
GROUP 01 - CHÚ HỀ LÀM PHẦN MỀM

LOGBLOCK SOCIAL MEDIA SERVICE
Use-Case Specification

Version 1.4

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

Revision History

Date	Version	Description	Author
08/11/2024	1.0	Use-case Diagram scenarios initialization.	Ngũ Kiệt Hùng
12/11/2024	1.1	UC2, UC3, UC9	Trình Cao An
13/11/2024	1.2	UC1, UC4: Nguyễn Thế Thanh Long. U6, UC7: Trần Thanh Long. UC10, UC11, UC12: Trần Nguyễn Nhật Cường.	Nguyễn Thế Thanh Long, Trần Thanh Long, Trần Nguyễn Nhật Cường
14/11/2024	1.3	UC5, UC8: Ngũ Kiệt Hùng	Ngũ Kiệt Hùng
27/11/2024	1.4	Overhauling all Use Cases (UC01-UC24), Reducing Diagrams to only two main diagrams depicting Core Use Cases and Specialized Use Cases. More detailed specification of each Use Cases.	Ngũ Kiệt Hùng

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

Table of Contents

Related Data Tables:	7
1. UC01: Account Authentication Through Google Authentication	8
1.1 Brief Description	8
1.2 Flow of Events	9
1.3 Preconditions	9
1.4 Postconditions	9
2. UC02: Account Deletion	10
2.1 Brief Description	10
2.2 Flow of Events	10
2.3 Preconditions	10
2.4 Postconditions	10
3. UC03: Profile Biography Updating	11
3.1 Brief Description	11
3.2 Flow of Events	11
3.3 Preconditions	12
3.4 Postconditions	12
4. UC04: Updating Biography Description	12
4.1 Brief Description	12
4.2 Flow of Events	12
4.3 Preconditions	12
4.4 Postconditions	13
5. UC05: Updating Profile Picture	13
5.1 Brief Description	13
5.2 Flow of Events	13
5.3 Preconditions	14
5.4 Postconditions	14
6. UC06: Updating Profile Display Name	14
6.1 Brief Description	14
6.2 Flow of Events	14
6.3 Preconditions	14
6.4 Postconditions	14
7. UC07: Original Post Pinning	15
7.1 Brief Description	15
7.2 Flow of Events	15
7.3 Preconditions	15
7.4 Postconditions	15

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

8. UC08: Post Data Retrieval	15
8.1 Brief Description	15
8.2 Flow of Events	15
8.3 Preconditions	16
8.4 Postconditions	16
9. UC09: View Targeted Profile Information	16
9.1 Brief Description	16
9.2 Flow of Events	17
9.3 Preconditions	18
9.4 Postconditions	18
10. UC10: Target Profile Blocking	18
10.1 Brief Description	18
10.2 Flow of Events	18
10.3 Preconditions	19
10.4 Postconditions	19
10.5 Extension Points	19
11. UC11: Connect or Disconnect to Target Profile	19
11.1 Brief Description	19
11.2 Flow of Events	19
11.3 Preconditions	20
11.4 Postconditions	20
12. UC12: News Feed Post Generation Request	20
12.1 Brief Description	20
12.2 Flow of Events	20
12.3 Preconditions	21
12.4 Postconditions	21
13. UC13: Exploration Feed Post Generation Request	21
13.1 Brief Description	21
13.2 Flow of Events	22
13.3 Preconditions	22
13.4 Postconditions	22
14. UC14: User Delete Original Posting	23
14.1 Brief Description	23
14.2 Flow of Events	23
14.3 Preconditions	23
14.4 Postconditions	23
15. UC15: User Create Posting	24
15.1 Brief Description	24

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

15.2 Flow of Events	24
15.3 Preconditions	26
15.4 Postconditions	26
16. UC16: User Edit Posting	26
16.1 Brief Description	26
16.2 Flow of Events	26
16.3 Preconditions	28
16.4 Postconditions	28
17. UC17: Post Upvoting	28
17.1 Brief Description	28
17.2 Flow of Events	28
17.3 Preconditions	29
17.4 Postconditions	29
18. UC18: User Commenting	29
18.1 Brief Description	29
18.2 Flow of Events	29
18.3 Preconditions	30
18.4 Postconditions	30
19. UC19: Profile Searching	30
19.1 Brief Description	30
19.2 Flow of Events	30
19.3 Preconditions	31
19.4 Postconditions	31
20. UC20: Post Report	31
20.1 Brief Description	31
20.2 Flow of Events	31
20.3 Preconditions	31
20.4 Postconditions	32
21. UC21: Expert Suggested Solution Creation	32
21.1 Brief Description	32
21.2 Flow of Events	32
21.3 Preconditions	33
21.4 Postconditions	34
22. UC22: Dedicated Page to Monitor Reported Objects	34
22.1 Brief Description	34
22.2 Flow of Events	34
22.3 Preconditions	34
22.4 Postconditions	34

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

23. UC23: Remove Post	34
23.1 Brief Description	34
23.2 Flow of Events	34
23.3 Preconditions	35
23.4 Postconditions	35
24. UC24: Remove Account	35
24.1 Brief Description	35
24.2 Flow of Events	35
24.3 Preconditions	36
24.4 Postconditions	36

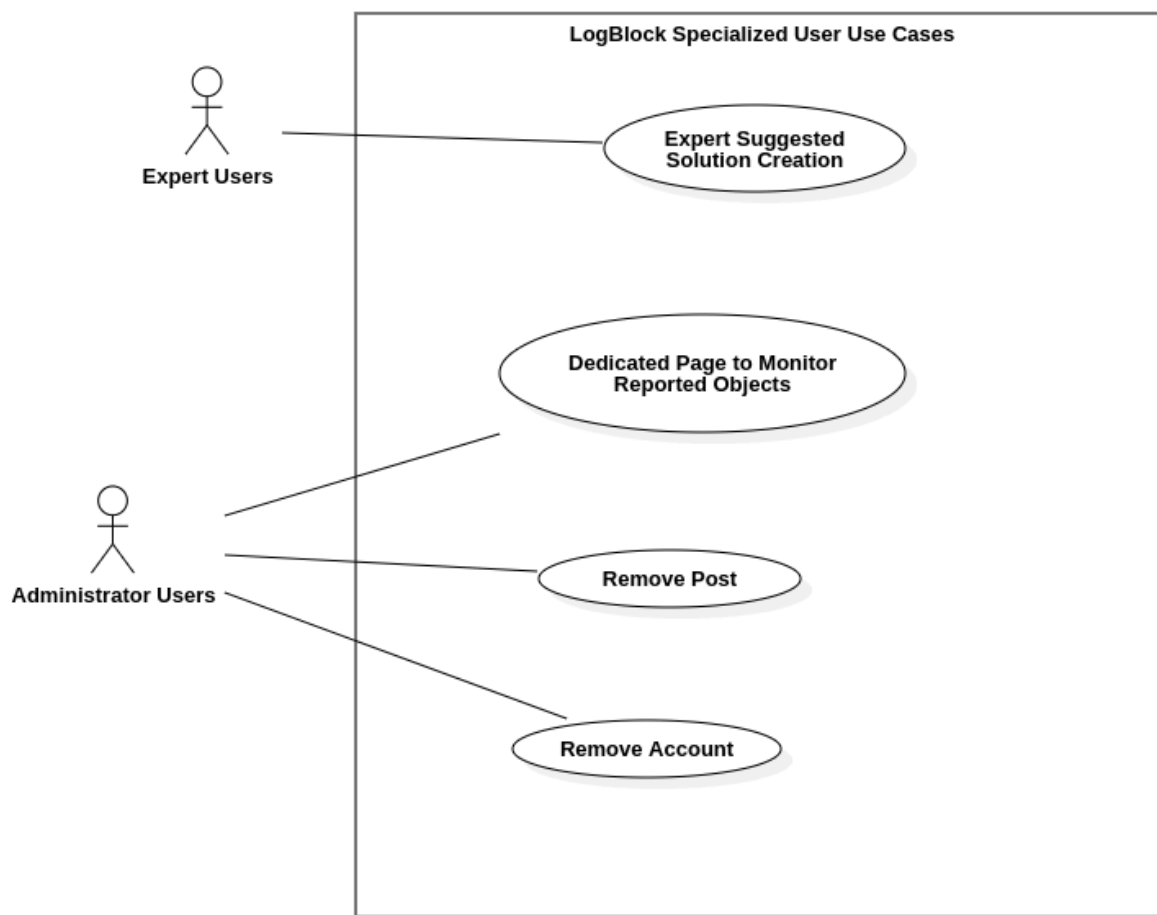
LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

Use-Case Specification: LogBlock

Based on The Core Functionalities of primary User Classes, *Two grouped* Use Case Diagrams depicting the core functionalities of LogBlock



LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	



Related Data Tables:

1. A User Profile in the **Profile Table** entry has the following fields:
 - User Profile unique identifier (UserID): A unique identifier to identify each User globally.
 - User Email address (UserEmail): The email address which the User uses to authenticate to LogBlock.
 - User display name (DisplayName): The user displays name, can be duplicated.
 - User Biography Description (BiographyDesc): A paragraph of text in Markdown format, that will be used to display a short description of the User in the User's Personal Profile Card.
 - User Biography Profile Image (ProfileImage): The main pictures that will be displayed as the User's image-representation.
 - User privilege level (PrivLevel): The privilege of the User, with 0 denoting a Registered User, 1 is an Expert User and 2 is an Administration User.
 - Pinned Posts list (PinnedPosts): A list of Post ids foreign keys to the **Posting Table** that are originally authored by the User.
 - User Blocked Profiles list (BlockedProfiles): A list of Profile ids foreign keys to the **Profile Table** that the User has already blocked.
2. A Post entry in the **Posting Table** has the following fields:
 - Post unique identifier (PostID): a unique identifier to identify each post globally.

