Cases 1-4: Justin Huynh

Use Case UC1: User sign up and login

Stakeholders:

User-wants to interact with the media platform

System admins- wants to ensure proper storage of user data

Preconditions: User has a valid email address, The user does not have an account

tied to this email address, the platform is not experiencing technical issues.

Success guarantee: User account is saved on the platform. Users can log into their

account, edit their profile, create posts, and perform other use cases exclusive

to users with accounts.

Main Success Scenario: 1. A user visits the create account page of the platform. 2

The system displays a form for users to enter information for the account they wish to create.

3. The user enters their name, date of birth, username, and password. 4. The user clicks a checkbox to agree to the terms and conditions. 5. The user submits the form. 6. The system

validates the input is in the correct format. 7. The website prompts

the user for an email address. 8. The user requests to sign up using their google account.

An external browser link opens prompting the user to sign into their google account. The

user signs into their google account and the system sends a verification code to their

google account. 9. The user enters the verification code, which is verified by the system.

The system creates the user account and stores the user's account in the platform.

Alternate Flows:

1a. The user tries to create a post before logging in. The user is prompted to log in. The

user indicates they want to create an account. The user is redirected to the create account

page.

4a. The user does not click the checkbox agreeing to the terms and conditions and submits the form.

The system displays an error message asking the user to agree to the terms and conditions.

6a. One or more tokens of data submitted by the user are invalid as inputs. The system displays

an error message detailing which token is invalid and how to fix it. The system prompts the user

to change information.

6b. The username is already taken. The system displays an error message and prompts the user to

change their requested username.

6c. The password is too weak. The system displays an error message detailing why the user's password

doesn't meet the platform's requirements and prompts the user to enter a new password.

8a. The user requests to sign up with an email address domain other than google. The user enters

their email address and the system sends a verification code to the email.

9a. The verification code entered by the user is incorrect. The system displays an error message and

prompts the user to enter the verification code again.

9b. The user doesn't receive a verification code in their email. The user requests the system to email

their address with another verification code. The system sends the user another verification code. The

user enters the verification code.

9c. The verification code entered by the user has expired. The system displays an error message and prompts

the user to request a new verification code.

Technology and Data List:

3-Data is entered by a keyboard.

Frequency of Occurrence: Occurs whenever a new user signs up to create an account. Sign up occurs with a low frequency, log in occurs with a high frequency.

Use Case UC2: Create Posts

Stakeholders:

Users - want to interact with the community on the platform and post questions/answers

Preconditions - The user is logged in to the site, the platform is not experiencing technical issues.

Success guarantee - The post is created and appears on the site for other users to see.

Main success scenario - 1. The user indicates they are going to create a post to the system. 2. The system asks which community/group message board the user wants to post on. 3. The user fills out a post creation form with data for the title of the post, the post text, and optional attached media files. 4. The user submits a post request. 5. The system saves the post to the site.

Extensions -

2a - The user posts without specifying the community board to post to. The system displays an error message and prompts the user to specify a community to post in.

3a - The user is missing required fields. The system displays an error message and prompts the user to fill in the required fields.

3b - Unsupported file type. The user attempts to attach a file with an invalid format. The system displays an error message and prompts the user to remove the invalid attachments before posting.

Technology and Data List -

3 - Text data is entered by the user with a keyboard.

Frequency of Occurrence - Occurs whenever a user wants to create a post. Posts are created with a high frequency.

Use Case UC3: Display Feed

Stakeholders:

Users - searching for relevant content to their interests

Preconditions - the user is logged in to the site, the platform is not experiencing technical issues

Success guarantee - The user is able to see and interact with posts on the site.

Main success scenario - 1. The user visits their home feed section of the platform. 2. The system loads content relevant to the user based on their interests and saved posts. 3. The user interacts with a post on the feed, sending the system a request to like/comment on the post. The user can request to see the full post and the system redirects the user with a link to the full post. 4. If the user has reached the bottom of the page, the system will load more posts. 5. If the user requests to refresh the page, the system will refresh the feed with new content if available.

Extensions - 2a. The system did not find content relevant to the user. The system will fetch trending and frequently visited posts. 3a. The user does not interact with the post. The user may continue viewing posts on their feed. 3b. The user reports that a post is not interesting to them. The system stops recommending posts similar to the one the user reported.

Special Requirements - The feed must load posts quickly. The feed should not take more than ten seconds to load a new set of posts.

Technology and Data List -

4-User likely to navigate through the feed with a mouse or scrollbar.

