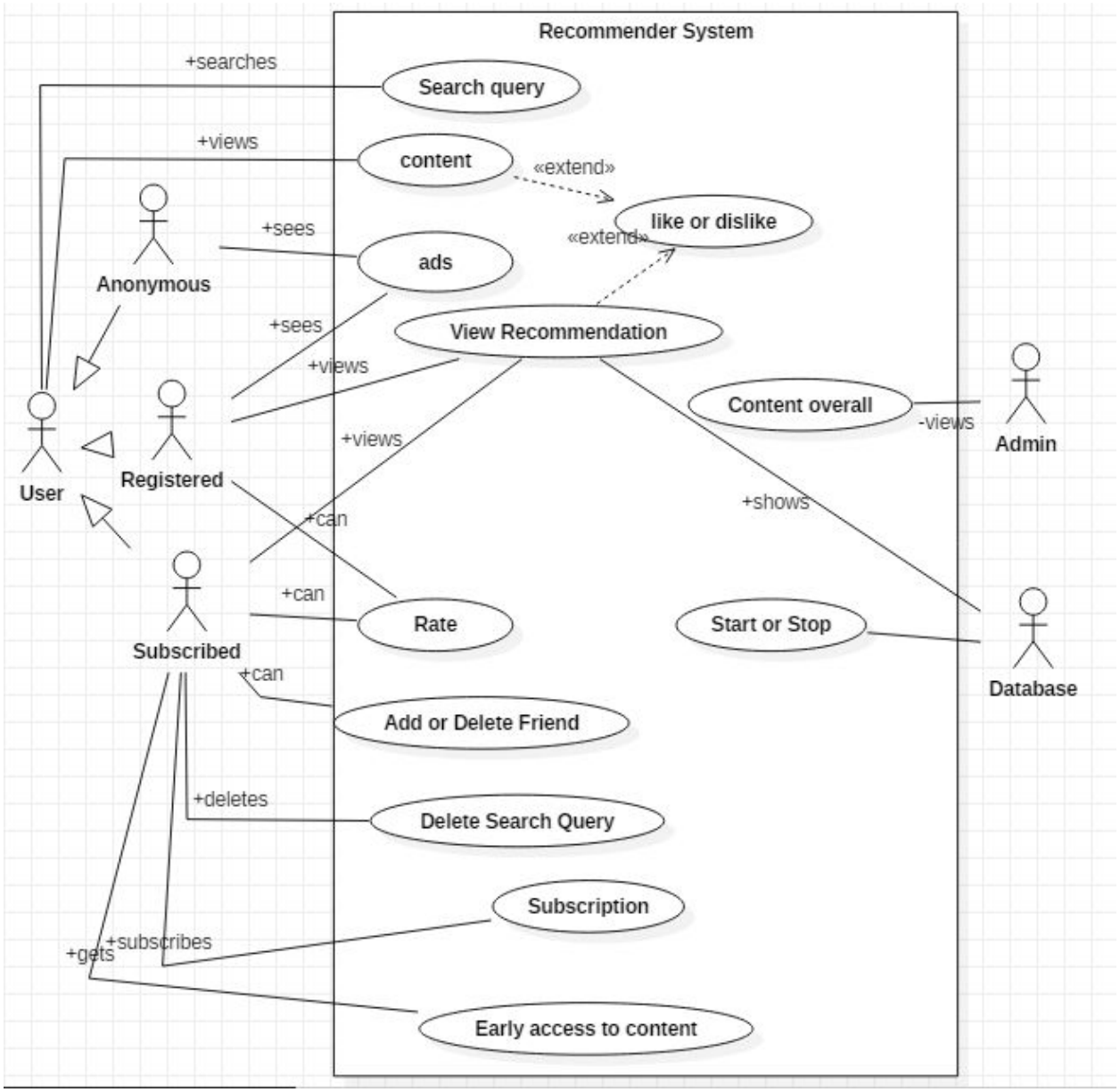
Recommendation System
Use Case Diagram and Use case description

Naveen Suresh - 01FB16ECS222
Neelesh C. A. - 01FB16ECS223
Ninaad R Rao - 01FB16ECS232

Use case diagram for recommender system:



Use case description for recommender system:

1.