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

- Post original author (OriginalAuthor): a foreign key to the UserID field of the **Profile Table**, describes the first author of the Post.
 - Post caption (PostCaption): a textual field in the format of Markdown text, denoting the Post captioning.
 - Post attached media (PostMedia): a list of URLs to the media attached to this Post.
 - Post creation date (PostCreation): the date of which the Post is originally created.
 - Post latest update date (PostLastUpdate): the date of which the Post was last edited.
 - Post tagging list (PostTagging): the list of foreign keys to the UserID field of the **Profile Table**, describes the list of User that has been tagged to this Post.
 - Post upvotes(PostUpvote): the list of foreign keys to the UserID field of the **Profile Table**, describes the list of Users that have upvoted this post.
3. A Comment entry in the **Commenting Table** has the following fields:
- Post unique identifier (PostID): a foreign key to a Post entry in the **Posting Table**, which the comment belongs to.
 - Comment unique identifier (CommentID): a unique identifier to identify unique comments in each Post.
 - Comment original author (CommentAuthor): a foreign key to the UserID field of the **Profile Table**, describes the author of the Comment.
 - Comment caption (CommentCaption): a textual field in the format of Markdown text, denoting the Comment content..
 - Comment creation date (CommentCreation): the date of which the Comment is originally created.
4. A Connection entry in the **Connection Table** has the following fields:
- Connection initiator (Connector): The UserID of the User who initially reaches out and connects with the connected target.
 - Connection target (ConnectedTo): The UserID of the Targeted Connecting User.
 - Connected date (ConnectionDate): The date on which the connection is initiated.
5. A Expert Suggested Solution (ESS) entry in the **Expert Solutions Table** has the following fields:
- Post unique identifier (PostID): a foreign key to a Post entry in the **Posting Table**, which the comment belongs to.
 - Solution unique identifier (SolID): a unique identifier to identify unique Solutions in a Post.
 - Solution author (SolAuthor): a foreign key to the UserID field of the **Profile Table**, describes the author of the Solution.
 - Solution caption (SolCaption): a textual field in the format of Markdown text, denoting the Solution captioning.
 - Solution attached media (SolMedia): a list of URLs to the media attached to this Solution.
 - Solution creation date (SolCreation): the date of which the Solution is originally created.
 - Solution latest update date (SolLastUpdate): the date of which the Solution was last edited.
6. A Report entry in the **Reporting Table** has the following fields:
- Report unique identifier (ReportID): The unique identifier to identify each report globally.
 - Reporter (Reporter): The User who made the reports.
 - Reported Post (ReportPostID): The identifier of the Post that has been reported.
 - Reported date (ReportDate): The date on which the report is created.

1. UC01: Account Authentication Through Google Authentication

1.1 Brief Description

This use case illustrates the behavior of the service regarding Guest Users and Registered users, as well as User Classes inherited from Registered User, when entering the domain of the service.

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

1.2 Flow of Events

■ Basic Flow

- **Step 1:** User access the site through the registered domain
- **Step 2:** The Service displays a button that redirects the User to Google's OAuth Service.
- **Step 3:** User follows the steps guided by Google's OAuth Service.
- **Step 4:** Finishing the process, Google's OAuth Service will return a User Access Token, which LogBlock will store on User's browser local storage.
- **Step 5:** The Service records that the process is successful, The Service locates the appropriate Profile associated with the email from Google's OAuth and two JWT tokens are generated, one for authorization and one for authorization rotating (renewal) process for that Profile Access.
- **Step 6:** The User is redirected to the News Feed.

■ Alternative Flows

First Alternative Flow - Authenticated User

- **Condition:** the user is already logged in, meaning the User has a valid access token that has not expired, or if it is already expired, the accompanying rotating token will be used to renew the access token.
- **At step 1:** The Service has recognized that the user is authenticated and tokens held by the User are valid. The Service skips **Step 2 to 5** and proceeds with **Step 6**.

Second Alternative Flow - Failure of authentication

- **Condition:** Google's OAuth authentication process is unsuccessful.
- **At step 4:** if the authentication process fails, LogBlock will display the cause of failure based on Google's OAuth feedback and restart the process at **step 1**.

Third Alternative Flow - New User Registration

- **Condition:** The User's email obtained from Google's OAuth does not exist in the Service record.
- **At step 5:** if LogBlock cannot locate an appropriate Profile associated with the email from Google's OAuth, a new Profile will be created with default biographical information by the following steps:
- **Step 5.1:** LogBlock creates a new User Profile entry in the Profile Database with UserID generated by increasing the ID of the latest User Profile.
- **Step 5.2:** LogBlock assigns the User's Google Email address to the userEmail field.
- **Step 5.3:** LogBlock assigns a random generated string to the DisplayName field.
- **Step 5.4:** LogBlock assigns an empty value to the BiographyDesc field.
- **Step 5.5:** LogBlock assigns the ProfileImage field with the default Profile Image provided by LogBlock.
- **Step 5.6:** LogBlock assigns the PrivLevel field with value 0 (Registered User).
- **Step 5.7:** LogBlock assigns the PinnedPosts and BlockedProfiles with empty lists.
- The steps continue from **Step 5** of the **Basic Flow**.

1.3 Preconditions

- The User accesses the main site URL.
- The User must have a Google Account.
- The Profile Table must be able to insert a new entry into.

1.4 Postconditions

- After successful login or registration, the user is directed to the News Feed.
- If any of **step 3 or 4 fails**, LogBlock will display an error message of failure cause and restart at **step 1**.

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

2. UC02: Account Deletion

2.1 Brief Description

This Use Case describes the process of how a User can Delete the Profile that is associated with their email address and the process of deletion on LogBlock's side.

2.2 Flow of Events

- Basic Flow
 - **Step 1:** User access the Setting Page from the Profile Page.
 - **Step 2:** User uses the Deletion Button.
 - **Step 3:** LogBlock prompts a confirmation for Account Deletion.
 - **Step 4:** User accepts the prompt by clicking on the Accept button.
 - **Step 5:** LogBlock searches for Original Posts where the OriginalAuthor field matches the User's id and removes those entries from the Posting Table.
 - **Step 6:** LogBlock searches for Post entries whose PostTagging field includes the User's id and removes tagging of the User from those Post entries in the Posting Table.
 - **Step 7:** LogBlock searches for Comment entries whose OriginalAuthor field matches the User's id and removes those comments entries from the Commenting Table.
 - **Step 8:** LogBlock removes the Connection entries in which the User's id is present in either the Connector field or the ConnectedTo field.
 - **Step 9:** LogBlock finds all Solution entries that are authored by the Targeted User Profile from the Expert Solutions Table, and removes all entries that satisfy the criteria.
 - **Step 10:** LogBlock removes this User's Profile entry from the Profile Table.
 - **Step 11:** LogBlock displays an Account Deletion successful message.
 - **Step 12:** LogBlock redirects the User back to the landing page without any tokens.
- Alternative Flows

First Alternative Flow - User Cancel Deletion Prompt

- **Condition:** the user does not accept the confirmation prompt at **Step 4**.
- **At Step 4**, if the User does not accept the confirmation, the dialog box will disappear and the User is presented with the Setting Page.

Second Alternative Flow - Alternative Prompt Acceptance method

- **Condition:** The User reaches **Step 3**.
- **At Step 4**, Aside from using the Accept Button, the User can also press Enter on the keyboard in order to accept the prompt.

2.3 Preconditions

- The User must be logged in to the platform.
- The User profile must exist on the platform.
- The User must access the Use Case from the Setting Page.

2.4 Postconditions

- Upon Account Deletion process succession, All User Profile's related content will be deleted.
- The User will be redirected to the landing page without any authorization tokens.

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

3. UC03: Profile Biography Updating

3.1 Brief Description

This Use Case presents the included steps in which a Registered User and User classes inherited from Registered User can proceed to update their profile basic biography.

3.2 Flow of Events

- Basic Flow
 - **Step 1:** The User accesses the Profile Page.
 - **Step 2:** The User clicks on the Edit button on the Profile Card.
 - **Step 3:** The User is prompted with **Updating Biography Description Use Case, Updating Profile Picture Use Case** and **Updating Profile Display Name Use Case**.
 - **Step 4:** The User follows flows from the aforementioned Use Cases in any order.
 - **Step 5:** After completing the changes, the User clicks on the Save button to save the changes.
 - **Step 6:** LogBlock logs all changes, and proceeds to locate the User Profile entry that matches the User's id in the Profile table.
 - **Step 7:** LogBlock updates the BiographyDesc field in the correct User Profile entry.
 - **Step 8:** LogBlock stores the User Uploaded Profile Picture into a bucket storage, and deletes the old profile picture of the User based on the current URL value of the ProfileImage field in the User Profile entry.
 - **Step 9:** LogBlock updates the ProfileImage field in the Profile Table with the picture's address in the bucket storage.
 - **Step 10:** Back-End Service reports successful status to LogBlock
 - **Step 11:** LogBlock displays Biography Updating has been successfully processed.
 - **Step 12:** User acknowledges the status by clicking the OK button.
 - **Step 13:** LogBlock refreshes the page to display newly corrected information.
- Alternative Flows

First Alternative Flow - Invalid Profile Picture Editing:

- **Condition:** The **Updating Profile Picture Use Case** reports failure due to invalid picture format.
- The System displays an error message and prompts the user to correct it.
- The System blocks the Save button due to the incorrect picture format.

Second Alternative Flow - Invalid Biography Description Editing:

- **Condition:** The **Updating Biography Description Use Case** reports failure due to invalid invalid characters input from the User.
- LogBlock displays an error message and prompts the user to correct it.
- LogBlock blocks the Save button due to the incorrect picture format.

Third Alternative Flow - Invalid Display Name Editing:

- **Condition:** The **Updating Display name Use Case** reports failure due to invalid invalid characters input from the User.
- LogBlock displays an error message and prompts the user to correct it.
- LogBlock blocks the Save button due to the incorrect picture format.

Fourth Alternative Flow - Discard Editing:

- **Condition:** the User declines changes made to the Biography Updating Process.
- At any moment from **step 3 to step 5**, the User can choose to discard all changes by clicking on the Discard button, which will discard all changes to the related editing fields and returns the User to the Profile Page.

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

Fifth Alternative Flow - Alternative Prompt Acceptance method

- **Condition:** The User reaches **Step 5**,
- **At Step 5**, Aside from using the Save Button, the User can also press Enter on the keyboard in order to save the changes.

Sixth Alternative Flow - User does not change the Profile Image

- **Condition:** The User does not upload any image during **Step 3 to step 5**.
- **At step 8**, LogBlock skips **Step 8 and Step 9**, continuing at **Step 10**.

3.3 Preconditions

- The User must be logged in to the platform.
- The User profile must exist on the platform.
- The Back-End Service bucket storage must have enough storage to store the newly uploaded picture.
- The preconditions of Use Cases that are extended to this Use Case must all be satisfied.

3.4 Postconditions

- Upon successful profile update, the System displays the updated profile with a new biography description, profile picture.
- If the User discards all changes, the system returns the User to the Profile Page without changing anything.

4. UC04: Updating Biography Description

4.1 Brief Description

This use case illustrates the sub-process of recording the changes to the User's profile biography description section.