Frequency of Occurrence - Occurs when users want to browse through their feed for content. Browsing occurs at a very high frequency.

Use Case UC4 - Change User Settings

Stakeholders:

User - changing their settings on the platform

Preconditions - The user is logged into their account.

Success Guarantee - The user successfully changed their settings.

Main Success Scenario - 1. The user navigates to their settings menu. 2. The user navigates to the setting category of the setting they wish to change. 3. The user fills in data pertaining to the changes they desire to make. 4. The user submits the changes they wish to make. 5. The system updates the user’s settings and displays a confirmation message.

Extensions - 3a. The user enters invalid input. The system displays an error message explaining which change is invalid and prompts the user to reenter information. The user corrects the invalid input and resubmits changes.

Technology and Data List -

3-Users are likely to fill in data with a keyboard and mouse.

Frequency of Occurrence - Occurs when a user wants to change their settings. Settings changes occur with a low frequency.

Case 5-8: Raman Khatri

Use Case UC5: User sends a message request to another user

Stakeholders:

Users: Seeking to establish communication with other users, typically professionals or academics, for networking or collaboration purposes.

Preconditions:

* The user is logged in to the platform.
* The recipient user has messaging enabled.
* The platform is not experiencing any technical issues.

Success guarantee:

The message request is successfully sent to the recipient user, and the sender is notified that the request has been sent.

Main success scenario:

1. The user navigates to the profile of another user they want to message.
2. The system displays a button to "Send Message Request" if messaging is enabled for the recipient.
3. The user clicks on the "Send Message Request" button.
4. The system prompts the user to write an introductory message (optional).
5. The user submits the message request.
6. The system confirms the request, notifying the sender that the message has been sent.
7. The system sends a notification to the recipient about the message request. The recipient can choose to accept or decline the request.

Extensions:

2a. Messaging is disabled for the recipient user. The system will notify the sender that messaging is not available for this user.

5a. The user cancels the message request. The system discards the request without sending it.

7a. The recipient declines the request. The system will notify the sender that the request has been declined, and no further messages can be exchanged unless the recipient changes their preference.

Technology and Data List:

* The system must track message requests in the database to ensure requests are visible in both the sender's and recipient's notification logs.
* Data on whether the recipient has enabled/disabled messaging must be available in real-time.

Frequency of Occurrence:

Occurs whenever users wish to initiate a direct message for professional or academic networking. This happens frequently as the platform facilitates connections between users.

Use Case UC:6 Users message each other (all messages after the first message)

Stakeholders:

Users: Engaging in ongoing conversations with other users for networking, collaboration, or discussion purposes.

Preconditions:

* Both users have accepted the initial message request and are connected for direct messaging.
* The users are logged into the platform.
* The platform is not experiencing any technical issues.

Success guarantee:

* Users can send and receive messages seamlessly, with confirmation that the messages have been successfully delivered and read.

Main success scenario:

1. User A navigates to their messaging inbox and selects the conversation with User B.
2. The system loads the conversation history between User A and User B.
3. User A writes a new message in the text input field and clicks "Send."
4. The system sends the message to User B and confirms delivery with a visual indicator (e.g., "Delivered" status).
5. User B receives the message in real-time and the system notifies them of the new message with a notification or sound.
6. User B reads the message, and if read receipts are enabled, the system updates the status to "Read" for User A.
7. The conversation continues as User B responds, and the system follows the same flow (steps 3-6) for each new message.

Extensions:

3a. User A's message fails to send due to network issues. The system displays an error message and attempts to resend automatically when the connection is restored.

5a. User B does not have real-time notifications enabled. The system stores the message and delivers it once User B is active on the platform.

6a. User A has disabled read receipts. The system will not update the message status to "Read," regardless of whether User B views the message.

6b. Either user chooses to delete a message. The system confirms deletion on the user's side but retains the message for the other participant unless both users delete it.

Technology and Data List:

* Real-time messaging relies on WebSocket or similar technology to ensure instant message delivery.
* Message status (e.g., sent, delivered, read) should be tracked and stored in the database.
* The system must maintain an archive of all conversations, with the ability for users to retrieve older messages.

Frequency of Occurrence:

Occurs frequently as users engage in ongoing conversations throughout their time on the platform, especially for collaborations or extended discussions.

Use Case UC7: User shares a post (via url)

Stakeholders:

Users: Seeking to share content (posts) from the platform with others, either within or outside the platform.

External Users: Recipients who may view the shared post via the URL, even if they are not platform members.