Name : Search query
Summary: User searches for a query. If the user was a registered or a subscribed user, then the query helps in giving recommendation.
Actor: User
Precondition: None.
Main path: User clicks on the search bar. User types the content he wants to search. He clicks on the submit button.
Description: The user types the query he wants to search for. He clicks on the submit button and after submitting he gets the desired results from the database that stores all the relevant results. After this, if it is an anonymous user, then recommendation happens per session. Else, if the user is registered or subscribed user, then the search query helps in giving more fine tuned recommendation in the future.
Exception: If the query results are not present in the database. An invalid message is displayed
Post condition: In case of query text not present in the database, the user is still in the query page and he can search some alternative.

2.

Name : Content
Summary : User views content. The content is based on the recommendation that is got from previous search queries or likes/dislike information.
Actor: User
Precondition: None
Main path : The content is displayed on the screen.
Description: The user views content from the system and appropriate results are displayed for him as and when required.
Exception: None
Post condition: User views better recommendation later on.

3.

Name : ads
Summary : A user who is not subscribed sees ads in the page.
Actor: Anonymous user, Registered user.

Precondition: None
Main path : The user who is not a subscribed user sees ads on the right pane or the bottom.
Description: The anonymous user and registered user see ads in the display page since they are not subscribed user. The ads are relevant to their search queries.
Exception: If the user is subscribed user.
Post condition: In case of subscribed user, he can report that he is seeing an ad and appropriate correction could be done immediately by the system. In case of registered and anonymous users, they continue to see ads and if they don't want to see ads, then they can get an option to become subscribed users.

4.

Name : View recommendation
Summary : a user who is not anonymous user gets to view history specific recommendation.
Actor : Registered user, Subscribed user, database
Precondition : the user needs to be logged in/ should sign up if anonymous.
Main path : The user clicks on view recommendation. A set of recommendation is displayed based on past data queried on the database.
Description : the registered user and subscribed user get to see recommendation based on their liked or disliked content in the past. They can even change their liked or disliked content and view modified recommendation
The database displays appropriate recommendation for the queries that are received. If a particular content is absent, then a blank page needs to be displayed
Exception : If the user is not logged in.
If the database is not able to recommend to the query.
Post condition : if the user modifies the database based on his likes and dislikes, then appropriate changes need to be reflected thereon.
If the database is not able to recommend to the query, then appropriate message should be sent by the server to display it to the user. Database might not be able to recommend for two reasons. Either the database is undergoing maintenance or the search query is something the database was not able to handle.

5.

Name : Rate
Summary : a registered or subscribed user can rate the content accordingly
Actor : Registered user, subscribed user
Precondition : the user needs to be logged in / should sign up if its new user to view this option.
Main path : User who is not an anonymous user can rate a content . The user clicks on the rate button. If he is registered or subscribed user, then a text box appears where he can enter a rating on a scale of 1 to 5.

Description : the registered or subscribed user can rate a particular content based on his views.
Exception : if the user views this option when he hasn't signed in.
Post condition : This rating information needs to be stored in the database if the user has signed in and needs to help in the recommendation system in a more specific manner.

6.

Name : Delete Search Query
Summary : helps in deleting a search query
Actor : subscribed user
Precondition : the user needs to be a subscribed member.
Main path: The user clicks on the delete search query. The search query is deleted from the database and the recommendation is modified accordingly.
Description : The subscribed user can delete a search query if he does not want the search to modify the recommendation.
Exception : If the user is not subscribed user.
Post condition : if the user is not subscribed user and clicks on this option, then he could be redirected to subscription page to subscribe to the page.

7.

Name : Subscription
Summary : helps user renew his subscription
Actor : subscribed user
Preconditon : the user needs to be a subscribed user with less than 1 month left in subscription time.
Main path: the user clicks on subscription button. If his subscription is due, then he is sent to the payment screen. If his subscription is not due, then an error page is displayed.
Description : when the user clicks the subscription link, he can renew his subscription to the system and continue to enjoy the desired benefits.
Exception : None.
Post condition : A message is shown telling that he has succesfully renewed subscription.