4.2 Flow of Events

■ Basic Flow

- **Step 1:** The User reaches **Step 4** from **UC03**.
- **Step 3:** LogBlock locates the User Profile entry matching the User's id in the Profile Table.
- **Step 4:** LogBlock retrieves the current BiographyDesc field of the User Profile entry from the previous step.
- **Step 5:** The User is prompted with a text field, pre-filled with the current biography description retrieved from the previous step.
- **Step 6:** The User makes changes to the text field.
- **Step 7:** LogBlock registers changes made by the User, which are saved to User's browser local storage for backup.

First Alternative Flow - Invalid Input Text

- **Condition:** The User enters a non-supported Unicode character
- **At step 2**, if the User entered a non UTF-8 character, the border stroke of the text field will turn red and an error message will appear, stating that the User has entered a non-supporting character and prompts the User to change it.
- The Use Case reports Invalid Biography Description Editing to **UC03**.

4.3 Preconditions

- The User must be logged in to the platform.
- The User profile must exist on the platform.
- The User must have already reached **Step 4** from **UC03**.
- The User's browser must be able to edit the text field presented by LogBlock.

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

4.4 Postconditions

- Upon each character entered, LogBlock will save the editing states into the User's browser local storage for backup, in case the editing gets interrupted by external means.
- The Use Case will report Invalid Input Text if any of the entered characters is a non-UTF-8 character.

5. UC05: Updating Profile Picture

5.1 Brief Description

This Use Case describes the sub-process of updating the User's Biography, specifically, this Use Case handles with Updating Profile Picture.

5.2 Flow of Events

■ Basic Flow

- **Step 1:** The User reaches **Step 4** from **UC03**.
- **Step 2:** The User is presented with a dialog box, with a section that the user can drag and drop pictures from the User's device to the dialog box.
- **Step 3:** The User's browser uploads the picture file to LogBlock's temporary memory.
- **Step 4:** LogBlock receives the picture and processes the image, considering the format of the picture file, the size of the image.
- **Step 5:** Upon successful inspection, LogBlock displays a successful upload of the picture to the User for confirmation.

■ Alternative Flows

First Alternative Flow - Invalid Picture Format

- **Condition:** at **Step 3**, the User's uploaded file is not the kind of picture that LogBlock can process (PNG, bitmap, JPEG).
- The dialog box's border stroke will turn red. LogBlock will display an error message, stating that the User has uploaded a non-supported picture format and prompts the User to change it.
- The Use Case reports Invalid Profile Picture Editing to **UC03**.

Second Alternative Flow - Invalid Picture Size

- **Condition:** at **Step 3**, the User's uploaded file is too large. (maximum size is 512x512 pixels)
- The dialog box's border stroke will turn red. LogBlock will display an error message, stating that the User has uploaded a picture whose size is too large and prompts the User to change it.
- The Use Case reports Invalid Profile Picture Editing to **UC03**.

Third Alternative Flow - Alternative Profile Picture Uploading by URL

- **Condition:** at **Step 2**, the User is also presented with a text field, prompting the User to enter the URL of a picture. From here, the steps are followed:
- **Step 3:** The User clicks on the Retrieve button next to the URL field.
- **Step 4:** LogBlock receives the picture URL and tries to get the media through that URL.
- **Step 4.1:** If the picture cannot be retrieved, the dialog box's border stroke will turn red, LogBlock will display an error message, stating that the User has entered a non-retrievable URL and prompts the User to change it or try other means.
- **Step 4.2:** If the picture is successfully retrieved, the process proceeds from **Step 4** as mentioned in the **Basic Flow**.

5.3 Preconditions

- The User must be logged in to the platform.

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

- The User profile must exist on the platform.
- The User must have already reached **Step 4** from **UC03**.
- The User must be able to upload a picture using the browser's built in media uploading mechanism.
- Alternatively, the User can upload the picture by specifying a picture URL, requiring the URL to be able to edit the text field.

5.4 Postconditions

- Upon success, LogBlock will save the User's uploaded picture as temporary memory, which might be submitted for update completion in **UC03**.
- Upon success, LogBlock will display a successful dialog for User to acknowledge the successful upload.

6. UC06: Updating Profile Display Name

6.1 Brief Description

This use case illustrates the sub-process of recording the changes to the User's profile display name.

6.2 Flow of Events

■ Basic Flow

- **Step 1:** The User reaches **Step 4** from **UC03**.
- **Step 2:** LogBlock locates the User Profile entry matching the User's id in the Profile Table.
- **Step 3:** LogBlock retrieves the current DisplayName field of the User Profile entry from the previous step.
- **Step 2:** The User is prompted with a text field, pre-filled with the current display name retrieved from the previous step.
- **Step 3:** The User makes changes to the text field.
- **Step 4:** LogBlock registers changes made by the User, which are saved to User's browser local storage for backup.

First Alternative Flow - Invalid Input Text

- **Condition:** The User enters a non-supported Unicode character
- **At step 2,** if the User entered a non UTF-8 character, the border stroke of the text field will turn red and an error message will appear, stating that the User has entered a non-supporting character and prompts the User to change it.
- The Use Case reports Invalid Display Name Editing to **UC03**.

6.3 Preconditions

- The User must be logged in to the platform.
- The User profile must exist on the platform.
- The User must have already reached **Step 4** from **UC03**.
- The User's browser must be able to edit the text field presented by LogBlock.

6.4 Postconditions

- Upon each character entered, LogBlock will save the editing states into the User's browser local storage for backup, in case the editing gets interrupted by external means.
- The Use Case will report Invalid Input Text if any of the entered characters is a non-UTF-8 character.

7. UC07: Original Post Pinning

7.1 Brief Description

This Use Case describes the process of how a Registered User and User classes inherited from Registered User can pin certain posts to be displayed on their Profile.

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

7.2 Flow of Events

■ Basic Flow

- **Step 1:** The User accesses the Personal Profile Page of the User.
- **Step 2:** LogBlock displays the User's Personal Profile and Posts.
- **Step 3:** The User clicks on the targeted Post to be pinned.
- **Step 4:** LogBlock displays the targeted Post as a box.
- **Step 5:** The User clicks on the Pin Button on the Post.
- **Step 6:** LogBlock acknowledges the Pinning command, saving the post id.
- **Step 7:** LogBlock finds the User's entry in the Profile Table.
- **Step 8:** LogBlock finds the PinnedPosts attribute in the User's Profile entry, and appends the targeted post id into the value of the attribute.
- **Step 9:** LogBlock informs the User of the successful pinning.
- **Step 10:** LogBlock refreshes the Profile Page for User to view the recent change.

First Alternative Flow - Pinned Post is already pinned

- **Condition:** The targeting post to pin has its id already present in the User's PinnedPosts attribute.
- **At Step 8,** if the post id already presents in the list, the previous value is removed, the list shortened to compensate for the empty slot, and **Step 8** proceeds as normal.

7.3 Preconditions

- The User must be logged in to the platform.
- The User profile must exist on the platform.
- The targeting Post id must exist in the Posting Table and the OriginalAuthor field of the entry must match the User's id in order to pin it to the User's Personal Profile Page.

7.4 Postconditions

- Upon success, the PinnedPosts field will be updated accordingly to include the targeted post.
- The user's Personal Profile view is updated to display the new changes.

8. UC08: Post Data Retrieval

8.1 Brief Description

This Use Case details the process of how LogBlock will retrieve data from the Posting Table. This Use Case also acts as the Sub-Use Case used by other major Use Cases.

8.2 Flow of Events

■ Basic Flow

- **Step 1:** User requests for a Post Data Retrieval by appending the Post id after LogBlock's top level URL.
- **Step 2:** LogBlock searches for the Post entry from the Posting Table that matches the Post id.
- **Step 3:** LogBlock saves the Post entry result.
- **Step 5:** LogBlock post-checks if the saved Post entry contains the OriginalAuthor that is blocked by the User and removes entries that satisfy the criteria.
- **Step 6:** LogBlock finds the User Profile entry that has UserID matches the OriginalAuthor field.
- **Step 7:** LogBlock saves the ProfileImage and DisplayName of the User Profile entry retrieved from the previous step.
- **Step 8:** LogBlock finds all Comment entries in the Commenting Table that have the PostID field matching the current Post's id.
- **Step 9:** LogBlock aggregates the query results and retrieves the number of entries as *Comment count*.
- **Step 10:** LogBlock displays the Post Card, containing at the header: the OriginalAuthor's Profile Image, Display Name and the Post Creation date. The Post Card header also includes a More button, allowing Users to click on to open a small dialog with Report option.

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

- **Step 11:** LogBlock generates the Post body, containing: the Post Caption, rendered through a Markdown engine, attached media below the Post Caption
- **Step 12:** LogBlock generates the Post footer, containing: the Comment button, showing the *comment count* and can be clicked on to redirect to the Post Comment section. Alongside the Comment button is the number of upvotes of the Post by counting the element in the PostUpvote list and the Upvote or Remove Upvote button.
- Alternative Flows

First Alternative Flow - Request From Other Use Cases

- **Condition:** The Post Data Retrieval request is commanded by other Use Cases on behalf of the User.
- **At Step 1,** The other use cases will provide the requesting User's id and the requesting Post's id.
- **Step 2 to step 9** follows the basic flow.
- **Step 10:** The Post entry result, the Comment entries results list, the OriginalAuthor ProfileImage, the OriginalAuthor DisplayName is sent to the Use Case for further processing.

Second Alternative Flow - Unrecognizable Post id

- **Condition:** LogBlock cannot find the requested Post entry with the matching Post id in the Posting table.
- **At Step 2,** if the searching is failed, LogBlock will display an error 404 message for the User, and a button to prompt the User back to the News Feed.

Third Alternative Flow - Original Author additional items:

- **Condition:** The Post's OriginalAuthor id matches with the User's id.
- **At Step 10,** the More button at the Post Card header also allows the User to edit or delete the Post.

8.3 Preconditions

- The User must be logged in to the platform.
- The User profile must exist on the platform.
- The requested Post id must be of a valid Post entry in the Posting Table.

8.4 Postconditions

- The Post Card, including relevant information of a Post, is displayed to User
- For requests coming from other Use Cases, the relevant Post data is sent to the Use Case for further processing.

9. UC09: View Targeted Profile Information

9.1 Brief Description

This Use Case describes how a Registered User and User classes inherited from Registered User can view the User's own Profile Page (labeled Personal Profile Page) and also view other Users Profile Page (labeled External Profile Page).

