# 1 New Member Creates Account

Use Case: New member creates account on *Members Only.*

**CHARACTERISTIC INFORMATION**

Success End Condition: Account successfully created.

Failed End Condition: Account not created.

Primary Actor: Potential New Member

Trigger: Current member sends new member invitation link to join *Members Only*.

MAIN SUCCESS SCENARIO

Josh gets sent a link from a friend to join Members Only. Naturally, Josh is very excited and is eager to join the site. Josh finds the link in an email from Members Only and selects it. He is now brought to a Members Only webpage that asks him to confirm if he wants to join Members Only. Josh will of course select yes on joining Members Only. After accepting, he will be brought to a *Sign Up* page on the website. Josh will have to fill out a form will personal information such as First Name, Last Name, Street Address, City/Town, Zip Code, Country (U.S. only to start), Email Address, and a Password. Josh will then select a box which says that he accepts the *Terms and Conditions* of Members Only. Finally, Josh will submit this form, and he will be sent a confirmation email about joining Members Only. Josh will go to this email, select the account confirmation link, effectively finalizing the creation of his account and bringing him to his own Members Only account. Afterwards, Josh will be sent a credit card charge that he will have to verify on the site to confirm that he says he is.

SIGNING UP WITH A CODE

Brianna manages to get her hands on an invitation code to Members Only from one of her favorite idols. To use this, Brianna navigates to the Members Only site and is brought to the login page. Since Brianna doesn’t currently have an account, she will have to choose the option to sign up with a code. After clicking this, Brianna will be brought to a new web page where she will be asked to enter her code. Brianna will enter the code and then hit a submit button. Afterwards, she will be brought through the same account creation process as described previously.

# 2 New Member Invitation Link Expires

Use Case: New Member does not want to join *Members Only* or forgets about invitation, invitation expires.

**CHARACTERISTIC INFORMATION**

Success End Condition: Invitation expires.

Failed End Condition: Invitation is accepted/rejected.

Primary Actor: Potential New Member

Trigger: Current member sends potential new member invitation link to join *Members Only*.

MAIN SUCCESS SCENARIO

Liam receives an invitation link to Members Only. However, he is very busy and forgets to about it for more than 4 days. Because of this, the email link will expire, and he will lose this opportunity to join the site. To be able to join again he will have to be sent another invitation link.

# 3 New Member Confirms Credit Card Charges

Use Case: New member confirms credit card payment details.

**CHARACTERISTIC INFORMATION**

Success End Condition: Confirmation of credit card details.

Failed End Condition: Credit card details unconfirmed.

Primary Actor: New Member

Trigger: New member creates account on *Members Only.*  [SEE USE CASE# 1]

MAIN SUCCESS SCENARIO

After successfully creating her account, Catherine will have to enter her credit card details so that we can confirm she is who she says she is. Catherine will receive an email to confirm her payment details. Catherine will select this link which will bring her to a form on the Members Only site where she will enter a valid 16-digit credit card number, the name the card is under, the expiration date of the card, and the 3 digit CVV associated with the credit. Catherine will then choose to SUBMIT this form. She will then be brought back to her Members Only page.

# 4 New Member Confirms Credit Card Charge

Use Case: New member confirms initial credit card charge.

**CHARACTERISTIC INFORMATION**

Success End Condition: Confirmation of credit card charge value, allows full access to *Members Only.*

Failed End Condition: Non-confirmation of credit card charge value, no full access to *Members Only.*

Primary Actor: New Member

Trigger: New member confirms credit card details [SEE USE CASE# 3]

MAIN SUCCESS SCENARIO

After Catherine successfully enters her credit card information, she will be charged a random amount between $0.20 and $0.45 that she will need to confirm on the Members Only site to verify she is who she says she is. Catherine will receive an email shortly after creating her account that notifies her that she has been charged for the first time. Catherine will then navigate to her online banking and check for a charge from Members Only. After finding the charge, Catherine will remember the amount and navigate back to notification email from Members Only. Catherine will select the hyperlink in the email that brings her to a form with an entry field for the value of the charge. Catherine will then enter the charge amount and select SUBMIT. If she submits the correct amount then she is brought back to her Members Only page as a verified Member. Else, she will be asked to resubmit the amount she was charged until she submits the right amount.

