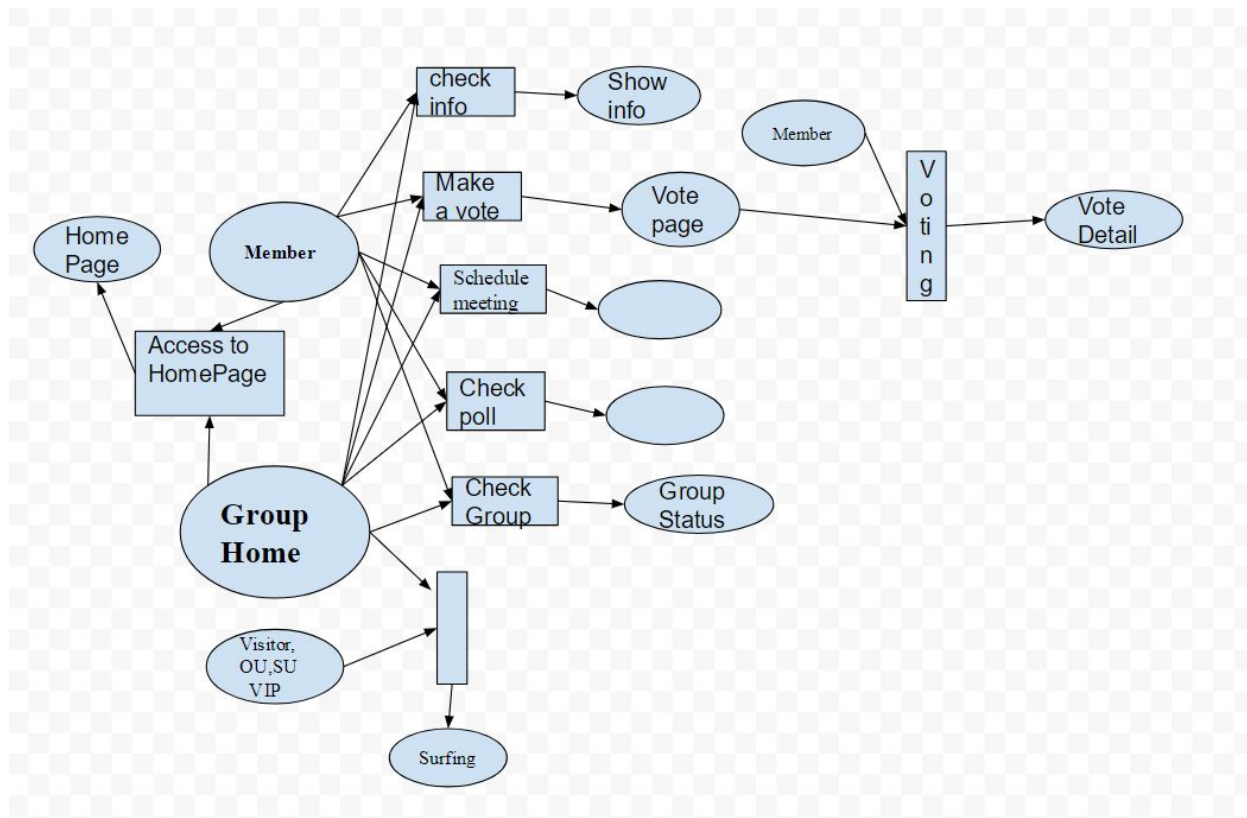
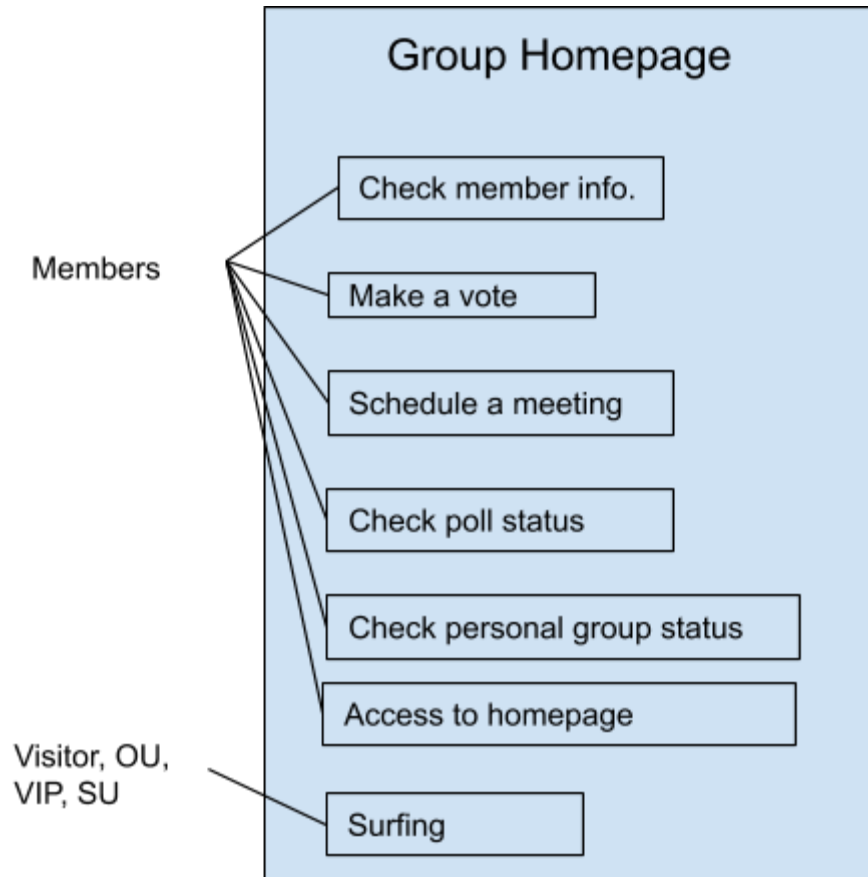
Interaction



