Use Case # 1

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

1. New member receives email invitation link to join *Members Only*.

2. User clicks on invitation link.

3. User is brought to *Members Only* webpage to ACCEPT or DENY invitation.

4. User clicks ACCEPT on webpage.

5. User is brought to *Sign Up* webpage on *Members Only*.

6. User fills in required fields in textboxes.

6a. Required fields include: First Name, Last Name, Street Address, City/Town, Zip Code, Country

(U.S. only option), valid U.S. Phone Number (with area code), valid Email Address, valid

Password.

7. User checks box as acceptance of Terms of Service.

8. User clicks SUBMIT. Confirmation email is sent to user.

9. User clicks confirmation link sent to email address provided above.

10. User is brought to *Members Only* webpage, confirming account creation.

10. Account is created.

SIGNING UP WITH A CODE

1. New Member receives code from an Idol

2. New Member directs themselves to Members Only

3. New Member hits button to sign up with code

4. New Member enters code and hits enter

5. New Member enters credit card info and password

6. New Member Checks credit card statement to see amount charged to card

7. New Member enters amount charged into system

8. New Member accepts terms of service

9. New Member is redirected to Members Only front page

# 2 DELETE

Use Case: New member does not want to join *Members Only*, denies invitation.

**CHARACTERISTIC INFORMATION**

Success End Condition: Invitation rejected.

Failed End Condition: Invitation not rejected.

Primary Actor: Potential New Member

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

MAIN SUCCESS SCENARIO

1. New member receives email invitation link to join *Members Only*.

2. User clicks on invitation link.

3. User is brought to *Members Only* webpage to ACCEPT or DENY invitation.

4. User clicks DENY on webpage.

5. User is brought to confirmation webpage, asking to confirm their selection to DENY invitation.

6. User clicks CONFIRM.

7. User is brought to webpage thanking them for their reply.

7. Invitation rejected.

Use Case # 3

Use Case: New member does not want to join *Members Only*, 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

1. New member receives email invitation link to join *Members Only*.

2. User ignores email.

3. Invitation expires after 3 days.

SUB-VARIATIONS

1. User clicks invitation link.

1a. User is brought to *Members Only* webpage to ACCEPT or DENY invitation.

1b. User clicks DENY on webpage.

1c. User is brought to confirmation webpage, asking to confirm their selection to DENY

invitation.

1d. User leaves webpage/does **not** click CONFIRM.

1e. Invitation expires after 3 days.

Use Case # 4

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

1. After successful CASE# 1, user receives email to confirm payment details to begin using

*Members Only* with link.

2. User clicks link in email to confirm payment details.

3. User is brought to webpage to fill out credit card details.

4. User enters valid 16-digit credit card number.

4a. Credit card must be under name/address as user used to sign up for *Members Only*.

5. User enters 3-digit CVV of credit card.

6. User checks box confirming monthly billings.

7. User clicks SUBMIT.

8. User is shown PROCESSING webpage.

9. User is brought to *Members Only* webpage, confirming credit card charge.

SUB-VARIATIONS

Use Case # 5

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# 4]

MAIN SUCCESS SCENARIO

1. After successful USE CASE# 4, user receives email to confirm charge amount.

2. User clicks on hyperlink in email to confirm charge amount.

3. User is brought to *Members Only* webpage, with entry field for value of charge.

4. User enters the correct dollar value charged to credit card during USE CASE #4.

5. User clicks CONFIRM.

6. User is redirected to *Members Only* webpage confirming their account is ACTIVE.

7. Credit card charge value confirmed, full access to *Members Only* granted.

SUB-VARIATIONS

1. After successful USE CASE# 4,

Use Case # 6

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

1. User directs themselves to *Members Only* home page.

2. User fills in email field.

2a. Must be valid email address.

3. User correctly fills in password field.

3a. Must be valid password associated with valid email address.

4. User clicks the LOGIN button, next to password field.

4. User is redirected to their *Members Only* feed webpage.

MEMBER FORGOT PASSWORD

1. Member directs themselves to Members Only page

2. Member fills in their email and password

3. Member clicks the LOGIN button.

4. System checks email to see is that is in the database

5. System checks password to if that matches the email

6. System denies member access to account and instructs member to reenter password

MEMBER FORGOT EMAIL

1. Member directs themselves to Members Only page

2. Member fills in their email and password

3. Member clicks the “login” button

4. System checks email to see is that is in the database

6. System denies member access to account and instructs member to reenter email

Use Case # 7

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

1. User directs self to *Members Only* homepage.

2. User clicks SETTINGS from homepage.

3. User is redirected to page of various settings.

4. User clicks CHANGE PASSWORD.

5. User enters new password in textbox, meeting security requirements.

6. User re-enters new password into confirmation textbox.

7. User clicks CONFIRM.

8. User is redirected to webpage, confirming successful password change.

9. Member successfully changes password.

Use Case # 8

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.