# 5 Member Logs in to Members Only

Use Case: Member logs into *Members Only*.

**CHARACTERISTIC INFORMATION**

Success End Condition: Member successfully logs in.

Failed End Condition: Member does not successfully log in.

Primary Actor: Member

Trigger: Member wants to log in to *Members Only*.

MAIN SUCCESS SCENARIO

Johnson wants to login in to Members Only and check his feed. To do this, Johnson will navigate to the Members Only homepage, which includes a login for existing users form. Johnson will enter his email address and password associated with his Members Only account. Johnson will then select LOGIN at the bottom of the form. If his email and password are correct, he will then be redirected to his Members Only feed.

MEMBER FORGOT PASSWORD / INPUTS INVALID PASSWORD

Johnson wants to login to Members Only, but he is stuck at the login form on the homepage of the site because he has forgotten the correct password associated with his account. Johnson will then select a link that says, “Forgot Password”. This will redirect him to a different form on the Members Only site where he will fill out his email and several other details associated with his account such as his name. Johnson will then be sent an email that sends him a new temporary password to login to his account. Johnson can then reset his password once he logs back into his account and accesses his settings.

MEMBER FORGOT EMAIL / INPUTS INVALID EMAIL

Johnson directs himself to the Members Only homepage and tries to login to his account. However, Johnson has somehow forgotten the email associated with his Members Only account. After entering his information, Johnson will click submit. The system will receive the email and attempt to verify who Johnson is, however because it is the incorrect email Johnson will be denied access to Members Only until he can recall the correct email address associated with his account.

# 6 Member Changes Password

Use Case: Member changes password.

**CHARACTERISTIC INFORMATION**

Success End Condition: Member successfully changes password.

Failed End Condition: Member unsuccessfully changes password.

Primary Actor: Member

Trigger: Member not happy with current password or member wants additional security.

MAIN SUCCESS SCENARIO

Larry wants to change his password, either for security reasons or because he just logged in with a temporary password because he forgot his previous one. Larry will navigate to the SETTINGS portion of his account after immediately logging in. Larry will then navigate and choose the CHANGE PASSWORD option. Larry will be brought to a separate, private form where he will be prompted to enter a new password. After entering the new password, Larry will be prompted to enter it a second time as to confirm the password and eliminate the possibility of any typos. Larry will then click CONFIRM and the system will verify that the passwords are identical. If they are, then Larry will be redirected back to his Members Only feed. Otherwise, Larry will be prompted to enter and reenter the new password again until he successfully enters the same password twice.

# 7 Member Changes Visibility Settings

Use Case: Member changes visibility setting to private.

**CHARACTERISTIC INFORMATION**

Success End Condition: Account successfully changed to private.

Failed End Condition: Account unsuccessfully changed to private.

Primary Actor: Member

Trigger: Member prefers more privacy for various reasons.

MEMBER DOESN’T WANT NON-FOLLOWERS SEEING THEIR PERSONAL INFORMATION

Danny wants to hide his activity from non-followers on Members Only. To do this, Danny will go to his personal page and access the SETTINGS portion of the page. Danny will then navigate to the toggle button which allows them to switch between a privatized and un-privatized profile. Since Danny wants to privatize his page, he will select this toggle button to the on setting, effectively hiding his activity from non-followers.

# 8 Member Blocks/Reports Another Member

Use Case: Member wants to report another member for illicit behavior.

**CHARACTERISTIC INFORMATION**

Success End Condition: Other Member blocked or reported.

Failed End Condition: Other Member receives no penalty.

Primary Actor: Member

Trigger: Member is triggered by behavior of another Member.

BLOCK A FOLLOWER

Syed wants to block one of his followers that ruining his experience on Members Only. Syed will go to their Members Only personal page. Syed will then go to their followers list and navigate to the person they are concerned about. Syed will then choose to block this follower, which will disallow them from seeing any of Syed’s information, status updates, posted pictures, or any of Syed’s activities on the site.