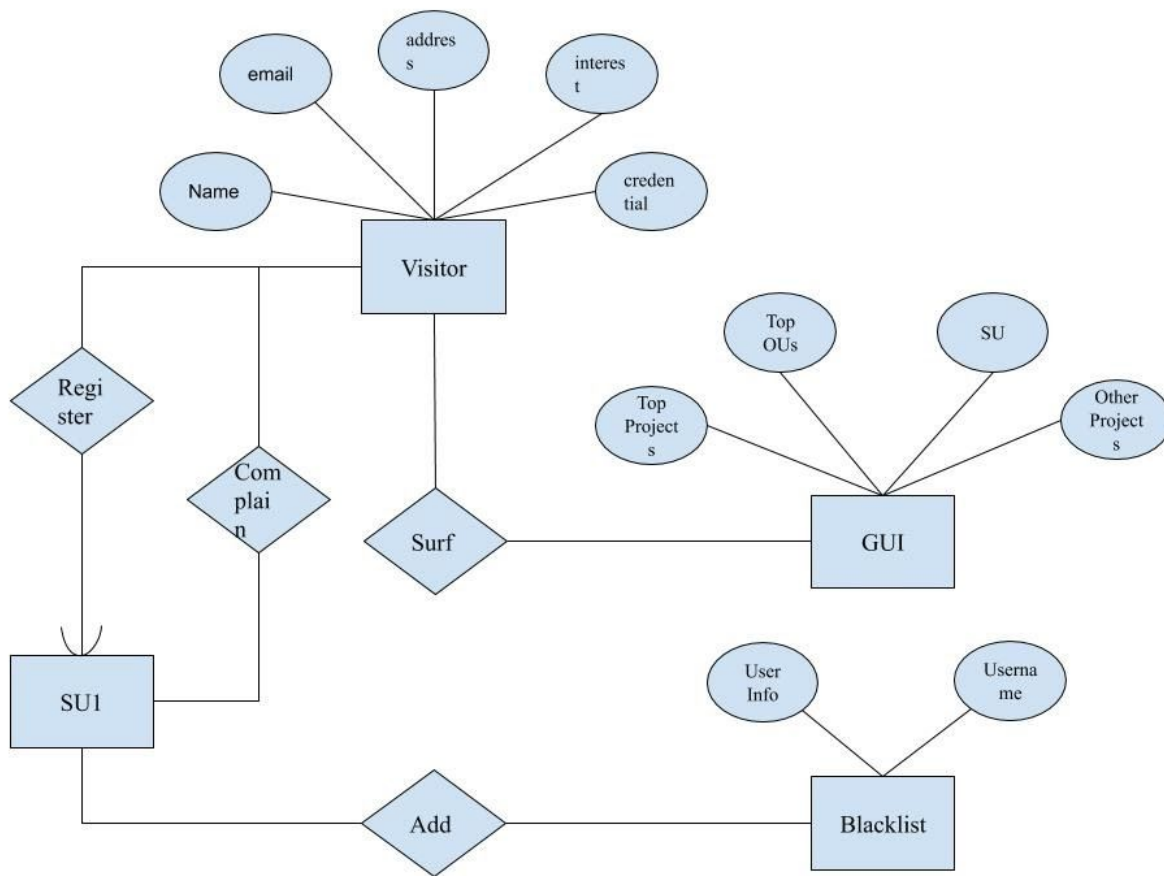


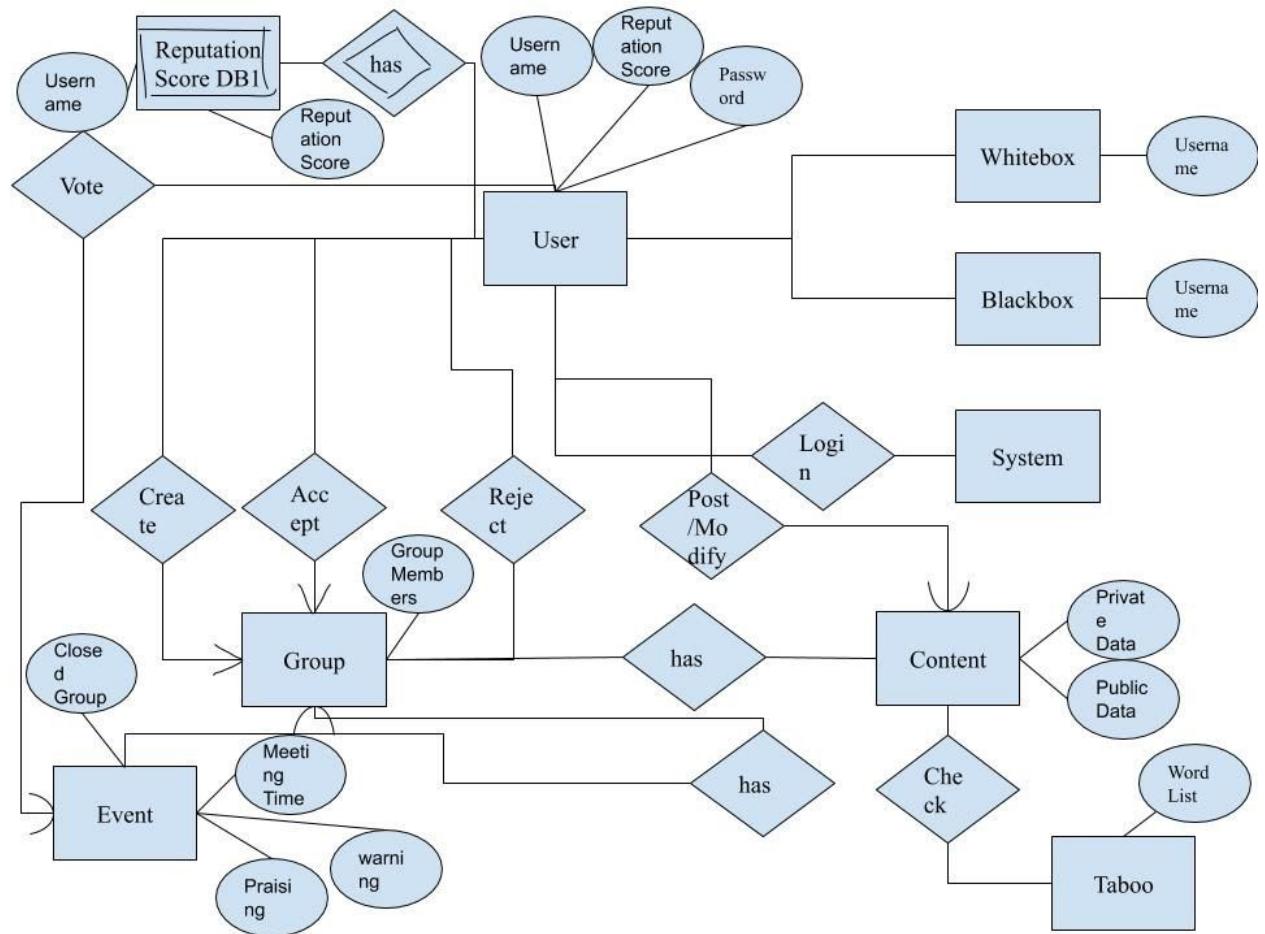


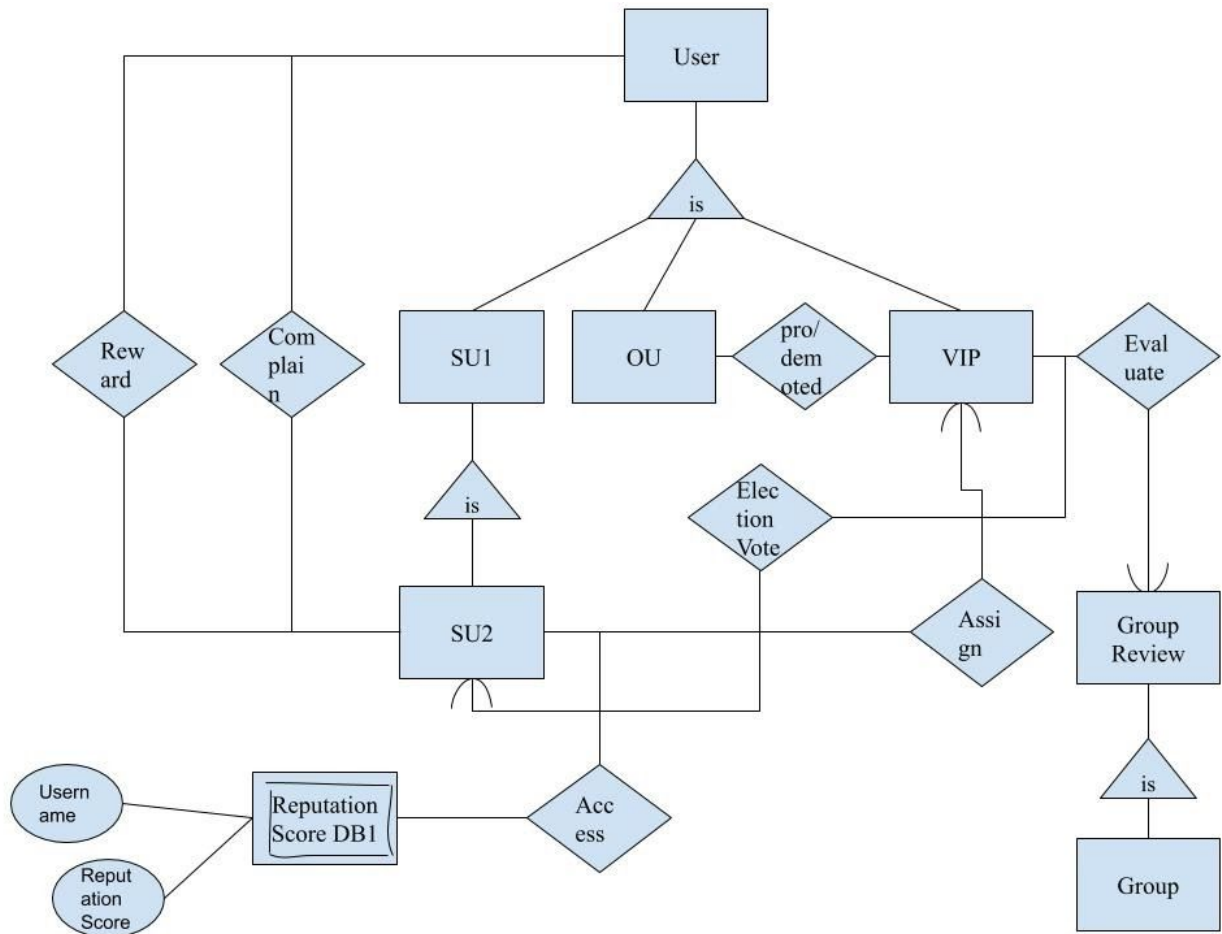




3. E-R Diagram for the entire system







4. Detailed Design

```
Function access_system (){
    As visitor:
        Display guest mode page;
    As user:
        Display login page;
}
Function sign_up (){
    Require filling up information;
    If information got approved by superuser, generated new account and sent message to
applicant;
    If rejected, sent message to applicant;
}
Function log_in (){
    Get account information;
    If account information match with database, then jump to home page;
    If information invalid, then go back to login page;
}
Function upload_project (){
    Accept uploading pdf project to home page;
    If superuser approved, project posted in home page;
    If disapproved, sent email to user;
}
Function project_search (){