MAIN SUCCESS SCENARIO

1. Member will go to their personal page

2. member will go to their followers and find the person they are concerned about

3. Member will choose to block this follower

4. Person is blocked

REPORT MEMBER

1. Member will go to their personal page

2. member will go to their followers and find the person they are concerned about

3. Member will choose the option to report a follower

4. Member will be redirected to a report issue page with a form detailing other Member’s name and why they are being reported

5. Member will fill out form

6. Member will hit the “Send” button

7. System will take the text in text box and add that to a text file, and send that to the admins

8. Admins will review report and decide further actions

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

1. Member will go to their personal page

2. Member will access the setting portion of their page

3. Member finds the toggle switch that allows them to switch their profile to private or not

4. Member toggles this switch, shutting off their profile page and feed to non followers

Use Case # 9

Use Case: 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: Member

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

CREDIT CARD DECLINED/EXPIRED/CANCELLED

1. Member will login to their profile

2. Member will navigate to the settings page on their personal profile

3. Member will access their account information page

4. Member will choose change credit card feature

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

5a. User will enter name of card

5b. User will enter card number

5c. User will enter expiration date

5d. User will enter CVV

6. Member will hit the submit form button

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 # 10

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

**CHARACTERISTIC INFORMATION**

Success End Condition: confirm credit card charge

Failed End Condition: charge wrong

Primary Actor: 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 the member

2. Credit card system will charge member a random amount between $0.20 and $0.45.

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

6. Member will click link provided in email

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

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

8a. Member 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. Member doesn’t enter amount in time and is blocked from account

2. When member 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

4. Credits card system sends another random charge

5. System sends Member another email

6. System starts a timer for 3 days

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

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

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

Use Case # 11

Use Case: Members updates their *Members Only* profile.

**CHARACTERISTIC INFORMATION**

Success End Condition: Profile successfully updated.

Failed End Condition: Profile unsuccessfully updated.

Primary Actor: Member

Trigger: Member wants to add content to *Members Only* profile.

UPDATE WHERE MEMBER GOES TO SCHOOL

1. Member will login to their Members Only profile

2. Member will navigate to their personal profile page

3. Member will access the settings page on their personal profile

4. Member will click on change personal information on their settings page

5. Member will update their place of education

6. Member will hit the “submit” button

7. System web server will receive the change and backend services will process it

7a. Backend will send changes to database, adding it to the list of schools/places of education that this Member has attended or shown interest in

MEMBER WANTS TO UPDATE WHERE THEY WORK

1. Member will login to their Members Only profile

2. Member will navigate to their personal page

3. Member will access the settings page on their personal profile

4. Member will click on change personal information on their settings page

5. Member will navigate to the work/jobs section on this page and hit change/update/add

6. Member will add a new job in the input form them hit “submit”

7. System web server will receive the change and backend services will process it

7a. Backend will send changes to database, adding it to the list of jobs that the members has worked

MEMBER WANTS TO UPDATE THERE THEY LIVE

1. Member will login to their Members Only profile

2. Member will navigate to their personal page

3. Member will access the settings page on their personal profile

4. Member will click on change personal information on their settings page

5. Member will navigate to the residence section on this page and hit change/update

6. Member will update where they live and hit “submit”

7. System web server will receive the change and backend services will process it

7a. Backend will send changes to database, adding it to the list of places that the members has lived

MEMBER WANTS TO UPDATE THEIR *ABOUT ME* SECTION

1. Member will login to their Members Only profile

2. Member will navigate to their personal page

3. Member will access the settings page on their personal profile

4. Member will click on change personal information on their settings page

5. Member will navigate to the about me section on this page and hit change/update/add

6. Member will edit their about me section in a provided text box and hit “submit”

7. System web server will receive the change and backend services will process it

7a. Backend will send changes to database, and update their about me section in the database

MEMBER WANT TO UPDATE PROFILE PICTURE

1. Member will login to their Members Only profile

2. Member will navigate to their personal page

3. Member will click on their active profile picture

4. A series of options will pop up, allowing the member to choose to change their profile picture

5. The user will click update profile picture

6. The user can upload a new photo from the storage on their personal device or change it to a

photo they have already uploaded.

7. After selecting a photo the member will click save

8. the request will be sent to the Web Server

9. Backend services will handle the request and change the members profile picture

10. Backed services will also save this profile picture to the database

Use Case # 12

Use Case: 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: Member

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

MEMBER POSTS LINKS IN STATUS

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

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

2a. tinyURL generator will shorten the URL given

3. The member will hit “post”

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

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

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

Use Case # 13

Use Case: Member leaves comment on friend’s status.

**CHARACTERISTIC INFORMATION**

Success End Condition: Comment successfully posted.

Failed End Condition: Comment unsuccessfully posted.