REPORT MEMBER

Phil wants to report another member on Members Only for an offensive action. Phil will go to his personal page and access his followers list, navigating to the person he is concerned with. Phil will then choose the option to report this follower. Phil will then be redirected to a report issue page that contains a form where he will enter the member’s name he is concerned with and other details about why they are being reported. After filling out the form, Phil will click SUBMIT, sending the form to the Web Server. This information will then be redirected to Admins so that they can review the report and decide any further action.

Member updates credit card information

**CHARACTERISTIC INFORMATION**

Success End Condition: Credit card information successfully updated.

Failed End Condition: Credit card information unsuccessfully updated.

Primary Actor: Alex (member)

Trigger: Old credit card canceled/expired, or want to update to new credit card.

CREDIT CARD DECLINED/EXPIRED/CANCELLED

1. Alex will login to their profile

2. Alex will navigate to their settings page on their personal profile

3. Member will choose change credit card button

4. System will prompt user with text boxes to enter new credit card information

4a. User will enter name of card

4b. User will enter card number

4c. User will enter expiration date

4d. User will enter CVV

5. Member will hit the submit form button

6. Credit card will then be verified to make sure it is valid

7. System will send this to the database to update the current credit card information with this information

8. credit card information updates

Member Confirms Tri-Monthly Credit Card Charge

Use Case: Member confirms tri-monthly credit card billing.

**CHARACTERISTIC INFORMATION**

Success End Condition: confirm credit card charge

Failed End Condition: Credit card inactive

Primary Actor: Alex (Member)

Trigger: Member need to do their routine credit card charge check

MAIN SUCCESS SCENARIO

1. System will tell the credit card software it is time to charge Alex

2. Credit card system will charge member a random amount between $0.20 and $0.40.

3. System will send automated email at specified time period telling member to validate the charge made to their account

3a. Email will include a link that will direct the user to the page to enter the amount charged

4. System will start a timer for 3 days

5. Alex will login to their personal bank account and check for the charge from Members Only

6. Alex will click link provided in email

7. Alex will be redirected to the webpage with a form where they can enter the amount they were charged by the system

8. Alex will fill in the text box with that amount

8a. Alex will hit the “Submit form” button

9. Webserver will receive this information and back end services will process it

9a. System will compare this amount to the amount that was charged

9b. If right, member will be directed back to the Members Only home page

9c. If wrong, member will be blocked out of Members Only until they enter the right amount

AMOUNT NOT ENTERED IN TIME

1. Alex doesn’t enter amount in time and is blocked from account

2. When Alex tries to access Members Only, they will be directed to a page telling them to send another charge

3. System tells credit card system to send another charge between $0.20 and $0.40

4. Credits card system sends another random charge

5. System sends Alex another email

6. System starts a timer for 3 days

7. Alex logs into their bank account and sees the charge from Members Only

8. Alex clicks button on email and is redirected to Members Only

9. Alex enters amount in and is either allowed back to website or entered the wrong amount

Member posts a status update on personal profile.

**CHARACTERISTIC INFORMATION**

Success End Condition: Members successfully updates status.

Failed End Condition: Members unsuccessfully updates status.

Primary Actor: Alex (Member)

Trigger: Alex wants to update followers on some current event.

MEMBER POSTS LINKS IN STATUS

1. Alex will click an option on their profile page that reads “create new post”

2. Alex will type URL, and can add any additional text to the post

2a. tinyURL generator will shorten the URL given

3. Alex will hit “post”

4. The webserver will receive the request send it to the backend

5. Backend will process the request and post Alex link to their profile feed

6. Backend services will send the link to the database to store it as something Alex is interested in for potential future advertisement reasons

Member leaves comment on friend’s status.

**CHARACTERISTIC INFORMATION**

Success End Condition: Comment successfully posted.

Failed End Condition: Comment unsuccessfully posted.

Primary Actor: Alex (Member)

Trigger: Alex sees friend’s post and they wants to reply to it.