9.2 Flow of Events

- Basic Flow
 - **Step 1:** The User is redirected to the Targeted Profile Page.
 - **Step 2:** LogBlock finds the Targeted Profile id from the Profile Table.
 - **Step 3:** LogBlock retrieves the Profile Biography, Profile Picture URL and Pinned Posts from the Profile Table.
 - **Step 4:** LogBlock finds: entries in the Connection Table where the Connector field matches the User id;

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

entries in the Connection Table where the ConnectedTo field matches the User id.

- **Step 5:** LogBlock saves the entries where the User is the Connector as *Connecting to* count; the entries where the User is Connected by External User as *Connected by* count.
 - **Step 6:** LogBlock displays the Profile Card, including the Targeted User's Profile Picture, Biography Description, the *Connecting to* and *Connected by* counts.
 - **Step 7:** LogBlock finds all Post entries whose OriginalAuthor field matches the Targeted Profile id from the Posting Table by following the **UC08: Post Data Retrieval Use Case**,
 - **Step 8:** LogBlocks aggregates Posts id that the Targeted Profile is the Original Author into a list.
 - **Step 9:** The list is sorted, with pinned posts placed before non-pinned posts. The order in which pinned posts are sorted is matched with the Pinned Posts list's order. Non-pinned posts are sorted in the same manner.
 - **Step 10:** LogBlock displays the posts to the User according to the sorted order:
 - **Step 11:** LogBlock displays the Post Card, containing at the header: the OriginalAuthor's Profile Image, Display Name and the Post Creation date. The Post Card header also includes a More button, allowing Users to click on to open a small dialog with Report option.
 - **Step 12:** LogBlock generates the Post body, containing: the Post Caption, rendered through a Markdown engine, attached media below the Post Caption
 - **Step 13:** LogBlock generates the Post footer, containing: the Comment button, showing the *comment count* and can be clicked on to redirect to the Post Comment section. Alongside the Comment button is the number of upvotes of the Post by counting the number of elements in the PostUpvote list; and the Upvote or Remove Upvote button.
 - **Step 14:** Pinned Posts are also attached with a Pinned icon besides the Targeted Profile's name on the Post's Card
- Alternative Flows

First Alternative Flow - Targeted Profile Page is User's Personal Profile Page

- **Condition:** The User accesses the User's Personal Profile Page.
- **At Step 7**, LogBlock also displays the Edit button in order for the User to update the User's Biography.

Second Alternative Flow - Original Author additional items:

- **Condition:** The Post's OriginalAuthor id matches with the User's id.
- **At step 11**, the More button at the Post Card header also allows the User to edit or delete the Post.

Third Alternative Flow - Targeted Profile Page is an External Profile Page

- **Condition:** The User accesses an External Profile Page.
- **At Step 7**, LogBlock also displays the **Block** button in order for the User to Block the Target Profile
- **At Step 7**, LogBlock also displays the **Connect** button if the User has not Connected to the External Profile Page, otherwise, LogBlock will display the **Unconnect** button.

Fourth Alternative Flow - Targeted Profile has Expert Privilege

- **Condition:** The External Targeted Profile is of privilege level 1.
- **At Step 11**, LogBlock also displays an "*Expert*" tag next to the External Targeted User's Display Name.

Fifth Alternative Flow - Targeted Profile has Administration Privilege

- **Condition:** The External Targeted Profile is of privilege level 2.
- **At Step 11**, LogBlock also displays an "*Administration*" tag next to the External Targeted User's Display Name.

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

9.3 Preconditions

- The User must be logged in to the platform.
- The User profile must exist on the platform.
- The Targeted Profile id must be of a valid User Profile that exists on the platform.
- The Targeted Profile id must not be present in the User's BlockedProfiles attribute value in the User Profile entry in the Profile Table.
- The preconditions of Use Cases that are included in this Use Case must all be satisfied.

9.4 Postconditions

- Upon success, the Targeted Profile Page is displayed along with the Targeted Profile's Biography Information, The Posts that are originally authored by the Targeted Profile Page.

10. UC10: Target Profile Blocking

10.1 Brief Description

This Use Case describes the process of how a Registered User and User classes inherited from Registered User can block another User's Profile (labeled the External Target Profile), preventing contents from the Target Profile to be presented to the User.

10.2 Flow of Events

- Basic Flow
 - **Step 1:** The User navigates to the External Target User's Profile Page.
 - **Step 2:** The User clicks on the **Block** button.
 - **Step 3:** LogBlock prompts the User for blocking action confirmation with a dialog box.
 - **Step 4:** The User accepts the prompt by clicking on the Accept button.
 - **Step 5:** LogBlock finds the User Profile entry in the Profile Table.
 - **Step 6:** LogBlock adds the External Target Profile id to the BlockedProfiles attribute value of the User Profile entry.
 - **Step 7:** LogBlock informs the User of the successful blocking action.
 - **Step 8:** LogBlock redirects the User back to the News Feed.
- Alternative Flow

First Alternative Flow - User Cancel Deletion Prompt

- **Condition:** the user does not accept the confirmation prompt at **Step 4**.
- **At step 4**, if the User does not accept the confirmation, the dialog box will disappear and the User is presented with the External Target Profile Page.

Second Alternative Flow - Alternative Prompt Acceptance method

- **Condition:** The User reaches **Step 3**.
- **At Step 4**, Aside from using the Accept Button, the User can also press Enter on the keyboard in order to accept the prompt.

Third Alternative Flow - The User is Connecting to the External Target Profile

- **Condition:** The User reaches **Step 6**, and the User is already Connecting to the External Target Profile
- **At Step 6**, the flow proceed as follows:
- **Step 6.1**, LogBlock finds the entries in which either the User id matches the Connector field and the External Target Profile id matches the ConnectedTo field, and/or the User id matches the ConnectedTo field and the External Target Profile id matches the Connector field.
- **Step 6.2**, LogBlock removes the entries found from the previous **step 6.1**.

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

- The flow continues from **Step 7** of the **Basic Flow**.

Fourth Alternative Flow - Unexpected already-exists in BlockedProfiles attribute

- **Condition:** The User reaches **Step 6**, and the External User Profile id already exists in the BlockedProfiles attribute value of the User's entry in the Profile Table.
- Upon detection of already-existing at **Step 6**, LogBlock will discard the remaining steps and proceed as follows:
- **Step 6**, LogBlock informs the User of an unexpected error of having displayed the Blocking button for a non-displayable Profile for the User, because the User has already blocked the External Target Profile.
- **Step 7:** LogBlock redirects the User back to the News Feed.

10.3 Preconditions

- The User must be logged in to the platform.
- The User profile must exist on the platform.
- The External Target Profile id must be of a valid User Profile that exists on the platform.
- The External Target Profile id must not match with the User's Profile id.
- The External Target Profile id must not already exist in the User's entry BlockedProfiles attribute value.

10.4 Postconditions

- Upon success, the External Target Profile id is added to the User's entry BlockedProfiles attribute value.
- Upon success, the User is redirected to the News Feed, and any contents related to the External Target User won't be presented to the User.

10.5 Extension Points

- **Unblocking Mechanism.** The Service might consider an unblocking mechanism in the near future.

11. UC11: Connect or Disconnect to Target Profile

11.1 Brief Description

This Use Case details how a Registered User and User classes inherited from Registered User can connect to another User Profile (labeled External Target Profile) to receive occasional updates of the External Target Profile's new posting via the News Feed Page.

11.2 Flow of Events

- Basic Flow
 - **Step 1:** The User navigates to the External Target Profile Page.
 - **Step 2:** The User clicks on the Connect button on the External Target Profile Card.
 - **Step 3:** LogBlock saves the User's id and the External Target Profile id.
 - **Step 4:** LogBlock inserts an entry into the Connection Table, with Connector field filled with the User's id, ConnectedTo field filled with the External Target Profile id; and the ConnectionDate field is initialized to the date of the action.
 - **Step 5:** LogBlock informs the User that the User has successfully connected to the External Target Profile.
 - **Step 6:** LogBlock refreshes the External Target Profile Page, revealing newly changed connection counts of the External Target Profile.
- Alternative Flows

First Alternative Flow - Unconnect to External Target Profile

- **Condition:** the User must have already connected to the External Target Profile.
- **At step 2**, the User clicks on the **Unconnect** button rather than the Connect button on the External Target Profile Card.

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

- **At step 3**, LogBlock saves the User's id and the External target Profile's id.
- **At step 4**, LogBlock removes the entry in which the User id matches the Connector field and the External Target Profile id matches the ConnectedTo field in the Connection Table.
- **At step 5**, LogBlock informs the User that the User has successfully unconnected from the External Target Profile.
- **Step 6**: LogBlock refreshes the External Target Profile Page, revealing newly changed connection counts of the External Target Profile.

Second Alternative Flow - Unexpected entry in Connection Table

- **Condition:** the User uses the Connect button but there is already a Connection entry indicating the connection between the User and the External Target Profile in the Connection Table.
- **At step 4**, LogBlock will tries to insert a new entry into the Connection Table, if an already-existed error occurs, LogBlock will proceed as follows:
- **Step 5**, LogBlock informs the User that the User has already connected to the External Target Profile and that this is an unexpected condition to happen.
- **Step 6**, LogBlock refreshes the External Target Profile Page in order to display correct information.

11.3 Preconditions

- The User must be logged in to the platform.
- The User profile must exist on the platform.
- The External Target Profile id must be of a valid User Profile that exists on the platform.
- The External Target Profile id must not match with the User's Profile id.
- The User must not already be connected to the External Target Profile in order to proceed with the Connect flow.
- The Connection Table must be able to insert a new entry into.
- The user must be connected to the External Target Profile in order to proceed with the Unconnect flow.

11.4 Postconditions

- The User will be informed that the connecting process is successful,
- The User will have an entry in the Connection Data indicating the connection between the User and the External Target Profile.
- The User's *Connected to* count will be updated to include the connection to the External Target User.
- The External Target Profile *Connected by* count will be updated to include the connection from the User.

12. UC12: News Feed Post Generation Request

12.1 Brief Description

This Use Case illustrates the process in which Posts are retrieved automatically when the Registered User and User classes inherited from Registered User scrolls down the News Feed, which are personalized based on the User's connection.

12.2 Flow of Events

- Basic Flow
 - **Step 1:** The User enters the News Feed Page.
 - **Step 2:** The User scrolls to the bottom of the News Feed Page, at first load, the News Feed is empty, so the User is already at the bottom of the News Feed.
 - **Step 3:** The User's browser makes a News Feed Post Generation Request, including the User's id, and a list of already retrieved Post ids in request (the list is stored locally on the User's browser local storage).
 - **Step 4:** LogBlock receives the request and saves the User's id and the already retrieved Post ids.
 - **Step 5:** LogBlock finds all entries in the Connection Table where the Connector field matches the User's id.