Preconditions:

* The user is logged into the platform.
* The post is public or the user has appropriate permissions to share it.
* The platform is not experiencing any technical issues.

Success guarantee:

* The post is successfully shared via a unique URL, and the recipient can view the post when they open the link.

Main success scenario:

1. The user navigates to the post they wish to share.
2. The system displays a "Share" button or option under the post.
3. The user clicks on the "Share" button, and the system presents sharing options, including "Copy URL."
4. The user selects "Copy URL," and the system generates a unique URL link to the post, which is copied to the user's clipboard.
5. The user shares the URL with others (via email, messaging apps, social media, etc.).
6. The recipient opens the shared URL.
7. The system loads the post for the recipient. If the post is public, the recipient can view it directly. If it is private, the system will request the recipient to log in or sign up to view the post (depending on permissions).

Extensions:

2a. The post is private and not shareable. The system will notify the user that the post cannot be shared due to privacy settings.

5a. The user chooses to share the post via integrated options (e.g., social media buttons). The system generates a shared link formatted for the chosen platform and allows the user to post it directly.

7a. The recipient does not have an account or is not logged in, and the post is private. The system prompts the recipient to log in or create an account to access the post.

Technology and Data List:

* The system must be able to generate unique, shareable URLs for posts.
* Post privacy settings (public/private) need to be checked in real-time to determine whether the post can be shared.
* External link previews (metadata) should be provided when sharing to social media or messaging apps.

Frequency of Occurrence:

* Occurs frequently as users share relevant content with peers, both within and outside the platform, to encourage networking or spark discussions.

Use Case UC8: User deletes their account

Stakeholders:

Users: Wishing to permanently remove their account and personal data from the platform.

Platform Administrators: Responsible for ensuring that account deletion processes comply with legal and data privacy regulations.

Preconditions:

* The user is logged into their account.
* The platform is not experiencing technical issues.
* The user must confirm their intention to delete the account, as this action is irreversible.

Success guarantee:

The user’s account and associated data are permanently deleted, and the user is informed of the successful deletion. The system may retain some anonymized data for statistical purposes if allowed by policy.

Main success scenario:

1. The user navigates to their account settings.
2. The system displays the option to "Delete Account."
3. The user selects "Delete Account," and the system prompts a confirmation request, explaining that the action is permanent and all associated data will be removed.
4. The user confirms their decision by clicking "Confirm" or entering their password to proceed.
5. The system processes the deletion request, removing all user data (e.g., profile, posts, messages, connections) and sends a final confirmation email to the user with details of the account deletion.
6. The system logs the user out and displays a confirmation message stating that the account has been successfully deleted.
7. The user can no longer log in or access the platform using their previous credentials.

Extensions:

3a. The user changes their mind before confirming and cancels the deletion process. The system does not delete any data and returns the user to the account settings page.

5a. The system encounters an error while processing the deletion. The system displays an error message, asks the user to try again, and provides an option to contact support.

6a. The platform offers a grace period (e.g., 30 days) for account restoration. The system sends a recovery link in the confirmation email that allows the user to undo the deletion within the grace period if they change their mind.

Technology and Data List:

* User data (e.g., profile information, posts, messages, interactions) must be fully deleted from the database or anonymized where necessary.
* The system should have mechanisms to handle data deletion requests in compliance with data protection and privacy laws.
* An account recovery mechanism must be available if a grace period for recovery is offered.

Frequency of Occurrence:

Occurs infrequently, typically when users decide to leave the platform permanently or switch to another service.

Case 9-12: Connor Marks

Use Case UC9: User searches for a post/user

Stakeholders:

Users - want to search for a post that answers a specific question and/or a user's profile where poster information and previous posts are recorded.

System admins - Want the search functionality to work appropriately such that users can find exactly what they are looking for.

Preconditions:

* The user is logged into the website.
* The system is set up such that it has a database of stored information about user profiles, previous posts, and a search functionality that works in finding requested information.

Success guarantee:

* The user finds the requested information entered into the search field.
* The user is able to navigate to the requested post or user profile and have the information displayed to them.

Main success scenario:  
 1.) The user has a question about an industry specific problem relating to their profession.

2.) The user logs onto the website and navigates to the search bar.

3.) The user enters a question matching the search criteria and a list of search results is displayed.

4.) The user finds the post they were looking for and clicks the link which takes them to the post with all the information displayed.

Extensions:

3a.) The user enters their question in the search field and the outcome shows 0 results matching their search criteria

1. The system displays a collection of suggested search results that are adjacent to the initial search request.
2. The user can choose to re-enter a new search result or click on a suggested search which will take them to that page.