MAIN SUCCESS SCENARIO

1. Alex will see their friend’s status on their profile feed

2. Alex will click the comment button that appears on the bottom of their friend’s post

3. The Webserver will get the request and pop up a text box overlay on the website

4. Alex will leave a comment, then hit “Post”

5. The Webserver will receive the request and send it to the backend services

6. Backend services will add it to the friend’s comment section on their post

Member invites a non-member to Members Only so that they can enjoy Members Only with them

**CHARACTERISTIC INFORMATION**

Success End Condition: Member sends invitation

Failed End Condition: Member does not send invitation

Primary Actor: Alex (Member/Idol)

Trigger: Alex wants to invite their friend

MAIN SUCCESS SCENARIO

1. Alex navigates to the “Invite New Member” button on their profile page

2. On this page, Alex will fill in new members email address

3. Webserver will receive this information and pass it to the backend

4. Backend will save new members email to database

5. Backend will tell email system to send a new invitation email

6. New invitation email will be sent to new member

6a. System will start a timer for 3 days

NEW MEMBER ACCEPTS OFFER

1. Alex will gain points

2. Webserver will receive this information and send it to the backend

3. Backend will add that many points to Alex’s point system

NEW MEMBER DECLINES OFFER/TIMER RUNS OUT

1. Alex loses points

2. Webserver will receive this information and send it to the backend

3. Backend will subtract that many points from Alex’s point system

NEW MEMBER SIGNES UP WITH SPECIAL CODE

1. Alex goes to Members Only website

2. Alex clicks “create account with Special Code” button

3. Alex is redirected to new page with a text box asking for the code

4. Alex types in that code

5. Code is processed by the webserver and sent to the back end

6. Back end compares that code to a code that was created for a special event (concert, large show)

7. Alex is redirected to a page to signs up (see above)

Member posts photo to profile.

**CHARACTERISTIC INFORMATION**

Success End Condition: Photo successfully posted to profile.

Failed End Condition: Photo failed to post

Primary Actor: Alex (Member)

MAIN SUCCESS SCENARIO

1. Alex will choose “Post Photo” on their profile feed.

2. Alex can navigate their device for the photo they wish to submit.

3. Alex can then add any additional text to the post that they wish.

4. Alex will then hit submit.

5. The Web Server will receive the request and send it to backend services.

6. Backend services will process and post the request.

6a. Backend will save image to database.

6b. Admins can use saved image for their own purposes.

Member removes/applies filter to photo.

**CHARACTERISTIC INFORMATION**

Success End Condition: Member changes image by adding or removing filter or content.

Failed End Condition: Photo remains the same.

Primary Actor: Alex (Member)

MAIN SUCCESS SCENARIO

1. Alex will go through the post a photo process, except immediately after they

choose their photo they will be given the option to apply a filter.

2. Alex will scroll through various premade filters available on Members Only.

3. Alex will choose a filter that will be applied to their photo.

4. The post photo process then continues from here.

REMOVING FILTER

1. Alex navigated to profile page and clicks on the photo they wish to remove the filter on.

2. Alex chooses “options”.

3. Alex chooses remove filters.

4. The filters will then be removed from the photo.

5. Alex will then be prompted to confirm their changes.

6. Alex will click “Confirm”.

7. Request will be sent to the web server.

8. Web server will send it to backend.

9. Backend services will replace that photo with the originally posted photo.

ADDING (SPONSORED) CONTENT

1. Alex will follow the posting photos process.

2. Alex will then click add content.

3. Alex can choose any of the items and place them anywhere in their photo using a drag and drop method.

4. Alex will choose to submit their altered photo.

5. The Web Server will receive the request and send it to backend services.

6. Backend services will process and post the request.

6a. Backend will save image to database.

6b. Admins can access image and add/delete content as they please.

REMOVING (SPONSORED) CONTENT

1. Alex will go to their photo with added content.

2. Alex will select options and be given an option to remove content from the photo.

3. Alex will be shown what is added content in their photo such that they can

navigate to it and choose to remove it.

4. Alex will submit their changes.

5. The Web Server will receive the request and send it to backend services.

6. Backend services will process and post the request.

6a. Backend services will save image to the database.

6b. Admins can use image to add/remove/etc.

7. Original poster will not be able to see these changes.

Use Case # 18

Use Case: Member replies to comment on their photo.

