Use Case Category: Manage Forum

Use Case Name: Delete posts

Scope: Cool Poetry Assister

Level: User Goal

Primary Actor: Admin

Stakeholders and Interests:

Admin wants to delete inappropriate posts.

User wants a clean forum.

Preconditions: Admin is logged in.

Success Guarantee: Post is deleted from the system.

Main Success Scenario:

1. Admin browsing poetry.

2. Admin reads an inappropriate post.

3. Admin chooses to delete the post from the forum.

4. System returns confirmation message.

5. Admin confirms or denies.

6. Selected post is deleted from the forum.

Extensions:

*a At any time, Admin cancels

1. System returns to browsing.

5b At any time, Post is deleted by someone else.

1. System informs Admin that the post has already been deleted.

2. System returns to browsing.

Frequency of occurrence: Whenever posts that don’t align with our forum rules are posted.

Open issues:

Use Case Category: Manage Forum

Use Case Name: Ban user

Scope: Cool Poetry Assister

Level: User Goal

Primary Actor: Admin

Stakeholders and Interests:

Admin wants to ban users whose behavior is against the code of conduct.

Preconditions: Admin is logged in.

Success Guarantee: User is banned from the system.

Main Success Scenario:

1. Admin browsing poetry.

2. Admin reads an inappropriate post or comment

3. Admin chooses to ban the user from the system.

4. System returns confirmation message.

5. Admin confirms or denies.

6. Selected user is banned from the forum.

Extensions:

*a At any time, Admin cancels

1. System returns to browsing.

5b At any time, User has already been banned by someone else.

1. System informs Admin that the user has already been banned.

2. System returns to browsing.

Frequency of occurrence: Whenever users post posts or comments that don’t align with our forum rules.

Open issues:

Use Case Category: Manage Forum

Use Case Name: Lock post

Scope: Cool Poetry Assister

Level: User Goal

Primary Actor: Admin

Stakeholders and Interests:

Admin wants to lock posts to keep a civil tone.

User wants a clean forum.

Preconditions: Admin is logged in.

Success Guarantee: Post is locked.

Main Success Scenario:

1. Admin browsing poetry.

2. Admin reads a comment section that is too heated.

3. Admin chooses to lock the post.

4. System returns confirmation message.

5. Admin confirms or denies.

6. Selected post is locked.

Extensions:

*a At any time, Admin cancels

1. System returns to browsing.

5b At any time, Post has been locked by someone else.

1. System informs Admin that the post has already been locked.

2. System returns to browsing.

Frequency of occurrence: Whenever the discussion is heated.

Open issues:

Use Case Category: Manage Forum

Use Case Name: Delete User Comments

Scope: Cool Poetry Assister

Level: User Goal

Primary Actor: Admin

Stakeholders and Interests:

Admin wants to delete user comments.

User wants a clean forum.

Preconditions: Admin is logged in.

Success Guarantee: Post is deleted from the system.

Main Success Scenario:

1. Admin browsing poetry.

2. Admin reads an inappropriate comment.

3. Admin chooses to delete the comment from the post.

4. System returns confirmation message.

5. Admin confirms or denies.

6. Selected comment is deleted from the post.

Extensions:

*a At any time, Admin cancels

1. System returns to browsing.

5b At any time, comment has been deleted by someone else.

1. System informs Admin that the comment has already been deleted.

2. System returns to browsing.

Frequency of occurrence: Whenever comment that don’t align with our code of conduct.

Open issues:

Use Case Category: Rate Poetry

Use Case Name: Comment Post

Scope: Cool Poetry Assister

Level: User Goal

Primary Actor: User

Stakeholders and Interests:

User want the ability to give constructive feedback on poetry posted on the forum.

Preconditions: User is logged in.

Success Guarantee: The comment is stored in the system.

Main Success Scenario:

1. User is browsing poetry.

2. User choose a post to engage with.

3. User choose to leave a comment on the post.

4. User write his/her feedback.

5. 'Post comment' is selected.

6. System stores comment.

Extensions:

*a At any time, User cancels

1. System returns to viewing the post.

5b At any time, Comment is too long.

1. System informs user that the comment is too long.

2. System returns to writing a comment.

5c At any time, post has been locked

1. System informs the user that the post has been locked.

2. System returns to viewing the post.

5d Comment is empty

1. System informs user that the comment is empty

2. System returns to writing a comment.

Frequency of occurrence: Whenever users wants to give constructive feedback.

Open issues:

Use Case Category Manage post

Use Case Name: Create Post

Scope: Cool Poetry Assister

Level: User Goal

Primary Actor: User

Stakeholders and Interests:

User wants to create a post and make it public on the forum.

Preconditions: User is logged in.

Success Guarantee: The poetry is stored in the system.

Main Success Scenario:

1. User is browsing poetry.

2. User selects ‘create post’.

3. User writes his/her poetry.

4. User selects ‘submit post’

5. The post is submitted.

6. The post is pushed to the live feed.

Extensions:

4b. Post is empty

1. The post is empty

2. The post is not pushed to the live feed.

Frequency of occurrence: Whenever a user wants to create a post.