3b.) After the user enters the search, a broad range of results are returned which makes it difficult for the user to identify what they are looking for.

1. The system displays filtering options to narrow the search.
2. The user can make a new search.

Technology and Data List:

* 3a.) The search functionality could include voice recognition to provide accessibility to users
* 3b.) Filters can be applied to narrow the search

Frequency of Occurrence:

* This use case will occur very frequently as a large part of the website's functionality is contingent upon users being able to search for what they are looking for.

Use Case UC10: User receives notifications

Stakeholders:

Users: Want to receive notifications about new posts in their field.

Preconditions:

* The website has the functionality to send out catered notifications for users that elect to receive them.
* The website will default to notifications being off until the user is prompted to elect for them.
* The user has set up their profile.
* The users are able to customize their notification preferences.
* The website allows users to join groups in which notifications are enabled.

Success guarantee:

* A user who elects to receive notifications receives them when new posts, updates, or newsletters become available.
* A user who elects not to receive notifications will not receive them.

Main success scenario:  
 1.) A user sets up their profile.

2.) The system asks the user if they would like to receive notifications for updates, newsletters, and community posts for which they are a part of.

1. The user affirms receiving notifications
2. An update is made to the website and a request is made to send the user the notification.
3. The user receives the notification on their device.

Extensions:

1a.) A user has not set up their profile yet.

1. A new update has been made to the website.
2. The system marks the request to send users who have elected to receive notifications of this update.
3. The user does not receive the notification as they are off by default.

1b.) A user has set up their profile and has elected to receive notifications.

1. A new update has been made to the website.
2. The system makes the request to send users who have elected to receive notifications the notification of this update.
3. The user receives a notification about the update.
4. A link is embedded in the notification which allows the user to view the content.

Technology and Data List:

3a.) Notifications may be sent through email, text, or in-app.

Frequency of Occurrence:

* The user will only be asked once to enable notifications when setting up their profile
* The user may be able to modify their notification preferences at any time.
* The user can customize which notifications they receive

The frequency of occurrence will depend on the nature of the notifications a user has elected to receive.

Use Case UC11: User edits one of their posts.

Stakeholders:

Users: Want to be able to edit previously submitted posts to reflect new insights and/or to fix errors.

Preconditions:

* The website has the functionality for posts to be submitted
* The website will be able to display these posts to anyone, or a select group of people based on privacy settings.
* There is functionality for editing posts that have already been submitted.
* A user is logged into the website on the account in which the post was made.
* Users can view their profile and past posts.
* Users can only modify posts that they themselves made.

Success guarantee:

* The user can successfully login, create a post from their account, submit it, and revisit that post to make changes at a later time.

Main success scenario:

1. The user logs into the website.
2. The user navigates to their profile.
3. The user selects the post history tab.
4. The user finds and selects the post they made.
5. The user selects to modify the post.
6. The user modifies the post.
7. The user selects submit with these changes and the post is updated for everyone.

Extensions:

5a.) The user selects a post that they did not make.

1. The user is unable to locate the edit function as it only exists for the creator of that post, and not others.

6a.) The user edits their post, but decides not to save their changes.

1. The user can select an option for canceling an edit in progress.

Technology and Data List:

* 3a.) Post history is displayed differently on a mobile version of the website.
* 6a.) The user saves a draft of the changes of the post without submitting them.

Frequency of Occurrence:

* Relatively frequent. Users on other platforms tend to provide updates and change initial posts.

Use Case UC12: User applies for a verification tag.

Stakeholders:

Users: Users with a need to be verified want their profiles to reflect this.

Preconditions:

* The system can distinguish between verified and unverified users.
* The verification process works correctly in all cases.

Success guarantee:

* The user will be able to submit a request for a verification tag and the automated verification system will process their information and notify them when a result about the verification tag is processed. If the process successfully verifies the user, their profile will reflect this new change with the verified tag.

Main success scenario:

1. The user navigates to the verification application page.
2. The user fills out all information fields.
3. The user uploads any required/additional documents.
4. The system processes this information and makes a determination.
5. If the process was a success, the user will be sent a notification and their profile updated; else the process failed and the user will be notified of the failure to verify.

Extensions:

2a.) The user forgets to fill out one or more of the required fields.

* The submission button is grayed out and the user is prompted to fill in all required fields.

2b.) The user enters an invalid email address or another invalid field.

* The user will be prompted to enter valid information fields.

3a.) The user attempted to enter a document with an unsupported extension

* The user will be prompted to submit documentation with a list of accepted file types.