**CHARACTERISTIC INFORMATION**

Success End Condition: Member successfully replies to comment.

Failed End Condition: Comment fails to post.

Primary Actor: Member

MAIN SUCCESS SCENARIO

1. The Member’s photo will have a follower’s/friend’s comment on it that they want to

respond to.

2. The Member will navigate to the comments section under their posted photo and find

their friend’s comment.

3. The Member will choose “Reply to comment”.

4. The Member will click “Post reply”.

5. The Web Server will receive the request and send it to backend services.

6. The Web Server will receive the request and send it to backend services.

6a. Backend services will post it as a reply to a comment on the Member’s photo.

Use Case # 19

Use Case: Member flags photo that inappropriate/illegal.

**CHARACTERISTIC INFORMATION**

Success End Condition: Member flags image and it is removed.

Failed End Condition: Inappropriate image remains on site.

Primary Actor: Member

MAIN SUCCESS SCENARIO

1. The Member will see click the report button under a photo.

2. The Member will be prompted with a text box that asks them to submit their reason to

report the photo.

3. The Member will then click to submit the report.

4. Web server will receive the report and give it to backend services.

5. The Web Server will receive the request and send it to backend services.

6. Backend services will process the report and then send it to Admins to decide further

action.

6a. Admins will review report and photo and decide further action.

Use Case # 20

Use Case: Member invites a new member.

**CHARACTERISTIC INFORMATION**

Success End Condition: Member sends invite that a new member accepts to join site.

Failed End Condition: Potential new member rejects or ignores invite.

Primary Actor: Member

MAIN SUCCESS SCENARIO

1. Member want to send an invite link via SMS or email.

2. The Member will use an allotment of their overall points to send an invitation.

3. The user will navigate to the “Invite New Member” functionality on their profile page.

4. They will be redirected a page that gives them a form to fill out.

4a. The form will include the email address of the potential new member, their name, and

how the current Member knows them.

5. The Member will click submit.

6. The Web Server will receive the submission and pass it to backend services.

7. Backend services will receive the submission and send out an invitation link to the

provided email. Backend services will keep track of how long the email has been out before

it is accepted or rejected.

7a. If they join Member gains points.

7b. If they decline Member loses points.

7c. If the link expires Member loses points.

Member Unfollows Other Member (if not invitee) # 21

**CHARACTERISTIC INFORMATION**

Success End Condition: Member will no longer see other Member’s updates.

Failed End Condition: They still have to see other Member’s updates.

Primary Actor: Member

MAIN SUCCESS SCENARIO

1. The Member will navigate to the other member’s profile page that they wish to unfollow.

2. The Member will toggle the “Unfollow” button.

3. Backend services will hide the other member’s activity.

Use Case # 22

Use Case: Member blocks another Member.

**CHARACTERISTIC INFORMATION**

Success End Condition: Member blocks all interactions with other Member on site.

Failed End Condition: They are not allowed to block.

Primary Actor: Member

MAIN SUCCESS SCENARIO

1. The Member will go to their follower’s page.

2. The Member will find the block option and choose it.

3. The Member will be given an optional dialogue box where they can input why they are

blocking their follower.

3a. The Member can enter the reason they are blocking their follower and click submit.

3b. The Web Server will send this request to Admins.

3c. Admins can review the report and decide any further actions.

4. Backend services will receive this information from the web server and will block the

follower from accessing the member’s profile, information and from contacting them.

Use Case # 23

Use Case: Admin removes members access to Members Only because they were reported for acting wrongly