8.

Name : Content overall
Summary : view the overall content shared in the database
Actor : admin
Precondition : Needs to be a privileged user/admin and should have already logged in.

Main path : admin clicks on the content overall. He sees if there was any offensive content that was shared. If so, he can delete the same from the database and block the user from using the system.
Description : The admin can view all the contents being shared in the recommendation system . He can take action against users sharing offensive content and can either temporarily/permanently ban the user.
Exception : if the user is not an admin
Post condition : If the user is not an admin and clicks on this button, then he can be given an error exception.

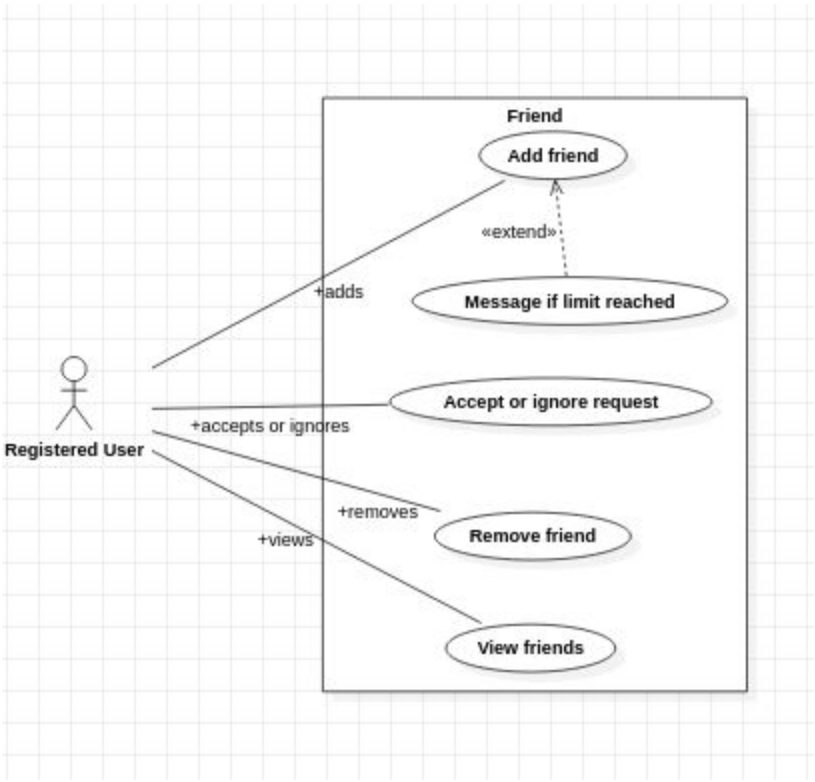
9.