4a.) There was a connection error when attempting to process the information

* Information entered by the user will be saved and will be automatically submitted again once connection is re-established.

Technology and Data List:

2a.) An autofill feature may be implemented to allow fields to be filled automatically based on profile information.

3a.) Additional file types added to the allowed list.

Frequency of Occurrence:

* Infrequent because only a fraction of users will need to apply for verification tags.

Case 13-16 Emmet Ledell

Use Case UC13: Poster marks question as resolved

Stakeholders:

User- wants to mark their question as resolved

Admins- want experts to stop wasting time answering an already-solved question.

Preconditions:

The user has posted a question and has gotten an answer that they deem satisfactory.

The user is logged in.

Success guarantee:

The question is marked as resolved and no more answers can be given to the question

Main success scenario: 1. The user navigates to the question they have had answered.2. The user selects to mark the question as resolved. 3. The system checks that the current user is the poster of the question. 4. The question has its status updated to resolved. 5. The system no longer allows answers to be given for the resolved question. 6. The user receives confirmation that the question has been marked as resolved

Extensions:

2a. The question is already marked resolved. The system prompts the user that the question is already resolved and asks them to select a different question.

3a. The user did not post the question. System denies the action and prompts that only the original poster can mark a question as resolved.

Technology and Data List:

User authentication system to verify that only the original poster can mark their question as resolved.

Database updates to change the question status and disable further responses.

Notifications or email systems to inform users about the change in status if notifications are enabled.

Frequency of Occurrence:

Ideally this happens as often as posts are made, so it should happen quite frequently

Use Case UC14: User blocks another user

Stakeholders:

User: Wants to block another user for whatever reason  
Blocked user: User that the current user wishes to block

Preconditions:

* The current user is logged in.
* The current user is on the profile of the user they want to block

Success guarantee:

* The user successfully blocks the blocked user.
* The blocked user can no longer see the original user's posts or interact with their account in any way, and vice versa.

Main success scenario:

1. The current user navigates to the profile of the user they wish to block. 2. The user selects the option to block the selected user. 3. The system prompts if they would like to block said user. 4. The system updates the blocked settings in the database, and no longer allows either user to see or interact with the other's posts.

Extensions:

2a. The user already has them blocked. The system alerts the user that they are already blocked and can not perform this action

4a. The blocked user tries to interact with the current user. The system alerts the user that they are blocked by said person and suspends the action

Technology and Data List:

* User authentication and session management to verify the current user's identity.
* A database to store and manage blocking relationships between users.
* Real-time updates to ensure blocking takes immediate effect across all platform features (e.g., messages, posts, notifications).

Frequency of Occurrence:

Occurs when a user want to block another use, so should be quite infrequent

Use Case UC15: User Upvotes/Downvotes Posts or Responses

Stakeholders:

User: wants to this pleasure or displeasure at a response

Poster/Responder: Is getting their post or response interacted with in a positive or negative way

Admin: May want to monitor if an account is upvoting or downvoting too many posts or responses

Preconditions:

The user is logged in  
The user is on a post or response they want to interact with  
The user has not previously voted on the post or response

Success guarantee:

The vote is registered and sent to the system database

The post reflects the new total

User vote history is registered so they can’t vote on the same thing multiple times

Main success scenario:

1. The user navigates to the post or response they wish to vote on. 2. The user selects to upvote or downvote said post. 3. The system checks if they have already given their current response to the post. 4. The system updates the database that stores a post vote count. 5. The system updates the user's voting history 6. The system reflects the new vote total from the database.

Extensions:

3a. The user has already given the same response to said post. The system alerts the user that they are not allowed to give the same response twice to a post and suspends the action

4a. The system can’t store new votes in the database. The system prompts the user that an error has occurred and to try and vote again later.

6a. The system has trouble communicating with the server. The system tries again, and if it fails repeatedly, alerts an administrator that something has gone wrong.

Technology and Data List:

Database for storing vote counts and tracking user voting history.

Real-time data syncing for immediate reflection of vote totals on posts and responses.

Admin tools for monitoring voting behavior across the platform.

Frequency of Occurrence:

This process should happen quite frequently as it should happen multiple times for every post or response given

Use Case UC16: User Reports Inappropriate Content or Behavior

Stakeholders:

User: Finds content offensive or inappropriate and wishes to report it  
Post of content: User who’s content was deemed inappropriate by fellow user  
System: Review each case of his to decide whether the content should be removed

Preconditions:  
The user is logged onto their account

The user is able to see the content that they wish to report

Success guarantee:

The report is sent to the system administrators who will manually review the content the content.  
The user who sent the report receives a message confirming the report was sent

The offending user receives a message that their post was deemed offensive by another user.