**CHARACTERISTIC INFORMATION**

Success End Condition: Admin removes member

Failed End Condition: Admin does not remove member

Primary Actor: Admin

Trigger: Current member was flagged and need to be removed from the site

MAIN SUCCESS SCENARIO

1. Admin directs themselves to this member’s profile

2. Admin hits the button to remove this member

3. Webserver receives request and sends that to the backend

4. Backend deleted this page by removing this profile page, but keeping all their data

5. User has been removed

Use Case # 24

Use Case: Admin removes content that has been flagged as inappropriate/illegal

**CHARACTERISTIC INFORMATION**

Success End Condition: Admin removes post

Failed End Condition: Admin does not remove post

Primary Actor: Admin

Trigger: current post is inappropriate and should not be on Members Only

MAIN SUCCESS SCENARIO

1. Admin directs themselves to the post that has been flagged

2. Admin hits the button labeled “remove post”

3. Webserver receives request to remove the post and sends that to the backend

4. Backend removes that post from that members profile and removes the post from the database

5. Post has been deleted

Use Case # 25

Use Case: Administrator edits members photo because they want to add a filter

**CHARACTERISTIC INFORMATION**

Success End Condition: Admin adds filter to members photo

Failed End Condition: Admin does not add filter to members photo

Primary Actor: Admin

Trigger: Photo doesn’t look good enough as it is

ADMIN WANTS TO ADD FILTER

1. Admin directs themselves to members photo

2. Admin clicks the edit button next to photo

3. Admin applies filter using the editing software

4. Admin hits the save button when they are done

5. Webserver receives request to save and sends that to the backend

6. Backend removed old photo and replaces that with the new edited photo

7. Picture has been filtered

ADMIN WANTS TO ADD SPONSORED CONTENT

1. Admin directs themselves to members photo

2. Admin clicks the edit button next to photo

3. Admin applies sponsored content using the editing software

4. Admin hits the save button when they are done

5. Webserver receives request to save and sends that to the backend

6. Backend removed old photo and replaces that with the new edited photo

7. Sponsored content has been added

Use Case # 26

Use Case: Idol wants to be able to do everything that a member can do.

**CHARACTERISTIC INFORMATION**

Success End Condition: Idol can do all member actions (privileged member)

Failed End Condition: Idol can’t do all member actions

Primary Actor: Idol

Trigger: Idol wants to use Members Only as in

IDOL WANTS TO DO MEMBERS ACTIONS

1. See user cases for members from User Story # 6 - 22

Use Case # 27

Use Case: Idol invites a new member to the site to increase the number of members

**CHARACTERISTIC INFORMATION**

Success End Condition: Idol successfully invites a new member to Members Only

Failed End Condition: Idol unsuccessfully invites a new member to Members Only

Primary Actor: Idol

Trigger: To increase the number of members and traffic to the website

IDOL INVITES A NEW MEMBER TO THE SITE

1. Idol clicks on part of website to send invitation link

2. Idol enters email address of invitee

3. System confirms it’s a valid email address

4. Idol confirms to send invitation

IDOL INVITED MEMBER WITH CODE

1. Idol generates invitation code

2. Idol can now hand this code out to potential new members

Use Case # 28

Use Case: Idol can be followed by anyone

**CHARACTERISTIC INFORMATION**

Success End Condition: Idol can be followed by everyone on Members Only

Failed End Condition: Member can’t follow Idol

Primary Actor: Idol

Trigger: A member wants to follow any Idol

IDOL CAN BE FOLLOWED BY ANYONE

1. Member directs site to the Idol’s page

2. Member locates the “follow” button on Idol’s page

3. Member clicks the follow button

4. Member is successfully following the idol (Idol can’t change visibility settings)

Use Case # 29

Use Case: Idol wants to post sponsored content on Members Only Site

**CHARACTERISTIC INFORMATION**

Success End Condition: Idol posts sponsored content

Failed End Condition: Idol fails to post sponsored content

Primary Actor: Idol

Trigger: Idol would like to profit by posting sponsored content to make money by increasing Members Only Traffic per agreement with Owner

IDOL WANTS TO POST SPONSORED CONTENT

1. Owner or Company wants a specific product in a post/photo/comment so to be sponsored

2. Actor from step (1) reaches out to Idol via Members Only

3. Idol reads notification from the previous actor

4. Idol either follows notification to post to sponsor or makes own post

5. Idol writes post including sponsored content.

Use Case # 30

Use Case: The system deducts a point from a member

**CHARACTERISTIC INFORMATION**

Success End Condition: Update a member’s credit information in a database and send an update to a frontend

Failed End Condition: Send an error to a frontend

Primary Actor: The system

Trigger: 1) Alex's invitation gets rejected or 2) a member's invitation gets expired or 3) a member makes negative actions