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

- **Step 6:** LogBlock retrieves the ConnectedTo field from the Connection entries results from the previous **step 5**.
- **Step 7:** For each ConnectedTo id, LogBlock finds the top 10 recent Posts ordered by PostCreation date that are originally authored by the ConnectedTo field and are not in the already retrieved Post ids from the Posting Table.
- **Step 8:** LogBlock retrieves the Posts Data by following the **UC08: Post Data Retrieval Use Case**, providing the Use Case with the lists of Post ids taken from **Step 7**.
- **Step 9:** LogBlock displays the posts to the User according to the sorted order:
- **Step 10:** LogBlock displays the Post Card, containing at the header: the OriginalAuthor's Profile Image, Display Name and the Post Creation date. The Post Card header also includes a More button, allowing Users to click on to open a small dialog with Report option.
- **Step 11:** LogBlock generates the Post body, containing: the Post Caption, rendered through a Markdown engine, attached media below the Post Caption
- **Step 12:** LogBlock generates the Post footer, containing: the Comment button, showing the *comment count* and can be clicked on to redirect to the Post Comment section. Alongside the Comment button is the number of upvotes of the Post by counting the number of elements in the PostUpvote list and the Upvote or Remove Upvote button.
- **Step 13:** If the User reaches the bottom of the News Feed Page, the process starts again from **Step 2**.

■ Alternative Flow

First Alternative Flow - No Post ids are retrieved

- **Condition:** The Basic Flow cannot retrieve any Post data for displays
- **At step 7**, if no Post data is available, the flow continues as follow:
- **Step 8:** LogBlock displays a "No Post available, maybe try connecting with other Users?" message to prompt User to search for External Connections.
- **Step 9:** LogBlock disables the automatic News Feed Post Generation request at the end of User's scroll.

Second Alternative Flow - Original Author additional items:

- **Condition:** The Post's OriginalAuthor id matches with the User's id.
- **At step 10**, the More button at the Post Card header also allows the User to edit or delete the Post.

12.3 Preconditions

- The User must be logged in to the platform.
- The User profile must exist on the platform.
- The User should have at least one Connection to External Users for meaningful Post Data Retrieval.
- The preconditions of Use Cases that are included in this Use Case must all be satisfied.

12.4 Postconditions

- The User will be displayed with newest posts from Profiles which the User connected to.
- The User will be able to keep scrolling down to view older posts, sorted by descending PostCreation date.

13. UC13: Exploration Feed Post Generation Request

13.1 Brief Description

This Use Case illustrates the process in which Posts are retrieved automatically when the User scrolls down the Exploration Feed, which are identical for all Users.

13.2 Flow of Events

■ Basic Flow

- **Step 1:** The User enters the Exploration Feed Page.
- **Step 2:** The User scrolls to the bottom of the Exploration Feed Page, at first load, the Exploration Feed is

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

empty, so the User is already at the bottom of the Exploration Feed.

- **Step 3:** The User's browser makes a Exploration Feed Post Generation Request, including the User's id, and a list of already retrieved Post ids in request (the list is stored locally on the User's browser local storage).
- **Step 4:** LogBlock receives the request and saves the User's id and the already retrieved Post ids.
- **Step 5:** LogBlock finds the Most Recent Posts, sorted by descending order of PostCreation attribute value from the Posting Table and whose id are not present in the already retrieved Post ids list.
- **Step 6:** LogBlock retrieves the Posts Data by following the **UC08: Post Data Retrieval Use Case**, providing the Use Case with the lists of Post ids taken from **Step 5**.
- **Step 7:** LogBlock displays the posts to the User according to the sorted order:
- **Step 8:** LogBlock displays the Post Card, containing at the header: the OriginalAuthor's Profile Image, Display Name and the Post Creation date
- **Step 9:** LogBlock generates the Post body, containing: the Post Caption, rendered through a Markdown engine, attached media below the Post Caption. The Post Card header also includes a More button, allowing Users to click on to open a small dialog with Report option.
- **Step 10:** LogBlock generates the Post footer, containing: the Comment button, showing the *comment count* and can be clicked on to redirect to the Post Comment section. Alongside the Comment button is the number of upvotes of the Post by counting the number of elements in the PostUpvote list and the Upvote or Remove Upvote button.
- **Step 11:** If the User reaches the bottom of the Exploration Feed Page, the process starts again from **Step 2**.

■ Alternative Flow

First Alternative Flow - No Post ids are retrieved

- **Condition:** The Basic Flow cannot retrieve any Post data for displays
- **At step 5**, if no Post data is available, the flow continues as follow:
- **Step 8:** LogBlock displays a "No Post available, maybe try posting something" message to prompt User to populate LogBlock with contents.
- **Step 9:** LogBlock disables the automatic Exploration Feed Post Generation request at the end of User's scroll.

Second Alternative Flow - Original Author additional items:

- **Condition:** The Post's OriginalAuthor id matches with the User's id.
- **At step 8**, the More button at the Post Card header also allows the User to edit or delete the Post.

13.3 Preconditions

- The User must be logged in to the platform.
- The User profile must exist on the platform.
- The preconditions of Use Cases that are included in this Use Case must all be satisfied.
- There should be at least one Post entry in the Posting Database.

13.4 Postconditions

- The User will be displayed with globally newest posts from other Users.
- The User will be able to keep scrolling down to view older posts, sorted by descending PostCreation date.

14. UC14: User Delete Original Posting

14.1 Brief Description

This use case illustrates the process of how a Registered User and User classes inherited from Registered User can delete an existing Post that the User originally posted and the updates LogBlock has to make in order to accompany the changes .

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

14.2 Flow of Events

- Basic Flow
 - **Step 1:** The User navigates to the Targeted Post's Card Page.
 - **Step 2:** The User clicks on the More icon on the Post.
 - **Step 3:** LogBlock displays a small dialog selection box, including the Edit Post option and the Delete Post option.
 - **Step 4:** The User clicks on the Delete Post button.
 - **Step 5:** LogBlock prompts the user to confirm the deletion of the post.
 - **Step 6:** The User accepts the prompt by clicking on the Accept button.
 - **Step 7:** LogBlock saves the Targeted Post's id.
 - **Step 8:** LogBlock retrieves the User Profile entry that matches the User's id from the Profile Table.
 - **Step 9:** LogBlock looks for the Targeted Post's id in the retrieved User Profile entry's PinnedPosts field, if present, LogBlock will remove it from the attribute value.
 - **Step 10:** LogBlock deletes entries from the Comment Table that have PostID fields match with the Targeted Post's id.
 - **Step 11:** LogBlock deletes entry from the Posting Table that has PostID field matches with the Targeted Post's id.
 - **Step 12:** LogBlock notifies the User of the successful deletion of the Post.
 - **Step 13:** LogBlock redirects the User back to the News Feed.

■ Alternative Flows

First Alternative Flow - User Cancel Post Deletion Prompt

- **Condition:** the user does not accept the confirmation prompt at **Step 6**.
- **At Step 6**, if the User does not accept the confirmation, the dialog box will disappear and the User is presented with the Post Card Page.

Second Alternative Flow - Alternative Prompt Acceptance method

- **Condition:** The User reaches **Step 5**.
- **At Step 5**, Aside from using the Accept Button, the User can also press Enter on the keyboard in order to accept the prompt.

14.3 Preconditions

- The User must be logged in to the platform.
- The User profile must exist on the platform.
- The User must be the Original Author of the Targeted Post to delete.
- The Deleting Post entry that matches the requested Post id to be deleted must exist in the Posting Table.

14.4 Postconditions

- After deletion, the User is prompted with a success dialog box.
- Entry of the Targeted Post in the Posting Table will be deleted.
- Comments which references Post entry id that matches the Targeted Post's id will also be deleted.
- The Post will not be retrieved for display any further.

15. UC15: User Create Posting

15.1 Brief Description

This Use Case demonstrates the steps that are necessary for a Registered User and User classes inherited from Registered User to create and upload a Post.

15.2 Flow of Events

- Basic Flow

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

- **Step 1:** User navigates to the News Feed Page.
- **Step 2:** User clicks on the Create Post button.
- **Step 3:** LogBlock displays a Post Creating Dialog box. The dialog box contains a form with:
 - A wide text field for User to record textual data for the post. Textual data will be recorded in Markdown format and denoted as *Text Data*.
 - A Media button for the User to insert media into the Post. Media files are saved temporarily, denoted *Media Data*, on LogBlock memory until Post creation finalization or Post creation cancellation.
 - A Tag button for User to tag relevant User for co-authoring or mentioning.
- **Step 4:** The User enters textual data into the *Text Data* field.
- **Step 5.1:** The User uploads relevant media into the *Media Data* field by using the Media button, which opens a dialog for the User to choose the uploading media file.
- **Step 5.2:** For each Media upload, LogBlock processes the media, considering the format of the uploaded file and the size.
- **Step 6.1:** LogBlock obtains a list of Connectors that connects to the User from the Connection Table.
- **Step 6.2:** The User uses the Tag button, which promptly opens up a mini-dialog, allowing the User to search for External Users whose id matches in the Connector field in a Connection entry in the Connection Table where the User's id matches the ConnectedTo field.
- **Step 6.3:** At each character insertion into the search field, LogBlock filters out the result from **Step 6.1** based on the User's current search field value, and displays the Profile Image and Display Name of the best-match search results.
- **Step 6.4:** The User chooses search results which best match what the User searches.
- **Step 6.5:** LogBlock retrieves the External User's id and saves to a list, denoted as *Tag Data*.
- **Step 7:** After finalizing all changes, the User clicks the Post button to initiate the Posting process.
- **Step 8:** LogBlock saves all User inputs along with the User's id.
- **Step 9.1:** LogBlock goes through User's uploaded *Media Data*, from **Step 5.2**, all uploaded Media should be of compatible file format and size.
- **Step 9.2:** LogBlock uploads each media to the centralized bucket storage of LogBlock and saves the URLs for the uploaded Medias.
- **Step 10:** LogBlock creates a Post entry in the Posting Table, filling out the following fields:
 - PostID: generated by incrementing the PostID of the latest Post in the Posting Table.
 - OriginalAuthor: filled with the User's id.
 - PostCaption: filled with *Text Data*.
 - PostMedia: filled with URLs from **Step 9.2**.
 - PostCreation: filled with the current date.
 - PostLastUpdate: filled with the current date.
 - PostTagging: filled with the *Tag Data* retrieved from **Step 6.5**.
 - PostUpvote: initialized with an empty list.
- **Step 11:** LogBlock informs the User of the successful Post Creation through a dialog.
- **Step 12:** LogBlock closes the Dialog Box, and refreshes the News Feed Page.