Open issues:

..

Browse poetry - Use Case

Use case name: Browse poetry
Scope: Cool poetry assister application
Level: user goal
Primary Actor: User
Stakeholders and Interests:

- User: Wants to flip through poetry threads on the main feed so he can see the titles and then open the posts which interests him.
- User: Wants to flip through its own poetry threads so that he can have an overview over his activity on the forum.
- User: Wants to filter the posts by specific criteria, like poetry genre or author, so that he can browse based on his preferences.

Preconditions: User is logged in.
Success Guarantee:Posts are filtered. Poetry threads are displayed on the main feed. A user's posts are displayed.
Main Succes Scenario:

1. User opens the live feed.
2. Poetry posts are displayed.

Extensions:
*a. At any time the user can chose to filter the live feed:

1. The user selects a criteria from a criteria list.
2. The live feed displays only the posts that have the selected criteria.

*b. At any time the user can chose to see its own posts.

1. The user requests the action.
2. The live feed is updated to only display the user's posts.

*c. At any time the user can choose to see the poetry posted by the users in its friends list.

1. The user requests the action.
2. The live feed is updated to only display the poetry posted by friends.

2a. If there are no posts to be displayed either because there are no posts at all or because there are no posts with the specific criteria then a message is displayed.

1. The system displays a message informing the user that there are no posts.

Frequency of Occurrence: Could be nearly continous.
Open Issues:

- What criteria should we add to the criteria list?
- How should the posts be displayed? Inside a list or as blocks building on top of each other?

Use case category: Manage post.

Use case name: Remove comments
Scope: Cool poetry assister application
Level: user goal
Primary Actor: User
Stakeholders and Interests:

- User: Wants to remove his own comment.

Preconditions:

- The user must have a post opened.
- The user must have a comment on that post.

Success Guarantee: The comment is removed.
Main Succes Scenario:

1. The user selects the coment that he wishes to remove.
2. The user requests the removal of the comment.
3. The user confirms the removal of the comment.
4. The comment is removed from the post

Extensions:
*a. At any time the user can abort the operation:

1. Use case ends.

4a. If the comment is succesufully removed a confirmation message is shown.

1. A confirmation message is displayed.

4b. If the comment could not be removed (database error), an error message is displayed.

1. An error message is displayed.

Frequency of Occurrence: Once per comment.
Open Issues:

- Do we need to inform the other users as well, that the comment has been deleted?

Use case category: Manage post.

Use case name: Edit post
Scope: Cool poetry assister application
Level: user goal
Primary Actor: User
Stakeholders and Interests:

- User: After posting something on the live feed, the user wants to edit the content of that post.

Preconditions:

- The user must have something posted into the live feed.
- The user must have the post opened.

Success Guarantee: The content of the poetry post is altered.
Main Succes Scenario:

1. The user selects one of his posts.
2. The user requests to edit the post.
3. The user edits the post.
4. The user confirms the changes.
5. System records the change.

Extensions:
*a. At any time, the user can abort the operation:

1. Use case ends.

4a. If the user wishes to abort the operation, then he shall return to editing the post.

1. The user requests to cancel the submission.
2. The user is returned to editing the post.

5a. If the system successfully recorded the operation than the user shall be notified with confirmation message.

1. The system displays that the operation has been successful.

5b. If the system fails to record the change than the user shall be notified by an error message.

1. The system displays an error message.

Frequency of Occurrence: Once per post.
Open Issues:

- Do we need to inform the other users as well that the post has been deleted?

Use case category: Manage post.

Use case name: Remove post.
Scope: Cool poetry assister application.
Level: user goal
Primary Actor: User
Stakeholders and Interests:

- User: After posting something into the live feed, the user of that post wants to remove it.
- Admin: If the admin finds a post inappropriate, he wants to remove it so that the feed stays relevant.

Preconditions:

- Both User and Admin need to have the feed opened.

Success Guarantee: The post is removed.
Main Succes Scenario:

1. The user selects one of his posts.
2. The user requests the post to be deleted.
3. The user confirms the removal.
4. The system records the changes.

Extensions:
*a. At any time the user can abort the operation:

1. Use case ends.

1a. If the Admin is logged in, than he can choose to remove any post from the live feed.

1. Admin can choose to remove any post.

3a. If the user cancels the removal than he is returned back to the live feed.

1. The user is returned back to the live feed.

4a. If the system succesufully recorded the operation than the user shall be notified with confirmation message.

1. The system displays that the operation has been successufull.

4b. If the system fails to record the change than the user shall be notified by an error message.

1. The system displays an error message.

Frequency of Occurrence: Can be performed as long as there are posts displayed on the feed.
Open Issues:

- Do we need to inform the other users that a post has been removed?

Use Case Category: Manage Account

Use Case Name: Create Account

Scope: Cool Poetry Assister

Level: User Goal

Primary Actor: User

Stakeholders and Interests:

User wants to create an account.