    Based on user input, do binary search;
    If found, display all results match or contain keywords;
    If not found, show no found message;
}
Function password_reset (){
    Take user input for old password;
    If match, ask user enter new password;
    Save new password into database;
```

```

        If not match, clear input, ask to enter again;
    }
    Function add_friend (){
        If B is in A's BlackList, display message "Fail"
        If user B isn't in user A friends' list, then add user B into A's list;
        Sort list use binary heap;
        Update new list in database;
        If B already in A's list; display message "B already is friend";
    }
    Function delete_friend (){
        If user B is in user A friends' list, then delete user B from A's list;
        Sort list use binary heap;
        Update new list in database;
        Else B doesn't exist in A's list; display message "B is not a friend";
    }
    Function detect_taboo_word (){
        Take user input;
        If detect taboo word in string; fail to send email or message to other user;
        If detect taboo word once; deduct 1 from user score;
        Else detect taboo word more than once; deduct 5 from user score;
        Replace taboo words with "**";
    }
    Function make_complaint (){
        Take user input for complaint;
        If get approved by superuser, then send back a email referring decision;
        Complaint for application: superuser will send message and fix it;
        Complaint for user: will deduct 5 points from target user scores;
    }
    Function send_compliment (){
        Take user input for compliment;
        If more than 5 users or VIPs agreed, send to superuser;
        If get approved by superuser, then send back a email referring decision;
    }