First Alternative Flow - Alternative Post form submission.

- **Condition:** the User reaches **Step 7**.
- **At Step 7**, if the User does not use the Post button, the User can also use the Enter key on the keyboard to submit the form.

Second Alternative Flow - Invalid posting

- **Condition:** Both *Text Data* and *Media Data* are empty.
- **At Step 4 to Step 7**, while both the *Text Data* and *Media Data* are empty, the Post button will be greyed out, along with the ability to submit a Post form.

Third Alternative Flow - Invalid Input Text

- **Condition:** The User enters a non-supported Unicode character

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

- **At Step 4 or 6.3**, if the User entered a non UTF-8 character, the border stroke of the text field will turn red and an error message will appear, stating that the User has entered a non-supporting character and prompts the User to change it. The Posting submission action will be blocked.

Fourth Alternative Flow - Invalid Picture Format

- **Condition: at Step 5.2**, the User's uploaded file is not the kind of picture that LogBlock can process (PNG, bitmap, JPEG).
- The dialog box's border stroke will turn red. LogBlock will display an error message, stating that the User has uploaded a non-supported picture format and prompts the User to change it. The Posting submission action will be blocked.

Fifth Alternative Flow - Invalid Picture Size

- **Condition: at Step 5.2**, the User's uploaded file is too large. (maximum size is 512x512 pixels)
- The dialog box's border stroke will turn red. LogBlock will display an error message, stating that the User has uploaded a picture whose size is too large and prompts the User to change it. The Posting submission action will be blocked.

Sixth Alternative Flow - Alternative Media Uploading by URL

- **Condition: at Step 5.1**, the User is also presented with a text field, prompting the User to enter the URL of a media file. From here, the steps are followed:
- **Step 5.1.1:** The User clicks on the Retrieve button next to the URL field.
- **Step 5.1.2:** LogBlock receives the picture URL and tries to get the media through that URL.
- **Step 5.1.3:** If the picture cannot be retrieved, the dialog box's border stroke will turn red, LogBlock will display an error message, stating that the User has entered a non-retrievable URL and prompts the User to change it or try other means.
- **Step 5.1.4:** If the picture is successfully retrieved, the process proceeds from **Step 5.2** as mentioned in the **Basic Flow**.

Seventh Alternative Flow - Discard Editing:

- **Condition:** the User quits the Posting process either by closing the Dialog box by clicking outside of it or by clicking the Close button.
- At any moment from **step 3 to step 7**, the User can choose to discard all uploaded contents by clicking on the Close button, closing the current browser window, or clicking outside of the Dialog Box form.
- The User is then returned to the News Feed Page.

Eighth Alternative Flow - No Media Upload:

- **Condition:** the User does not upload any media files in **Step 5**
- **Step 9** will be skipped.
- **At Step 10**, PostMedia will be initialized with an empty list.

Ninth Alternative Flow - No Tagging

- **Condition:** the User does not tag any other Profile in **Step 6**.
- **Step 6** will be skipped.
- **At Step 10**, PostTagging will be initialized with an empty list.

15.3 Preconditions

- The User must be logged in to the platform.
- The User profile must exist on the platform.
- The Back-End Service bucket storage must have enough storage to store the newly uploaded media files.
- The Posting Table must be able to insert a new entry into.

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

15.4 Postconditions

- The User is informed of the success of the Posting process.
- The User's News Feed Page is refreshed to include the User's Post at the top before aggregating other News Feed Post.
- External Users who are tagged by the User during **Step 6** will get a one-time notification. The notification will be sent as a pop-up if the External User is already logged in the current session, or it will be queued for pop-up for the External User's next logged in session.

16. UC16: User Edit Posting

16.1 Brief Description

This Use Case demonstrates the steps that are necessary for a Registered User and User classes inherited from Registered User to edit an existing Post..

16.2 Flow of Events

- Basic Flow
 - **Step 1:** User navigates to the Targeted Post Card.
 - **Step 2:** User clicks on the More button on the Targeted Post Card.
 - **Step 3:** LogBlock displays a small dialog selection box, including the Edit Post option and the Delete Post option.
 - **Step 4:** The User clicks on the Edit Post button.
 - **Step 5:** LogBlock gets the Post's id. LogBlock finds the Post entry that matches the Post's id in the Posting Table.
 - **Step 6:** LogBlock finds all User entries that match the User id presented in the Post entry's PostTagging field, and saves the Display Name of each User entry.
 - **Step 7:** LogBlock displays a Post Editing Dialog box. The dialog box contains a form with:
 - A wide text field for User to record textual data for the post. The field is pre-filled with PostCaption value from the Post entry retrieved at **Step 5**. Textual data will be recorded in Markdown format and denoted as *Text Data*.
 - A Media button for the User to insert media into the Post. The field is pre-filled with PostMedia value from the Post entry retrieved at **Step 5**, LogBlock displays a thumbnail of each Media. Media files are saved temporarily, denoted *Media Data*, on LogBlock memory until Post creation finalization or Post creation cancellation.
 - A Tag button for User to tag relevant User for co-authoring or mentioning, the field is pre-filled with a list of User ids from **Step 5**. Display Names retrieved from **Step 6** are displayed rather than the User ids for better User Experience.
 - **Step 8:** The User makes changes to the textual data into the *Text Data* field.
 - **Step 9:** The User uploads or removes relevant media into/from the *Media Data* field by using the Media button, which opens a dialog for the User to choose the uploading media file.
 - **Step 10:** After finalizing all changes, the User clicks the Save button to initiate the Editing process.
 - **Step 11:** LogBlock saves all User inputs along with the User's id.
 - **Step 12.1:** LogBlock goes through User's uploaded *Media Data*, all uploaded Media should be of compatible file format and size.
 - **Step 12.2** LogBlock cross-check with the original URLs in the PostMedia field of the Target Post entry and deletes media stored on the bucket storage that are not present in the new PostMedia value.
 - **Step 12.3:** LogBlock uploads new media to the centralized bucket storage of LogBlock and saves the URLs for the uploaded Medias.
 - **Step 13:** LogBlock updates the Post entry in the Posting Table that matches the Targeted Post's id, filling out the following fields:
 - PostCaption: filled with updated *Text Data*.
 - PostMedia: filled with updated URLs from **Step 12.3**.

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

- PostLastUpdate: filled with the current date.

- **Step 14:** LogBlock informs the User of the successful Post Editing through a dialog.
- **Step 15:** LogBlock closes the Dialog Box, and refreshes the Page the User is currently in.

First Alternative Flow - Alternative Post editing submission.

- **Condition:** the User reaches **Step 10**.
- **At Step 10**, if the User does not use the Save button, the User can also use the Enter key on the keyboard to submit the form.

Second Alternative Flow - Invalid editing

- **Condition:** Both *Text Data* and *Media Data* are empty.
- **At Step 8 to Step 10**, while both the *Text Data* and *Media Data* is empty, the Save button will be greyed out, along with the ability to submit Editing form.

Third Alternative Flow - Invalid Input Text

- **Condition:** The User enters a non-supported Unicode character
- **At Step 8**, if the User entered a non UTF-8 character, the border stroke of the text field will turn red and an error message will appear, stating that the User has entered a non-supporting character and prompts the User to change it. The Editing submission action will be blocked.

Fourth Alternative Flow - Invalid Picture Format

- **Condition: at Step 12.1**, the User's uploaded file is not the kind of picture that LogBlock can process (PNG, bitmap, JPEG).
- The dialog box's border stroke will turn red. LogBlock will display an error message, stating that the User has uploaded a non-supported picture format and prompts the User to change it. The Editing submission action will be blocked.

Fifth Alternative Flow - Invalid Picture Size

- **Condition: at Step 12.1**, the User's uploaded file is too large. (maximum size is 512x512 pixels)
- The dialog box's border stroke will turn red. LogBlock will display an error message, stating that the User has uploaded a picture whose size is too large and prompts the User to change it. The Editing submission action will be blocked.

Sixth Alternative Flow - Alternative Media Uploading by URL

- **Condition:** at **Step 9**, the User is also presented with a text field, prompting the User to enter the URL of a media file. From here, the steps are followed:
- **Step 9.1:** The User clicks on the Retrieve button next to the URL field.
- **Step 9.2:** LogBlock receives the picture URL and tries to get the media through that URL.
- **Step 9.3:** If the picture cannot be retrieved, the dialog box's border stroke will turn red, LogBlock will display an error message, stating that the User has entered a non-retrievable URL and prompts the User to change it or try other means.
- **Step 9.4:** If the picture is successfully retrieved, the process proceeds from **Step 10** as mentioned in the **Basic Flow**.

Seventh Alternative Flow - Discard Editing:

- **Condition:** the User quits the Editing process either by closing the Dialog box by clicking outside of it or by clicking the Close button.
- At any moment from **step 7 to step 10**, the User can choose to discard all uploaded contents by clicking on the Close button, closing the current browser window, or clicking outside of the Dialog Box form.
- The User is then returned to the News Feed Page.

Eight Alternative Flow - Accepting Expert Suggested Solutions:

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

- **At Step 7**, the Editing Dialog also includes a Solutions button. clicking on it will proceed the follow steps:
- **At Step 7.1**, LogBlock retrieves Solution entries from the Expert Solutions Table that has the PostID field that matches the Post's id.
- **At Step 7.2**, LogBlock retrieves the DisplayName from the User entries that match the SolAuthor from the Solution entries obtained from **Step 10.1**.
- **At Step 7.3**, LogBlock displays small cards that display the Suggested Solution Author Display Name that User can click on to open up the Solution Card, which has the same User Interface as a Post.
- **At Step 7.4**, the User chooses to accept a Solution by clicking on the Accept button at the end of a Solution Card.
- **At Step 7.5**, LogBlock saves the *Text Data* with the value from the Solution's SolCaption, *Media Data* with the value from the Solution's SolMedia.
- **At Step 7.6**, LogBlock removes all entries from the Expert Solution Table which has the PostID field that matches the Editing Post's id.
- The next step follows **Step 11** from the **Basic Flow**.

16.3 Preconditions

- The User must be logged in to the platform.
- The User profile must exist on the platform.
- The Back-End Service bucket storage must have enough storage to store any newly uploaded media files.
- The Targeted Post id must match one Post entry in the Posting Table.
- The Targeted Post OriginalAuthor field must match the User's id.

16.4 Postconditions

- The User is informed of the success of the Editing process.
- The Post entry of the Targeted Post will be updated according to the newly updated changes.
- The User's News Feed Page is refreshed.