Main success scenario:

1. The user goes to the content which they believe to be offense in some way. 2. They press a button to start the reporting process. 3. They are prompted to give an optional description of why they reported the post. 4. The user submits the report. 5. The report is sent to the system administrators who will review the content. 6. A confirmation message is sent to the user letting them know their report went through. 7. The offending user is notified their post was deemed offensive

Extensions:

1a: Either user has blocked the other. The system alerts the user that they have either blocked or been blocked by other users and can not interact with their content

5a. Content has already been deemed acceptable. The system alerts the user that the content has already been reviewed and deemed ok. The report is terminated

2a, 5a: The content no longer exists. The system alerts the user that the content they are reporting no longer exists and terminates the action

Technology and Data List:

* Database to store reports and track which posts or users are flagged for review.
* Notification system to inform both the reporting user and the offending user about the report.
* Tools for administrators to review, approve, or reject reports, and potentially take action (e.g., content removal, user warnings, bans).

Frequency of Occurrence:

Reporting would likely only happen every so often.

Use Cases 17-19: Hadi Khan

Use Case UC17: User Follows a Topic or Category

Stakeholders:

* **Users**: Want to follow topics or categories to receive updates related to their interests.
* **System Administrators**: Ensure that users can follow topics without issues and receive timely notifications or updates.

Preconditions:

* The user is logged in to the platform.
* The system has a functional topic/category subscription mechanism.

Success guarantee:

* The user is successfully subscribed to the selected topic or category, and will receive relevant updates.

Main success scenario:  
 1. The user navigates to the platform’s topic or category section.

1. The user selects a topic or category to follow.
2. The system confirms the selection and subscribes the user to that topic/category..
3. The user starts receiving updates about new posts or content in the followed category.

Extensions:

* **3a**: The user is already following the topic. The system informs the user and does not allow duplicate subscriptions.
* **3b**: Technical issues prevent the system from saving the subscription. The user is prompted to try again later.

Technology and Data List:

* The user interface displays the available topics or categories.
* User inputs are taken via mouse or keyboard to select a topic.

Frequency of Occurrence:

* Occurs whenever a user wants to follow a new topic, typically with moderate frequency

Use Case UC18: User Saves a Post for Later

Stakeholders:

* **Users**: Want to save posts for future reference.
* **System Administrators**: Ensure that saved posts are stored properly and can be accessed by users later.

Preconditions:

* The user is logged in to the platform.
* The system has functionality for saving and retrieving saved posts.

Success guarantee:

* The post is saved to the user’s profile, and the user can view it later from their saved items list.

Main success scenario:

1. The user navigates to a post they are interested in.
2. The user clicks the “Save” button to save the post for later.
3. The system stores the post in the user’s saved items.
4. The user can access the saved post from their profile or saved items section.

Extensions:

* **2a**: The post is already saved. The system informs the user that the post has already been saved.
* **3a**: Technical issues prevent the post from being saved. The system prompts the user to try again later.

Technology and Data List:

* Data about the post is stored in the system, and the save request is submitted through user interaction with a mouse or keyboard.

Frequency of Occurrence:

* Saving posts happens frequently when users want to revisit content later.

Use Case UC19: Admin/Moderator Reviews Reported Content

Stakeholders:

* **Admin/Moderator**: Responsible for ensuring that inappropriate content is handled appropriately.
* **Users**: Report content that violates platform rules.

Preconditions:

* Content has been reported by users.
* Admin or moderator is logged in to their account with appropriate privileges.

Success guarantee:

* The reported content is reviewed, and appropriate actions are taken (such as removal or warning) is taken.

Main success scenario:

1. Admin logs into the platform and navigates to the reported content queue.
2. Admin selects a reported post to review.
3. The system displays details of the report and the content.
4. Admin decides whether to remove the post, issue a warning, or dismiss the report.
5. The system updates the status of the content based on the admin’s decision.

Extensions:

* **4a**: The admin decides to dismiss the report. The content remains on the platform, and the user who reported it is notified.
* **4b**: The admin decides to issue a warning to the user who posted the content. A warning is sent, and the user is notified of the infraction.
* **4c**: The admin removes the content, and it is no longer visible on the platform.

Technology and Data List:

* Content is reviewed via a web interface where reported posts are displayed for review.