MAIN SUCCESS SCENARIO:

1. The backend receives a request

2. The backend sent a request to the credit card company using a credit card software

3. If the request is denied

a. The system freezes the member’s account

b. The system changes the permission of the member in a database

3. If the request is successful

a. The database finds the matched member and takes a point off

b. The system sends updated point information to a frontend

c. The frontend updates the point information in a local storage

d. The member recognizes the changes

Use Case # 31

Use Case: A system awards a point from a member

**CHARACTERISTIC INFORMATION**

Success End Condition: Update a member’s credit information in a database and send an update to a frontend

Failed End Condition: Send an error to a frontend

Primary Actor: The system

Trigger: 1) Alex’s invitation gets accepted 2) Alex makes positive actions

MAIN SUCCESS SCENARIO:

1. The backend receives a request

2. The database finds the matched member information in a database

3. Updates the matched member’s information

4. The system sends updated point information to a frontend

5. The system updates the point information in a local storage

5. Alex recognizes the changes

Use Case # 32

Use Case: The system gets a login request from a frontend

**CHARACTERISTIC INFORMATION**

Success End Condition: Alex can log in

Failed End Condition: Sends an error to a frontend

Primary Actor: The system

Trigger: The frontend sends a login request to a backend

MAIN SUCCESS SCENARIO:

1. The backend gets a request that contains user information

2. The backend stores a member’s current IP address into the database

3. The backend confirms whether the information given is matched with one of the user data in the database

4. The backend sends tokens and matched user information to the frontend

5. The frontend redirects a member to the main page

Use Case # 33

Use Case: The system gets a logout request from a frontend

**CHARACTERISTIC INFORMATION**

Success End Condition: Alex do log-out

Failed End Condition: Alex can’t log-out

Primary Actor: The system

Trigger: Alex clicks a logout button in a frontend

MAIN SUCCESS SCENARIO:

1. The frontend detects when the logout button is clicked

2. The frontend sends Alex’s singed out time to a backend

3. The backend stores Alex’s logout time

4. The frontend destroys the session data in a local storage

5. The frontend redirects Aelx to a landing page

Use Case # 34

Use Case: The system gets a registration request from a frontend

**CHARACTERISTIC INFORMATION**

Success End Condition: The new member data is created in a database

Failed End Condition: The system sends an error to a frontend

Primary Actor: The system

Trigger: A potential member (Bob) fills out a registration form and clicks the register button

MAIN SUCCESS SCENARIO:

1. The backend gets a request that contains a potential user’s information

2. If the credit card information already exists

a. The backend sends an error to the frontend

b. Bob checks the error

3. If the credit card information does not exist in the database

a. The system checks the given credit card information is valid using credit card software

b. If the data is valid

1) Store new user information into the database

2) The frontend redirects Bob to a login page

c. If the data is NOT valid

1) The backend sends an error to the frontend

2) Bob checks the error

Use Case # 35

Use Case: The system collects a member’s interest

**CHARACTERISTIC INFORMATION**

Success End Condition: The system stores gathered information to a database

Failed End Condition: The system prints an error in a server

Primary Actor: The system

Trigger: Alex performs specific expected actions

MAIN SUCCESS SCENARIO:

1. The frontend detects an item being clicked when a member clicks a specific post or

1. The frontend recognizes an item(content) on the current screen (if a member stays longer than a particular second at the same page without scrolling down or going out to other pages)

2. The frontend sends the item(content) information to the backend

3. The backend receives the data and stores them in the database

Use Case # 36

Use Case: A system converts the original URL to the shortened URL

**CHARACTERISTIC INFORMATION**

Success End Condition: The system sends a shortened URL to a frontend

Failed End Condition: The system throws an error and keep using original URL

Primary Actor: The system

Trigger: The frontend sends the request to a backend

MAIN SUCCESS SCENARIO:

1. The frontend sends a request to the backend with original URL information

2. The backend gets the request and uses a hash function to generate a shortened URL

3. The system saves the shortened URL into the database

4. the system sends the shortened URL to the frontend