17. UC17: Post Upvoting

17.1 Brief Description

This Use Case describes how a Registered User and User classes inherited from Registered User can interact with a Post by Upvoting or by removing an existing Upvote from the Post.

17.2 Flow of Events

■ Basic Flow

- **Step 1:** The User navigates to the Targeted Post Card.
- **Step 2:** The User clicks on the Upvote button at the footer of the Post Card.
- **Step 3:** LogBlock finds the Post entry that matches the Targeted Post's id in the Posting Table.
- **Step 4:** LogBlock checks if the User's id is already present in the PostUpvote attribute value belonging to the Post entry retrieved from **Step 3**.
- **Step 5:** If not present, LogBlock adds the User's id to the list.
- **Step 6:** LogBlock informs the User of the successful upvoting action.
- **Step 7:** LogBlock updates the Targeted Post Card according to the newly added upvote cast.

■ Alternative Flows

First Alternative Flow - User already upvoted and is Removing Upvote

- **Condition:** The User's id is already present in the Post entry's PostUpvote list at **Step 4**.
- **At Step 2**, the User clicks on the Remove Upvote button.
- **At Step 3:** LogBlock finds the Post entry that matches the Targeted Post's id in the Posting Table.
- **At Step 4:** LogBlock checks if the User's id is already present in the PostUpvote attribute value belonging to the Post entry retrieved from **Step 3**.
- **At Step 5**, the check returns present, LogBlock will remove the User's id from the PostUpvote list.

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

- **At Step 6:** LogBlock informs the User of the successful removing upvote action.
- **At Step 7:** LogBlock updates the Targeted Post Card according to the newly removed upvote cast.

17.3 Preconditions

- The User must be logged in to the platform.
- The User profile must exist on the platform.
- The Targeted Post must exist as a Post entry in the Posting Table.
- The User must not be blocked by the Post Original Author.
- The User's upvote must present only once or none in a Post's PostUpvote list.

17.4 Postconditions

- Upon success, The User is informed of the successful upvoting/un-upvoting action.
- The PostUpvote list of the Targeted Post is updated accordingly.
- The User's view of the Targeted Post Card is updated according to the new changes.

18. UC18: User Commenting

18.1 Brief Description

This Use Case demonstrates the steps that are necessary for a Registered User and User classes inherited from Registered User to comment to an existing Post.

18.2 Flow of Events

■ Basic Flow

- **Step 1:** User navigates to the Targeted Post Card
- **Step 2:** User clicks on the Comment button of the Targeted Post Card footer.
- **Step 3:** LogBlock displays a Comment Creating Dialog box. The dialog box contains a form with:
 - A wide text field for User to record textual data for the comment. Textual data will be recorded in Markdown format and denoted as *Text Data*.
- **Step 4:** The User enters textual data into the *Text Data* field.
- **Step 5:** After finalizing all changes, the User clicks the Comment button to initiate the Commenting process.
- **Step 6:** LogBlock saves all User inputs along with the User's id.
- **Step 7:** LogBlock creates a Comment entry in the Posting Table, filling out the following fields:
 - PostID: filled with the Targeted Post's id.
 - CommentID: generated by incrementing the CommentID of the latest Comment entry belonging to this Post. If no other Comment exists, default to 0.
 - CommentAuthor: filled with the User's id.
 - CommentCaption: filled with *Text Data*.
 - CommentCreation: filled with the current date.
- **Step 8:** LogBlock informs the User of the successful Comment Creation through a dialog.
- **Step 9:** LogBlock closes the Dialog Box, and refreshes the News Feed Page.

First Alternative Flow - Alternative Comment form submission.

- **Condition:** the User reaches **Step 5**.
- **At Step 5,** if the User does not use the Comment button, the User can also use the Enter key on the keyboard to submit the commenting form.

Second Alternative Flow - Invalid posting

- **Condition:** Both *Text Data* is empty.
- **At Step 4,** while the *Text Data* is empty, the Comment button will be greyed out, along with the ability to submit commenting form.

Third Alternative Flow - Invalid Input Text

- **Condition:** The User enters a non-supported Unicode character
- **At Step 4,** if the User entered a non UTF-8 character, the border stroke of the text field will turn red and an

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

error message will appear, stating that the User has entered a non-supporting character and prompts the User to change it. The Posting submission action will be blocked.

Seventh Alternative Flow - Discard Commenting:

- **Condition:** the User quits the Commenting process either by closing the Dialog box by clicking outside of it or by clicking the Close button.
- At any moment from **step 3 to step 5**, the User can choose to discard all uploaded contents by clicking on the Close button, closing the current browser window, or clicking outside of the Dialog Box form.
- The User is then returned to the News Feed Page.

18.3 Preconditions

- The User must be logged in to the platform.
- The User profile must exist on the platform.
- The Targeted Post must exist as an entry in the Posting Table.
- The User must not be blocked by the Targeted Post Original Author.

18.4 Postconditions

- The User is informed of the success of the commenting process.
- The Targeted Post Card is refreshed to include the User's Comment.
- External Users who are tagged by the User during **Step 6** will get a one-time notification. The notification will be sent as a pop-up if the External User is already logged in the current session, or it will be queued for pop-up for the External User's next logged in session.

19. UC19: Profile Searching

19.1 Brief Description

This Use Case describes how a Registered User and User classes inherited from Registered User can use the Search functionality at the Exploration Page in order to find External User Profiles.

19.2 Flow of Events

- Basic Flow
 - **Step 1:** The User navigates to the Exploration Feed.
 - **Step 2:** The User focuses on the Search Bar input field.
 - **Step 3:** The User enters value into the Search Bar input field.
 - **Step 4:** At each input character, LogBlock finds all entries in the Profile Table that have DisplayName or Email best matches with the User's searching input through regex matching.
 - **Step 5:** If more than zero results are matched from **Step 4**, LogBlock displays a list of Profile Summary Cards that contains the Searching Profile Image and The Searching Profile Display Name, along with the Email Address associated with the Searching Profile.
 - **Step 6:** The User clicks on the matching External Targeted User Profile result that satisfies the User.
 - **Step 7:** LogBlock redirects the User to the External Targeted User Profile Page.
- Alternative Flows

First Alternative Flow - No Search Result

- **Condition:** No search result matches the User's searching input.
- **At Step 5**, if no results are found and the User's searching input length is greater than 0, LogBlock will display a small text above the Search Bar, implying that the User's current input would not match any other profiles.

Second Alternative Flow - Invalid Input Text

- **Condition:** The User enters a non-supported Unicode character

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

- **At Step 4**, if the User entered a non UTF-8 character, the border stroke of the text field will turn red and an error message will appear, stating that the User has entered a non-supporting character and prompts the User to change it. The Posting submission action will be blocked.

19.3 Preconditions

- The User must be logged in to the platform.
- The User profile must exist on the platform.
- The User must access the Search Bar from the Exploration Feed.
- The Profile Table must have at least two different User Profiles.

19.4 Postconditions

- Upon successful search matching, LogBlock will display a list of Profile Summary Cards that best match the User's input.
- Upon User clicking a Profile Summary Card, the User will be redirected to the Profile Page associated with that Profile Summary Card.

20. UC20: Post Report

20.1 Brief Description

This Use Case describes how a Registered User and User classes inherited from Registered User can use the Report functionality to alert the Administration Team of hazardous contents that go against LogBlock's usage guidelines.

20.2 Flow of Events

■ Basic Flow

- **Step 1:** The User navigates to Targeted Post Card.
- **Step 2:** The User clicks on the More button on the header of the Targeted Post Card.
- **Step 3:** LogBlock displays a small dialog selection box, including only the Report option.
- **Step 4:** The User clicks on the Report option.
- **Step 5:** LogBlock prompts the User for Reporting Confirmation.
- **Step 6:** The User confirms the Report form submission.
- **Step 7:** LogBlock records a new entry in the Reporting table, filling the following fields:
 - ReportID: generated by taking the ReportID of the latest Report entry and incrementing it by one.
 - Reporter: filled with the User's id.
 - ReportPostID: filled with the Targeted Post's id.
 - ReportDate: filled with the current date.
- **Step 8:** LogBlock informs the User of the successful report.
- **Step 9:** LogBlock returns the User back to the News Feed Page.

First Alternative Flow - The User declines Reporting Confirmation

- **Condition:** The User does not accept the Reporting Confirmation.
- **At Step 5**, if the User chooses to decline the confirmation, the Report dialog will be destroyed and User returns to the Targeted Post Card.

20.3 Preconditions

- The User must be logged in to the platform.
- The User profile must exist on the platform.
- The User must not be the Original Author of the Targeted Post.
- The User must have not reported the Targeted Post before.

20.4 Postconditions

- Upon successful search matching, LogBlock will display a list of Profile Summary Cards that best match the User's input.
- Upon User clicking a Profile Summary Card, the User will be redirected to the Profile Page associated with that Profile Summary Card.

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

21. UC21: Expert Suggested Solution Creation

21.1 Brief Description

This Use Case demonstrates the steps that are necessary for an Expert User class to propose a Solution to an existing Post.

21.2 Flow of Events

■ Basic Flow

- **Step 1:** User navigates to the Targeted Post Card.
- **Step 2:** User clicks on the More button on the Targeted Post Card.
- **Step 3:** LogBlock displays a small dialog selection box, including the Report option and the Propose option.
- **Step 4:** The User clicks on the Propose button.
- **Step 5:** LogBlock gets the Post's id. LogBlock finds the Post entry that matches the Post's id in the Posting Table.
- **Step 6:** LogBlock displays a Solution Suggestion Dialog box. The dialog box contains a form with:
 - A wide text field for User to record textual data for the solution. Textual data will be recorded in Markdown format and denoted as *Text Data*.
 - A Media button for the User to insert media into the Post. Media files are saved temporarily, denoted as *Media Data*, on LogBlock memory until Solution creation finalization or Solution creation cancellation.
- **Step 7:** The User makes changes to the textual data into the *Text Data* field.
- **Step 8:** The User uploads media into the *Media Data* field by using the Media button, which opens a dialog for the User to choose the uploading media file.
- **Step 9:** After finalizing all changes, the User clicks the Submit button to initiate the Editing process.
- **Step 10:** LogBlock saves all User inputs along with the User's id.
- **Step 11.1:** LogBlock goes through User's uploaded *Media Data*, all uploaded Media should be of compatible file format and size.
- **Step 11.2:** LogBlock uploads new media to the centralized bucket storage of LogBlock and saves the URLs for the uploaded Medias.
- **Step 12:** LogBlock adds a Solution entry into the Expert Solutions Table, filling out the following fields:
 - PostID: filled with the Targeted Post's id.
 - SolID: generated by taking the SolID of the latest solution proposal belonging to the Targeted Post. If none Solution presents, default to 0.
 - SolAuthor: filled with the User's id.
 - SolCaption: filled with updated *Text Data*.
 - SolMedia: filled with updated URLs from **Step 11.2**.
 - SolCreation: filled with the current date.
 - SolLastUpdate: filled with the current date.
- **Step 13:** LogBlock informs the User of the successful Post Editing through a dialog.
- **Step 14:** LogBlock closes the Dialog Box, and refreshes the Page the User is currently in.