```

```

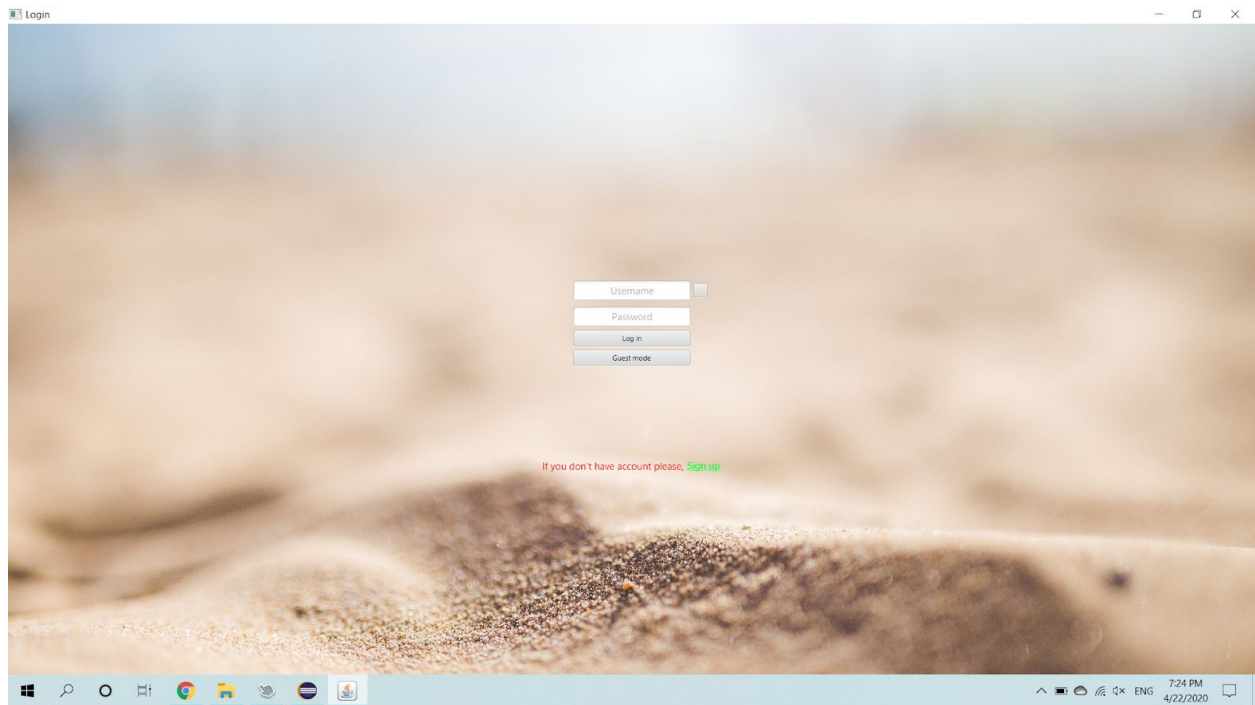
        If target user doesn't reach max number of compliments;
        Compliment for user: will add 0 - 5 points to target user scores based on
situation;
    }
    Function vote_for_SU () {
        If user is VIP;
            If one VIP has the highest vote count;
                receive a congrats message;
                adjust account type to superuser;
            else if two or more VIP have same highest vote;
                call vote_for_SU function again;
    }
    Function demotion () {
        If user is VIP && reputation score < 25;
            Send demotion message;
            Demote VIP to normal user;
    }
    Function promotion () {
        If user is normal user && reputation score > 30;
            Send promotion message;
            Promote normal user to VIP;
    }
    Function grouping () {
        If target user currently isn't in any group,
            If user in white_box or manually accepted; add into project member list;
            Else user in blacklist or manually rejected; send message to inviter;
    }
    Function out_of_group () {
        If vote_out count over half of group size;
            Delete user from project member list;
            User reputation score deduct by 10;
        Else if 3 warnings;