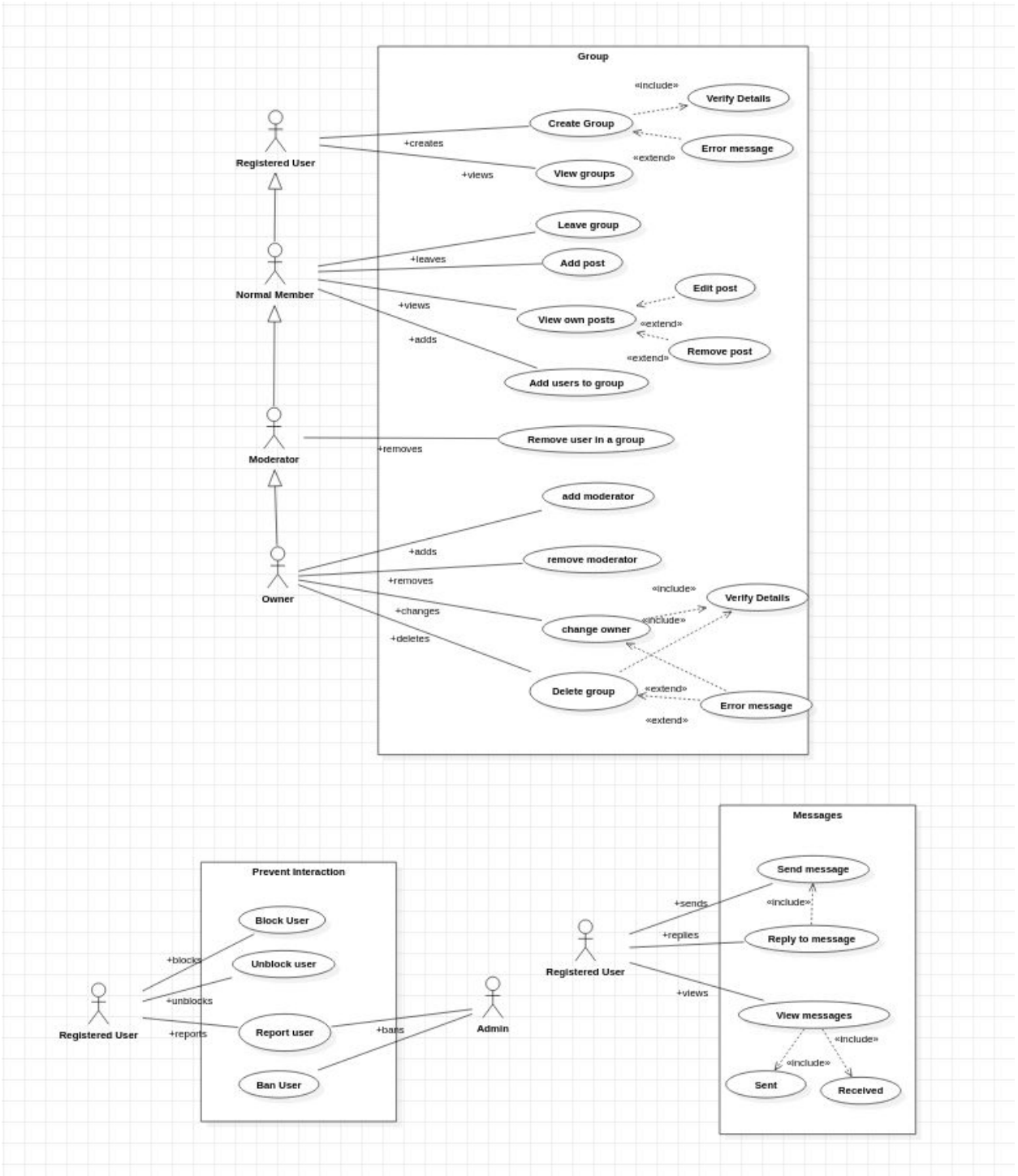
Name : Start or Stop
Summary : State of the database at a particular time
Actor : database
Precondition : None
Main path : Database is stopped during maintenance . It is started after completion of maintenance.
Description : This indicates the state of the database. If the database system is under maintenance then the state of the database is stopped and restarted later on.
Exception : If a particular user queries during maintenance.
Post condition : A page displaying the reason for outage be shown to all the users querying during this time.

10.

Name : Recommend
Summary : Recommending the content to a particular user
Actor : database
Precondition: The user has requested for a recommendation
Main path : the database services the recommendation by running its logic and appropriate recommendation content is shared with the user.
Description : The database needs to have the logic for recommending the content for a particular user. the database needs to modify as and when the recommendation is fine tuned
Exception : If a particular user does not get recommendation.
Post condition : if a particular user does not get recommendation, then appropriate message needs to be sent a message to modify the entire recommendation logic.

Use case diagram for friend functionality


Friend
Add friend
«extend»
Message if limit reached
+adds
Registered User
Accept or ignore request
+accepts or ignores
+removes
Remove friend
+views
View friends



Use case description:

1.

Name: Add friend

Summary: Sends a friend request to another user. If the other user accepts, he is added to the senders friend list. The friend can now be used by the recommendation engine for further recommendations. If the sender is blocked or reported by the receiver, no one gets a notificaiton.
Actor: Registered User.
Precondition: The user should be logged in. He should be viewing the profile of the person he wants to add. He shouldn't be friends with the user already. There shouldn't be a pending request from either side.
Main Path: The user clicks the add friend button.
A friend request is sent.
The receiver sees this friend request.
The receiver can choose to accept it or ignore it.
If it is accepted, both the users are friends to each other,
and appear on each others friend lists.
Exception: The receiver ignores the request.
The receiver has blocked/reported the sender.
The sender/receiver has reached the maximum limit of friends(the maximum limit is more for subscribed users)
Postcondition: In case of the main path, the database is updated that they are both friends. They appear on each other's friend list. If the receiver ignores the request, no notification is sent. The sender can still send a request again. If the receiver has blocked/reported the sender, no one gets a notification. The sender won't be able to send a request again. If the maximum number of friends is reached, the sender is sent a notification that this is this case.