Frequency of Occurrence:

* Moderators or admins frequently review reports, depending on the platform’s user base and content activity.

Use Cases 20-23: Jairen Cheek

Use Case UC20: User creates a group

Stakeholders:

Users - Group creators and participants

System Admin - Responsible for making sure groups are properly stored, and removed into database

Preconditions:

1. Author of the group must own and be logged into their account.
2. The creator’s profile must be complete and meet the necessary criteria .
3. The group must be assigned to a valid category or field , ensuring proper classification for users searching for groups. The group name must be unique.

Success guarantee:

The platform ensures that no duplicate groups exist, preventing confusion among users and maintaining a clean, organized structure for the community.

The group is successfully created in the system, and a confirmation message is displayed to the user .

All group details, including the name, description, rules, and settings, are correctly stored and displayed when users visit the group's page.The user who created the group is automatically assigned as the group administrator and has full control over managing the group.The newly created group appears in the appropriate group listings, making it visible and accessible to other users. Depending on the group’s privacy settings, it is accessible to the appropriate users , ensuring the group can fulfill its intended purpose.

Main success scenario:  
1.The user logs into their account.

2. The user clicks on "Groups" section and selects the option to create a new group .

3. The user is presented with the "Create Group" form with fields for group name, description, privacy settings, and any other relevant detail.

4. User fills in group details, providing all necessary information.

5. User submits group creation request. The system checks for preconditions and validates the group.

6. Group is created

7. User now has access and control of group.

Extensions:

1a. Invalid Login Attempt-

The user enters incorrect credentials, the system displays an error message, prompting the user to re-enter their information or reset their password.

2a. Group Creation Limit Exceeded-The user has exceeded their allowed number of group creations within a certain period. The system informs the user that they cannot create more groups at the moment and suggests trying again after the limit resets.

2b. Group Creation Feature Temporarily Unavailable- The system displays an error message and suggests trying again later. The user is notified when the feature becomes available.

3a. Group Name Already Exist- The system displays an error message ("Group name already in use") and prompts the user to choose a different name.

3b. Incomplete/Invalid Information- The user submits the form without filling in all required fields or enters invalid information.The system highlights the missing or incorrect fields and prompts the user to complete or correct them before resubmitting.

4a. Technical Error During Submission - The system informs the user of the error and retries the submission or prompts the user to try again later.

5a. Delay in Group Creation - Due to high system load or other factors, the group creation process is delayed.The system displays a message informing the user of the delay and sends a notification once the group is successfully created.

5b. Group Not Approved by System Admin - The system informs the user that their group is pending approval and will notify them once the admin reviews and approves it.

Technology and Data List:

* Databases: Both user and a group database
* Notification System: Alerts users when their group has been approved
* Admin toolset:Approve and remove groups

Frequency of Occurrence:

Moderately frequent.. Occurs when a user wants to create a new group.

Use Case UC21: User joins a group

Stakeholders:

Users - potential participants in groups & group owners

System Admin - Approval and removal of group members

Preconditions:  
1. The group must already exist in our database.

2.Users must own and login into an Agora account.

Success guarantee:

The user receives a notification confirming that they have successfully joined the group.

The user can now view, participate in, and engage with the group’s content (posts, discussions, events, etc.).

Main success scenario:  
1.User logs in

2. User clicks on “Groups” section of the platform based on interest.

3. User clicks group. The platform displays the group’s description, rules, and membership requirements(Open, Closed, Invite Only)

4. User clicks join.

5.User’s membership gets confirmed.

6. User now has access to selected group.

Extensions:

1a. Invalid Login Attempt-

The user enters incorrect credentials, and the system displays an error message, prompting the user to re-enter their information or reset their password.

2a. No Groups Match User’s Interests-

The platform displays a message indicating no groups match the user's search criteria.

3a. Group is Closed or Invite-Only-

The group is closed or invite-only, requiring the user to submit a request or invitation code.The user submits a request to join the group and waits for the administrator’s approval. Once approved, they are notified and granted access.

3b. Group is Full or Inactive:

The platform informs the user that the group has reached its member limit or has become inactive, and they are unable to join.The platform suggests similar groups that are open and active, allowing the user to explore alternatives.

4a. Group Admin Requires Manual Approval-

For a closed group, the user’s join request is pending administrator approval.

5a. Delay in Membership Confirmation-

Due to technical issues or admin delays, the confirmation takes longer than expected.

The user receives a message informing them of the delay, and the platform provides a timeframe for when their membership will be confirmed.