```

```
        Delete user from project member list;
        User reputation score deduct by 5;
    Else kicked by superuser;
        Delete user from project member list;
    }
Function Creat a Voting (){
    Insert all available vote
    Display them
}
Function follow a voting(){
    Display all available votes
    Pick one of them
    Total vote of that one +1}
```


5. System Screens

Login Page0



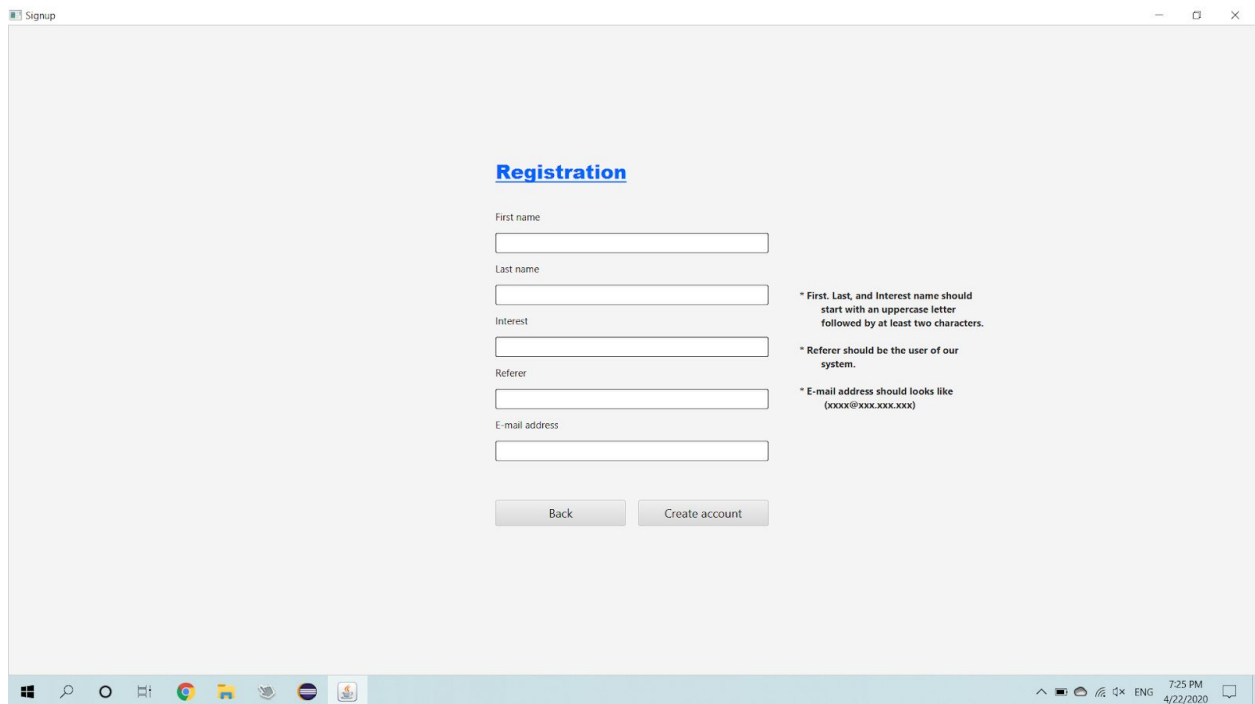
The screenshot shows a web browser window titled "Login". The background is a blurred image of sand dunes. In the center, there is a login form with the following fields and buttons:

- Username:
- Password:
- Log in:
- Guest mode:

Below the form, there is a red text prompt: "If you don't have account please, [Sign up](#)".

The Windows taskbar at the bottom shows the time as 7:24 PM on 4/22/2020.

Sign up page



The screenshot shows a web browser window titled "Signup". The background is a solid light gray. The main heading is "Registration" in blue. Below it, there are five input fields for registration:

- First name:
- Last name:
- Interest:
- Referer:
- E-mail address:

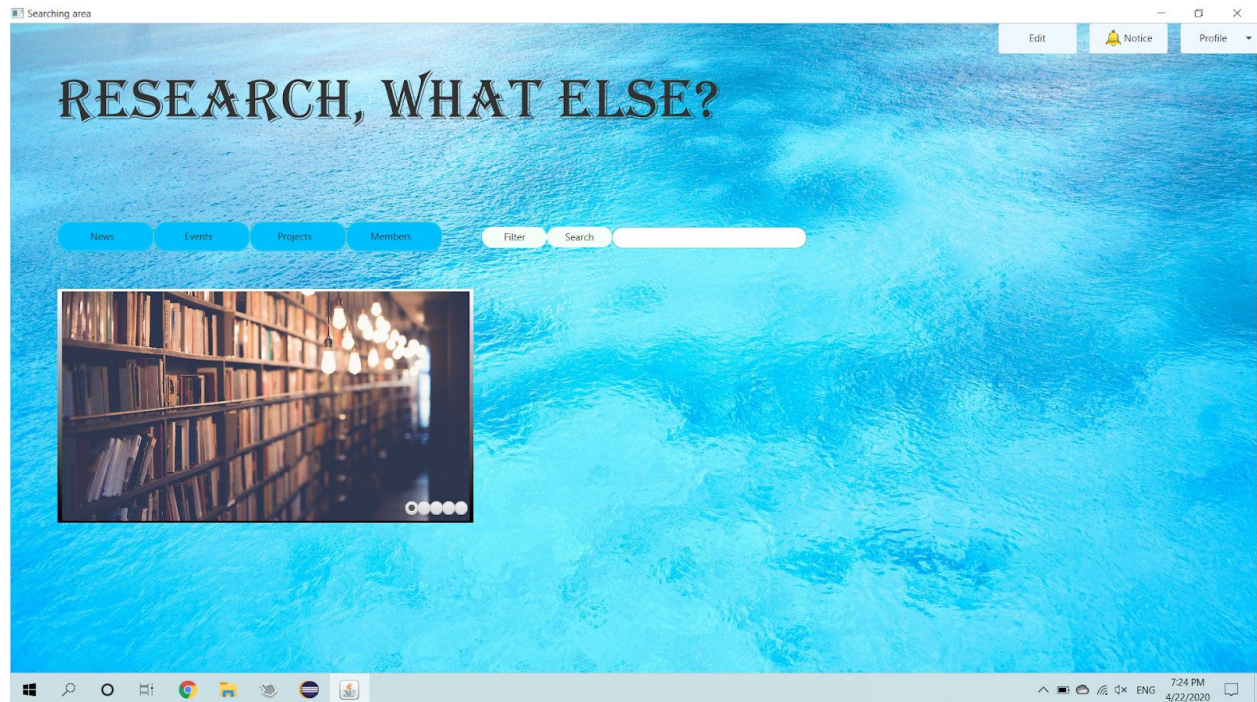
To the right of the input fields, there are three asterisked notes:

- * First, Last, and Interest name should start with an uppercase letter followed by at least two characters.
- * Referer should be the user of our system.
- * E-mail address should looks like (xxxx@xxx.xxx.xxx)

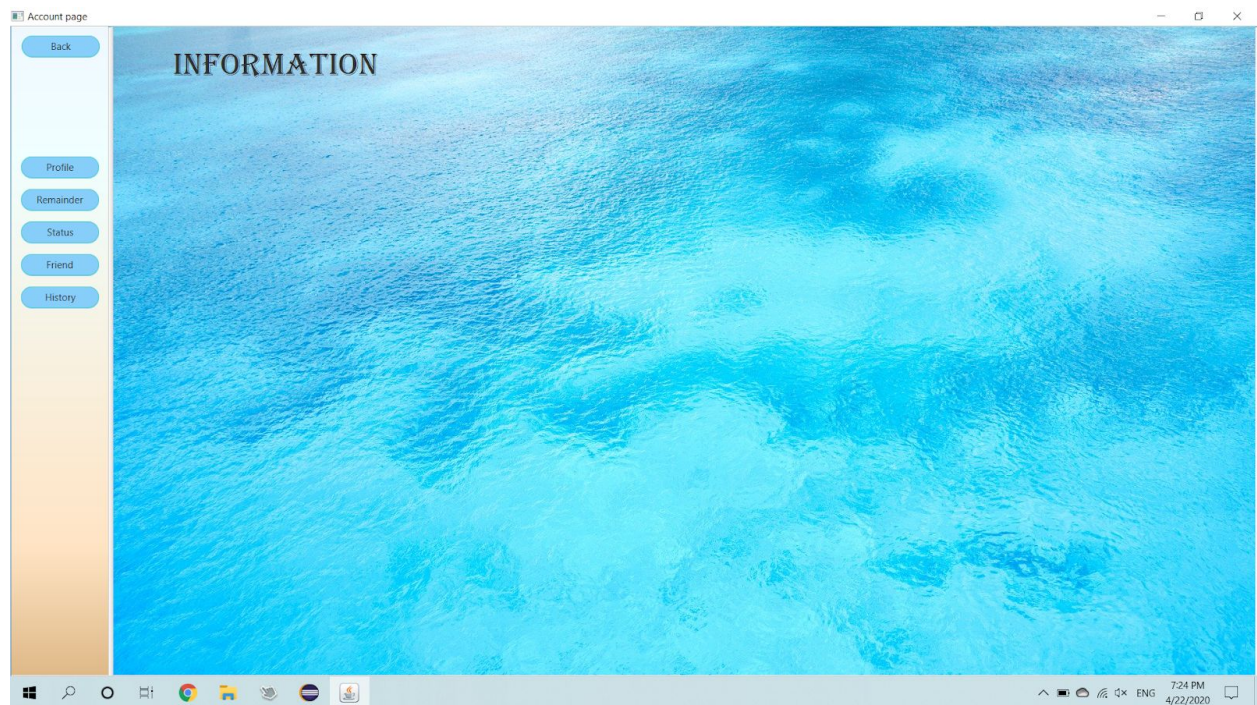
At the bottom of the form, there are two buttons: "Back" and "Create account".