2.
Name: Remove friend.
Summary: Removes a friend of a user. They will no longer appear on either user's friend list or be considered in the recommendation which considers only friends.
Actor: Registered User.
Precondition: The user should be logged in.
The user should be friends with the user he wants to remove.
He should be viewing the profile of the person he wants to remove.
Main Path: The user clicks the remove friend button. The friend is removed.
Exception: No exceptions
Postcondition: The database is updated that they are not friends anymore. The users aren't shown in each others friend list.

3.
Name: View friends.
Summary: The friends of a particular user are shown to him.

Actor: Registered User.
Precondition: The user should be logged in.
He should be in his profile
Main Path: The user clicks the view friends button.
The user is shown with a list of his friends in alphabetical order.
Exception: No exception.
Postcondition: The user is shown a list of his friends.

4.

Name: Send message.
Summary: Sends a message to a particular user. If the receiver has blocked/reported the sender, the receiver doesn't get the message.
Actor: Registered User.
Precondition: The user should be logged in. He should be viewing the profile of the person he wants to message.
Main Path: The user clicks the send message button.
He is shown a window where he can type the message.
The user types in the message.
He clicks the send button.
Exception: The receiver has blocked/reported the user.
Postcondition: In the main path, the message is sent to the receiver and he can view it.
The database is updated to reflect this.
In the exception, from the senders point of view,
it looks like he has sent the message successfully.
But the receiver doesn't know about it.

5.

Name: View messages
Summary: The user is shown the messages sent and received(Something like the gmail inbox and sent tabs)
Actor: Registered User,
Precondition: The user must be logged in. He should be viewing his profile.
Main Path: The user clicks the view messages button.
The user is shown the messages in chronological order.
Exception: No exception.
Postcondition: The user is shown a list of messages in chronological order. He has an option to view only sent/received messages.

6.

Name: Block User.
Summary: The user can block another user. He will not receive any messages/friend requests from the blocked user.
Actor: Registered User.
Precondition: The user must be logged in.
The user must be viewing the profile of the user he wants to block.
The user must not be blocked already.
Main Path: He clicks the block user button.
The user is blocked.
Exception: No exception.
Postcondition: The database is updated to reflect that the user has been blocked.
The user who is blocked can no longer send any messages/friend requests to the user who blocked him.
The user will not have an option to block the user again.
He will be given an option to unblock the user.

7.
Name: Report User.
Summary: This is used for a more serious offence, i.e. in cases of cyber bullying.
An admin will be notified if a user is reported, and he can take appropriate action.
Actor: Registered User.
Precondition: The user must be logged in.
The user must be viewing the profile of the user he wants to report.
The user must not be reported already.
Main Path: The user clicks the report button.
He is shown a window where he types out the reason for reporting the user.
He is shown a message asking if he's sure that he wants to report the user.
He clicks Ok.
The user is reported.
Exception: The user doesn't give any reason for reporting the other user.
The user clicks 'no' when asked if he's sure to report the user.
Postcondition: In the main path, the user is reported, one of the admins get a notification.
The database is updated to reflect the same.
If the user doesn't give a reason, he is not allowed the submit the report. The field is marked as required. If the user clicks 'no', the report box is closed, and he is shown the previous screen

8.
Name: Create Group.

Summary: A user can create a group where he invites people with similar interests.
This can be used a discussion forum, where people can talk about recommendations for users in the group, new releases etc.
Actor: Registered User.
Precondition: The user should be logged in.
He should be on the 'groups' page.
Main Path: The user clicks the create group button.
He is shown a window where he has to fill details like name of the group, category, genre's etc.
He enters the details.
A group is created with him as the owner of the group.
Exception: A group already exists with the same name.
The name of the group or category is left empty.
Postcondition: In the main path, a group is created, this information is added to the database.
The user who created the group is the owner of the group and has more privileges within the group.
If the group name already exists, he is shown an error message.
He remains in the same window.
If the group name/category is empty, he is shown an error message.
He remains in the same window.