5b. Membership Request Denied-

The group admin denies the user's membership request. The platform informs the user of the denial, offering feedback on why and suggesting other groups they might be interested in.

Technology and Data List:

Database: User, and Group database

Notification System: Alert users when they have joined a group

Admin tools: Add & Remove users from groups

Frequency of Occurrence:

Occurs every time a user wants to join a group, highly common.

Use Case UC22: User Participates in Polls or Surveys

Stakeholders:

Participant: The user who participates in polls or surveys.

Poll/Survey Creator: The individual or group who creates the polls or surveys.

System Admins: Responsible for overseeing the platform and ensuring that polls and surveys comply with platform rules.

Preconditions:  
1. User is logged in

2. Poll/Survey is active

3. Poll/Survey has not already been answered

Success guarantee:

The user's response is successfully recorded and stored in the system.

The system ensures that only valid, complete responses are submitted , and users are notified if any fields are incomplete or invalid.

Main success scenario:

1. User logs into account.
2. User navigates to poll via dashboard or group
3. User reads and responds to poll
4. User clicks submit button to submit response
5. System confirms submission and displays confirmation message saying “Your response has been submitted”
6. User participation is logged

Extensions:

1a. System prompts user to login - System prevents user from participating in poll, redirects user to login

2a. Closed poll - System tells the poll is no longer accepting response, and displays results of the poll

4a. Invalid response - System checks the input, finds it incomplete and tells users to recheck response before submitting.

Technology and Data List:

Databases: Pool /survey database storing data such as responses, metadata, and participation logs.

Feedback Display: Displays real-time results or feedback to users once they’ve submitted their response

Frequency of Occurrence:

Varies based on size and activity of groups the user interacts with, somewhat frequent.

Use Case UC23: User hosts a Poll or Survey

Stakeholders:

Users: creates ,manages and participates in the poll or survey.

System Admin: managing and overseeing the platform to ensure polls or surveys adhere to platform policies and guidelines

Preconditions:  
1. User is logged in

2. User must be a member in a group

3. User has appropriate permissions, some polls/surveys might be limited to certain user roles such as professionals or admins.

Success guarantee:

Poll/Survey is successfully created and published.

User receives confirmation.

Poll/Survey Appears in Relevant Listings

Main success scenario:  
1. User logs in.

2. User navigates to group they want to post the poll.

3. User fills out poll/survey details

The user enters all required information, including: title,question , response type and privacy setting.

4. User submits poll by clicking “Submit” button. The system checks if user has valid input (complete, valid question type, unique title)

5. Poll is successfully created, a confirmation message is displayed confirming the creation of poll.

6. User can manage and edit, look at analytics and monitor responses.

Extensions:

4a. User submits invalid input data - System detects the user has tried to submit an incomplete poll. Prompts the user to fix input.

6a. Editing restrictions - The system prevents the user from editing certain aspects of the poll

Technology and Data List:

Poll/Survey database: stores poll details, response and results

Admin tools: Allow users to create ,edit , publish polls or surveys.

Analytics and reporting tools: Provide data visualizations and analysis for poll results.

Frequency of Occurrence:

Varies based on activity and size of group poll is posted. Somewhat frequent.

Use Case UC24: User logs out

Stakeholders:

User: Wants to securely log out from their account to prevent unauthorized access.

System: Ensures a smooth logout process while maintaining security and session integrity.

Preconditions:  
The user is logged into their account.

The system is functioning properly, allowing users to interact with their account settings.

Success guarantee:

The user is successfully logged out of the platform.

The user's session is terminated, and no further access is possible without logging in again.

The system provides confirmation of successful logout.

Main success scenario:

1. The user navigates to their account or profile settings.
2. The user selects the "Log Out" option.
3. The system confirms that the user intends to log out (if applicable, especially on mobile or desktop apps).
4. The system terminates the user session, logging the user out.
5. The system redirects the user to the login page or homepage, displaying a message confirming that they have successfully logged out.

Extensions:

2a. The user closes the browser or app without logging out. The system automatically terminates the session after a timeout or upon detecting inactivity.

4a. There is a system error during logout. The system informs the user that the logout was unsuccessful and suggests they try again.

4b. The user logs out from multiple devices. The system ensures all active sessions for that user are terminated across devices.

Technology and Data List:

* User session management system to track and terminate active sessions.
* Security measures to clear authentication data (cookies, tokens, etc.) after logout.
* Database logging to track successful and unsuccessful logout attempts for security audits.

Frequency of Occurrence:

This process occurs frequently, as users regularly log out for security reasons, especially when accessing the platform on shared or public devices.