The Windows taskbar at the bottom shows the time as 7:25 PM on 4/22/2020.

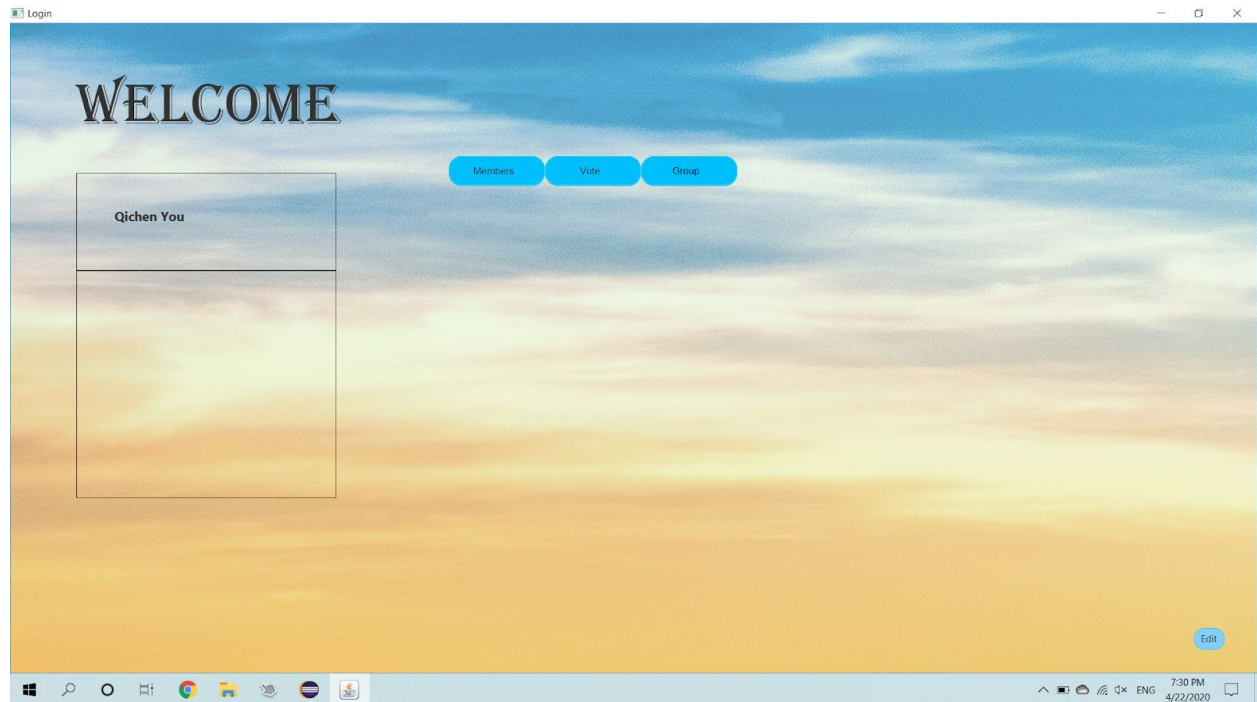
Homepage



Account page



Group page



6.

1. Generally, we have a half to an hour for our meeting.
2. All meetings were made occasionally.
3. Every member has a different attitude to the project so it turns out that some of the members will spend more time on the project than other members.
4. It is maybe hard for some team members to understand the code that was written by other members.

7.

<https://github.com/XinChenCSC/Active-Teaming-System.git>