9.

Name: Post on group
Summary: The user can add a post to the group which they are a member of.
Actor: Normal member of group.
Precondition: The user must be a member of the group.
The user must be logged in.
The user must be be in the group page.
Main Path: The user clicks on the add post button.
The user is shown a window where he can type a title and the body.
The user enters the title and the body.
The user clicks post.
Exception: Any field is empty.
Postcondition: A post is added to the group which can be viewed by all members of the group.
The information related to the post is added to the database.
This post now shows up on the view own posts of the user.
If any field is empty, he is shown an error message and he stays on the same page.

10.

Name: View own posts.
Summary: The user views the posts which he has posted himself. He can edit/remove these posts.
Actor: Normal member of the group.
Precondition: The user must be a member of the group.
The user must be logged in.
The user must be be in the group page.
Main Path: The user clicks on the view posts button. The user is shown all/some of the posts depending on the screen resolution, in chronological order.
The user can edit/delete these posts.
In case of edit, he is shown the message body, which he can edit and save.
In case of delete, the post is removed.
Exception: No exception.
Postcondition: The user is shown his posts.
If the user edits a post, that post is edited and everyone sees the edited post.
The database is updated with this.
If the user deletes a post, that post is deleted and no one can see the post anymore.
The database is updated with this.

11.
Name: Change owner.
Summary: This is used in case the owner of the group can't perform his duties anymore. He an assign a new owner.
Actor: Owner of the group.
Precondition: The user must be an owner of the group.
The user must be logged in.
The user must be be in the group settings page.
Main Path: The user clicks on the change owner button.
A message pops up asking if he's sure he wants to change owner.
The user clicks ok.
He is given an input field where he can enter the username of the new owner.
He enters the username and clicks change owner button.
He is given another message asking if he's sure.
He clicks ok.
Exception: The username does not exist.
He clicks no during one of the message prompts.
Postcondition: In the main path, the owner is changed. The new owner gets a message.
The old owner becomes a normal member of the group.
If the username does not exist, a message is shown and he remains on the same screen.

If he clicks no in either of the message prompts, he is taken the group home page.

12.

Name: Delete group.

Summary: This is used to delete the group. No one can view the group anymore except the admin.

Actor: Owner of the group.

Precondition: The user must be an owner of the group.

The user must be logged in.

The user must be in the group settings page.

Main Path: The user clicks delete group button.

He is shown a message asking if he's sure that he wants to delete the group and that the action is irreversible.

He presses ok.