Primary Actor: Member

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

MAIN SUCCESS SCENARIO

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

2. The member 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. Member 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

Use Case # 14

Use Case: 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: Member

Trigger: Member wants to invite their friend

MAIN SUCCESS SCENARIO

1. Member navigated to the “Invite New Member” functionality on their profile page

2. On this page, the member will fill in new members email address and how they know them. Once this is done member will hit submit

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. Member will gain points

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

3. Backend will add that many points to members points system

NEW MEMBER DECLINES OFFER/TIMER RUNS OUT

1. Member loses points

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

3. Backend will subtract that many points from members points system

DELETE

Use Case: Members unfollows Idol on *Members Only*.

**CHARACTERISTIC INFORMATION**

Success End Condition: Member unfollows Idol

Failed End Condition: Member keeps following Idol

Primary Actor: Member

Trigger: Member no longer wants to follow idol anymore because of lost interest.

MAIN SUCCESS SCENARIO

1. A member navigates to the Idols profile page

2. A member clicks the unfollow button

3. Webserver receives this information and sends it to the back end

4. Backend will no longer show content from that Idol

Use Case # 16

Use Case: Member posts photo to profile.

**CHARACTERISTIC INFORMATION**

Success End Condition: Photo successfully posted to profile.

Failed End Condition: Photo failed to post/blocked by admins.

Primary Actor: Member

MAIN SUCCESS SCENARIO

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

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

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

4. The Member 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.

Use Case # 17

Use Case: 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: Member

MAIN SUCCESS SCENARIO

1. The Member 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. The Member will scroll through various premade filters available on Members Only..

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

4. The post photo process then continues from here.

REMOVING FILTER

1. The Member clicks on the photo they wish to remove the filter on.

2. The Member chooses “options”.

3. The Member chooses remove filters.

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

5. The Member will then be prompted to confirm their changes.

6. The Member will click “Confirm”.

7. Request will be sent to the web server.

8. Web server will send it to backend.

9. Backend services will remove the filter from the photo.

ADDING (SPONSORED) CONTENT

1. The Member will follow the posting photos process.

2. The Member will then click add content.

3. The Member can choose any of the items and place them anywhere in their photo.

4. The Member 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. The Member will go to their photo with added content.

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

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

navigate to it and choose to remove it.

4. The Member 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.

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.

COMBINE WITH 21

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. Original Member/Poster can not see Admin’s changes.

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

ADMIN ADDS CONTENT INTO COMMENTS

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

DELETE

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)

DELETE

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: A 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: A system

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

MAIN SUCCESS SCENARIO:

1. A backend receives a request

2. A backend sent a request to the credit card company using a credit card sfotware

3. If the request is denied

a. A system freezes the member’s account

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

3. If the request is successful

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

b. A system sends updated point information to a frontend

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

d. A 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: A system

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

MAIN SUCCESS SCENARIO:

1. A backend receives a request

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

3. Updates the matched member’s information (adding a credit)

4. A system sends updated point information to a frontend

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

5. A member recognizes the changes

Use Case # 32

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

**CHARACTERISTIC INFORMATION**

Success End Condition: A member can log in

Failed End Condition: Sends an error to a frontend

Primary Actor: A system

Trigger: A frontend sends a login request to a backend

MAIN SUCCESS SCENARIO:

1. A backend gets a request that contains user information

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

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

4. A backend sends tokens and matched user information to a frontend

5. A frontend redirects a member to the main page

Use Case # 33

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

**CHARACTERISTIC INFORMATION**

Success End Condition: A member do log-out

Failed End Condition: A member can’t log-out

Primary Actor: A system

Trigger: A member clicks a logout button in a frontend

MAIN SUCCESS SCENARIO:

1. A frontend detects when the logout button is clicked

2. A frontend sends a member’s singed out time to a backend

3. A backend stores a member’s logout time

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

5. A frontend redirects a member to a landing page

Use Case # 34

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

**CHARACTERISTIC INFORMATION**

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

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

Primary Actor: A system

Trigger: A member fills out a registration form and clicks the register button

MAIN SUCCESS SCENARIO:

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

2. If the credit card information already exists

a. A backend sends an error to a frontend

b. A member checks the error

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

a. A 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 a database

2) A frontend redirects a member to a login page

c. If the data is NOT valid

1) A backend sends an error to a frontend

2) A member checks the error

Use Case # 35

Use Case: A system collects a member’s interest

**CHARACTERISTIC INFORMATION**

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

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

Primary Actor: A system

Trigger: A member performs specific expected actions

MAIN SUCCESS SCENARIO:

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

1. A 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. A frontend sends the item(content) information to a backend

3. A backend receives the data and stores them in a database

Use Case # 36

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

**CHARACTERISTIC INFORMATION**

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

Failed End Condition: A system throws an error

Primary Actor: A system

Trigger: A frontend sends the request to a backend

MAIN SUCCESS SCENARIO:

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

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

3. A system saves the shortened URL into a database

4. A system sends the shortened URL to a frontend