Preconditions: Software is installed.

Success Guarantee: Account is created and stores in the system.

Main Success Scenario:

1. Launch program.

2. System returns login window.

3. User selects ‘create account’.

4. System returns create account window.

5. User fills in username, password, full name, date of birth, email and a picture.

6. Selects create account

7. The system stores information

Extensions:

*a At any time, User cancels

1. System returns to the starting GUI.

6b Username is taken.

1. System informs the users that the username is currently not available.

2. Returns to create account.

6c Missing information.

1. System informs the user of the missing information.

2. Returns to create account.

6d Passwords doesn’t match.

1. System informs the user that the passwords don’t match.

2. Returns to create account.

Frequency of occurrence: Whenever someone wants to create a new account

Open issues:

Use Case Category: Manage Account

Use Case Name: Delete account

Scope: Cool Poetry Assister

Level: User Goal

Primary Actor: User

Stakeholders and Interests:

User wants to delete his/her account

Preconditions: Software is installed.

Success Guarantee: The specified account is deleted from the system

Main Success Scenario:

1. Launch program.
2. Login with account information.
3. Go to ‘manage account’.
4. Choose to delete account.
5. The system asks for confirmation.
6. The user chooses yes.
7. The account is deleted from the system

Extensions:

*a At any time, User cancels the confirmation message

1. System returns to ‘manage account’.

Frequency of occurrence: Whenever someone wants to delete his/her account.

Open issues:

Use Case Category: Manage Account

Use Case Name: Change username

Scope: Cool Poetry Assister

Level: User Goal

Primary Actor: User

Stakeholders and Interests:

User wants to change its name

Preconditions: Software is installed.

Success Guarantee: Username changes are stored in the system

Main Success Scenario:

1. Launch program.

2. Login with account information.

3. Go to ‘manage account’.

4. Choose to change username.

5. The system asks for a new username.

6. The user fills in the new username.

7. The user chooses to proceed.

8. The system asks for a password before changing.

9. The user fills in the required password.

10. The user chooses to proceed.

11. The account is deleted from the system

Extensions:

*a At any time, User cancels

1. System returns to the starting GUI.

*b Username is taken.

1. System informs the users that the username is currently not available.

2. Returns to create account.

*c Wrong password

1. System informs the user that the password is wrong.

2. The system asks for the password again.

Frequency of occurrence: Whenever someone wants to change username

Use Case Category: Manage Account

Use Case Name: Change password

Scope: Cool Poetry Assister

Level: User Goal

Primary Actor: User

Stakeholders and Interests:

User wants to change its password

Preconditions: Software is installed.

Success Guarantee: Password changes are stored in the system

Main Success Scenario:

1. Launch program.

2. Login with account information.

3. Go to ‘manage account’.

4. Choose to change password.

5. The system asks for a password before proceeding.

6. The user fills in the required password.

7. The system asks for a new password.

8. The user fills in the new password.

9. The user chooses to proceed.

10. Password changes are stored in the system.

Extensions:

*a At any time, User cancels

1. System returns to the starting GUI.

*b Password is the same as the current password.

1. System informs the users that the new password is the same as the current password.

2. The system asks for a new password.

*c Wrong password when proceeding

1. System informs the user that the password is wrong.

2. The system asks for the password again.

Frequency of occurrence: Whenever someone wants to change username

Use Case Category: Manage Account

Use Case Name: Login

Scope: Cool Poetry Assister

Level: User Goal

Primary Actor: User

Stakeholders and Interests:

User wants to login

Preconditions: Software is installed, and an account has been created.

Success Guarantee: The user is logged in so it can view the live feed

Main Success Scenario:

1. Launch program.

2. Login window is displayed

3. Enter password and username

4. Choose to login.

5. System checks the information with the database.

6. User is logged in and directed to a new window.

Extensions:

*a At any time, User cancels

1. System returns to the starting GUI.

4b Password and username are incorrect.

1. System informs the user that the username or password is incorrect.

2. returns the user to the login GUI.

Frequency of occurrence: Whenever a user wants to login.

Use Case Category: Manage Account

Use Case Name: My page

Scope: Cool Poetry Assister

Level: User Goal

Primary Actor: User

Stakeholders and Interests:

User wants to see information about himself/herself

Preconditions: Software is installed, and an account has been created.

Success Guarantee: The MyPage is shown displaying information about the user

Main Success Scenario:

1. Launch program.

2. Login window is displayed

3. Enter password and username

4. Choose to login.

5. System checks the information with the database.

6. User is logged in and directed to the livefeed.

7. User selects MyPage.

8. System opens MyPage displaying all information about the users, such as username, full name, date of birth, e-mail.

Extensions:

*a At any time, User cancels

1. System returns to the starting GUI.

4b Password and username are incorrect.

1. System informs the user that the username or password is incorrect.

2. returns the user to the login GUI.

Frequency of occurrence: Whenever a user wants to login.