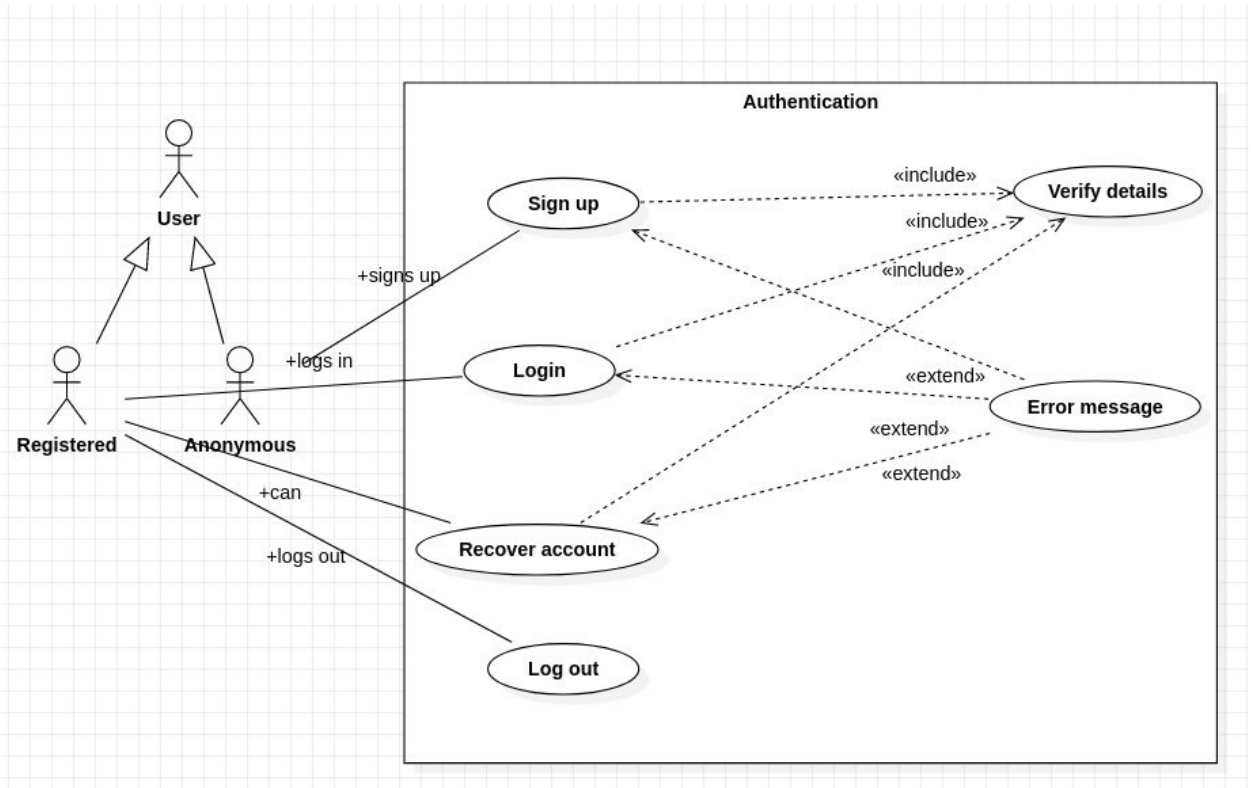
He is asked to enter the name of the group to confirm.

He clicks ok.

Exception: He cancels.

He enters wrong name.

Postcondition: In the main case, the group and everything related to it are deleted. The group no longer appears anywhere. Another group can be made with the deleted groups name. All the posts are deleted. The database is updated. If he cancels, he is redirected to the settings page. If he enters the wrong name, an error message is shown and he remains in the same window.



Name: Sign Up

Summary: Registers a user to the website. Adds their information to the database.
The user can login to the website as a registered user with those credentials.

Actor: Anonymous User.

Precondition: The user shouldn't be logged in already.

Main Path: The user clicks the sign up button.
He enters his username, email and password.
He confirms that his details are correct.
If all the details are valid, a link is sent to his email.
After clicking the link, a success screen is shown.
The user can now sign in with these credentials.

Exception: The details are invalid, e.g. the username/email is already in use, or the password doesn't meet the criteria.
A message is shown telling what is invalid.

Postcondition: In case of success, the user is redirected to the home screen.
The details are entered into the database.
In case of invalid details, the user is still in the sign up screen

Name: Login
Summary: A user who has entered valid details is logged on to the website.
Actor: Registered User.
Precondition: The user shouldn't be logged in already.
Main Path: The user clicks the login button.
He enters his email and password.
If the details are correct, he is logged onto the website.
Exception: Incorrect email/password. An error message is shown.
Postcondition: In case of success, the user is logged in.
In the case of the exception, the user is in the same page.

Name: Recover account
Summary: A user will be given a password recovery link to their email.
Actor: Registered User.
Precondition: The user shouldn't be logged in already.
Main Path: The user clicks the recover account button.
The user is asked his email.
If the email exists, an recovery link is sent to that email.
If it doesn't an error message is shown.
Exception: Email doesn't exist.
Postcondition: In case of success, a recovery link is sent to the email.
In the case of the exception, the user is in the same page.

Name: Log out
Summary: A user who is logged in is logged out.
Actor: Registered User.
Precondition: The user must be logged in.
Main Path: The user clicks the sign out button.
The user is logged out.
Exception: None.
Postcondition: The user is signed out.