First Alternative Flow - Alternative Solution submission.

- **Condition:** the User reaches **Step 9**.
- **At Step 9**, if the User does not use the Submit button, the User can also use the Enter key on the keyboard to submit the Solution form.

Second Alternative Flow - Invalid editing

- **Condition:** Both *Text Data* and *Media Data* are empty.
- **At Step 7 to Step 9**, while both the *Text Data* and *Media Data* are empty, the Submit button will be greyed out, along with the ability to submit Solution form.

Third Alternative Flow - Invalid Input Text

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

- **Condition:** The User enters a non-supported Unicode character
- **At Step 7**, if the User entered a non UTF-8 character, the border stroke of the text field will turn red and an error message will appear, stating that the User has entered a non-supporting character and prompts the User to change it. The Editing submission action will be blocked.

Fourth Alternative Flow - Invalid Picture Format

- **Condition: at Step 11.1**, the User's uploaded file is not the kind of picture that LogBlock can process (PNG, bitmap, JPEG).
- The dialog box's border stroke will turn red. LogBlock will display an error message, stating that the User has uploaded a non-supported picture format and prompts the User to change it. The Editing submission action will be blocked.

Fifth Alternative Flow - Invalid Picture Size

- **Condition: at Step 12.1**, the User's uploaded file is too large. (maximum size is 512x512 pixels)
- The dialog box's border stroke will turn red. LogBlock will display an error message, stating that the User has uploaded a picture whose size is too large and prompts the User to change it. The Editing submission action will be blocked.

Sixth Alternative Flow - Alternative Media Uploading by URL

- **Condition:** at **Step 8**, the User is also presented with a text field, prompting the User to enter the URL of a media file. From here, the steps are followed:
- **Step 8.1:** The User clicks on the Retrieve button next to the URL field.
- **Step 8.2:** LogBlock receives the picture URL and tries to get the media through that URL.
- **Step 8.3:** If the picture cannot be retrieved, the dialog box's border stroke will turn red, LogBlock will display an error message, stating that the User has entered a non-retrievable URL and prompts the User to change it or try other means.
- **Step 8.4:** If the picture is successfully retrieved, the process proceeds from **Step 9** as mentioned in the **Basic Flow**.

Seventh Alternative Flow - Discard Editing:

- **Condition:** the User quits the Solution submission process either by closing the Dialog box by clicking outside of it or by clicking the Close button.
- At any moment from **step 7 to step 9**, the User can choose to discard all uploaded contents by clicking on the Close button, closing the current browser window, or clicking outside of the Dialog Box form.
- The User is then returned to the News Feed Page.

21.3 Preconditions

- The User must be logged in to the platform.
- The User profile must exist on the platform.
- The User must have privilege level 1 (Expert level).
- The Back-End Service bucket storage must have enough storage to store any newly uploaded media files.
- The Targeted Post id must match one Post entry in the Posting Table.
- The Targeted Post OriginalAuthor field must not match the User's id.

21.4 Postconditions

- The User is informed of the success of the Solution submission process.
- The Solution can be viewed by the Original Author by navigating to the More button during the Post Editing Process.

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

22. UC22: Dedicated Page to Monitor Reported Objects

22.1 Brief Description

This Use Case describes how an Administration User can use the Dedicated Report Logging Page to monitor Reports sent by other Users.

22.2 Flow of Events

■ Basic Flow

- **Step 1:** The User navigates to Dedicated Report Logging Page.
- **Step 2:** LogBlock retrieves Report entries from the Reporting Table.
- **Step 3:** LogBlock displays Report entries, identified through their unique ReportID.
- **Step 4:** The User clicks on a Report entry.
- **Step 5:** LogBlock shows the content of the Targeted Post Card, along with the time at which the Report is filed and the Display Name, User id of the User who sent that Report.
- **Step 6:** User chooses to delete the Post, which follows **UC23: Remove Post**.

First Alternative Flow - The User discards the Report

- **Condition:** The User does not takes deletion action against the Post and prompt to discard the report
- **At Step 6,** the User can chooses to discard the report by clicking the Discard button.
- **Step 6.1:** LogBlock removes the Report entry from the Reporting Table
- **Step 6.2:** LogBlock informs the User that the Report has been discarded.
- **Step 6.3:** LogBlock reloads the Dedicated Report Logging Page to accompany new changes.

22.3 Preconditions

- The User must be logged in to the platform.
- The User profile must exist on the platform.
- The User must have privilege level 2 (Administration level).

22.4 Postconditions

- Upon successful the Targeted Post entry will either be deleted or remain based on the User's choice.
- The Report entry that was processed will be removed from the Reporting Table.

23. UC23: Remove Post

23.1 Brief Description

This Use Case describes how an Administration User can delete a Post that violates LogBlock core usage guidelines.

23.2 Flow of Events

■ Basic Flow

- **Step 1:** The User navigates to the Targeted Post Card.
- **Step 2:** The User clicks on the More button on the Targeted Post Card header section.
- **Step 3:** LogBlock displays a small dialog selection box, including the Report option and the Delete option.
- **Step 4:** The User clicks on the Delete option
- **Step 5:** LogBlock prompts the User for Post Deletion Confirmation
- **Step 6:** The User accepts the prompt by clicking the Accept button.
- **Step 7:** LogBlock finds the Original Author of the Post, and removes the Targeted Post's id from the Author's User entry PinnedPosts attribute if present in the Profile Table.
- **Step 8:** LogBlock removes all Comment entries from the Commenting Table that is associated with the Post.
- **Step 9:** LogBlocks removes the Targeted Post entry from the Posting Table.
- **Step 10:** LogBlock informs the User that the Targeted Post has been deleted.
- **Step 11:** LogBlock redirects the User back to the News Feed.

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

First Alternative Flow - User Cancel Post Deletion Prompt

- **Condition:** the user does not accept the confirmation prompt at **Step 6**.
- **At Step 6**, if the User does not accept the confirmation, the dialog box will disappear and the User is returned to the current page..

Second Alternative Flow - Alternative Prompt Acceptance method

- **Condition:** The User reaches **Step 6**.
- **At Step 6**, Aside from using the Accept Button, the User can also press Enter on the keyboard in order to accept the prompt.

Third Alternative Flow - User Delete Post from the Dedicated Report Logging Page.

- **Condition:** the User reaches **Step 6** from **UC22: Dedicated Page to Monitor Reported Objects**.
- **At Step 6**, if the User chooses to remove the Targeted Report Post, the step will continue as follow:
- **Step 7:** LogBlock finds the Original Author of the Post, and removes the Targeted Post's id from the Author's User entry PinnedPosts attribute if present in the Profile Table.
- **Step 8:** LogBlock removes all Comment entries from the Commenting Table that is associated with the Post.
- **Step 9:** LogBlocks removes the Targeted Post entry from the Posting Table.
- **Step 10:** LogBlock informs the User that the Targeted Post has been deleted.
- **Step 11:** LogBlock redirects the User back to the Dedicated Report Logging Page.

23.3 Preconditions

- The User must be logged in to the platform.
- The User profile must exist on the platform.
- The User must have privilege level 2 (Administration level).

23.4 Postconditions

- Upon successful the Targeted Post entry will be deleted from the Posting Table.
- Any references to the Targeted Post entry will also be deleted.
- The User will be informed of the successful deletion.

24. UC24: Remove Account

24.1 Brief Description

This Use Case describes how an Administration User can delete a User Profile Account that violates the core guidelines of LogBlock.

24.2 Flow of Events

■ Basic Flow

- **Step 1:** The User navigates to the Targeted User Profile Page
- **Step 2:** The User clicks on the More button on the Targeted User Profile Card.
- **Step 3:** LogBlock displays a small dialog selection box, including only the Delete Profile option.
- **Step 4:** The User clicks on the Delete Profile option.
- **Step 5:** LogBlock prompts the User for User Profile Deletion Confirmation.
- **Step 6:** The User accepts the prompt by clicking on the Accept button.
- **Step 7:** LogBlock finds all Comments entries that are authored by the Targeted User Profile, and removes all entries that satisfy the criteria.
- **Step 8:** LogBlock finds all Connections that contain the Targeted User Profile's id at either the Connector field or the ConnectedTo field, and removes all entries that satisfy the criteria.
- **Step 9:** LogBlock finds all Post entries that have the Targeted User Profile's id in the ProfileTagged attribute, and removes the id from the attribute.

LogBlock Social Media Service	Version: 1.4
Use-Case Specification	Date: 27/11/2024
<document identifier>	

- **Step 10:** LogBlock finds all Solution entries that are authored by the Targeted User Profile from the Expert Solutions Table, and removes all entries that satisfy the criteria.
- **Step 11:** LogBlock removes the Targeted User Profile entry from the Profile Table.
- **Step 12:** LogBlock automatically invalidates all access tokens generated for the Targeted User Profile.
- **Step 13:** LogBlock informs the User of the successful User Profile Account Deletion.
- **Step 14:** LogBlock redirects the User back to the News Feed Page.

First Alternative Flow - User Cancel Account Deletion Prompt

- **Condition:** the user does not accept the confirmation prompt at **Step 6**.
- **At Step 6**, if the User does not accept the confirmation, the dialog box will disappear and the User is returned to the Targeted User Profile Page.

Second Alternative Flow - Alternative Prompt Acceptance method

- **Condition:** The User reaches **Step 6**.
- **At Step 6**, Aside from using the Accept Button, the User can also press Enter on the keyboard in order to accept the prompt.

24.3 Preconditions

- The User must be logged in to the platform.
- The User profile must exist on the platform.
- The User must have privilege level 2 (Administration level).
- The Targeted User Profile must be a valid entry in the Profile Table.

24.4 Postconditions

- Upon successful the Targeted User Profile entry will be deleted from the Profile Table.
- Any references to the Targeted User profile entry will also be deleted.
- The User will be informed of the successful deletion.
- The Targeted User will be logged out of the platform.
- The Targeted user will be informed of the account deletion decision from the Administration Team.