

Software Project

Iteration 1

Bear Feed

Mushfika Rahman
Sadia Nasrin Tisha
Muhammad Ashfakur Arju
Tingshuo Miao
Noorah Ashmeel

23rd September 2022

1 Project Vision

The vision of this project is to create Baylor University news feed (Bear Feed), news feed portal for all Baylor University students and faculties.

Functional Requirements

- Users can log in to the system
- Users can register to the system
- User can ask for reset password
- Users can change information on their profile
- Users can create posts in the text format
- Users can add media to the post while creating the post
- Users have to add tags to the post
- Users can get notifications
- Users can search by tags
- Users can write comments
- Users can delete comments
- Users can add posts add their favorite
- Users can view other posts
- Users can filter news feeds by tags

- Users can create new tags
- Users can share on social media
- Users can like other posts in the feed
- Users can create events
- The system can fetch information from the external source

Non-Functional Requirements:

- Authentication to user login
- Password encryption
- Auto-suggestion
- Message queue

2 WireFrame

Wireframe

Login Page:

BEAR Feed 

Login

UserName

Password

Forgot Password:

Email:

Enter your OTP

.....

Sign UP

Sign UP

email:

username:

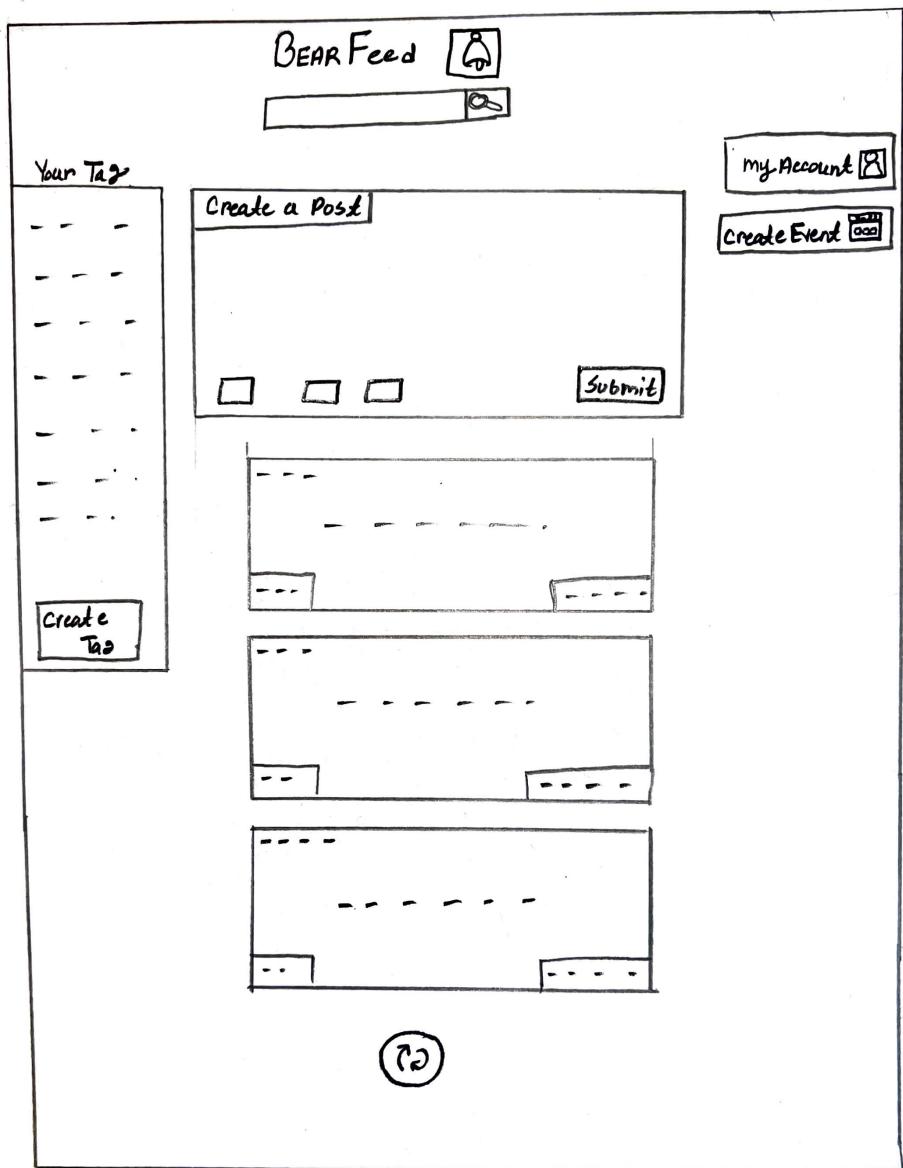
password:

Confirm password:

What you like:

Login

Main Page:



Create Post:

Create a Post

Text: Write something...

Tag: Add tag..

Add media

Submit

at click
image box pop-up

3 Fully-dressed Use Case

3.1 Use Case-1: Create Article

Use Case: Create Article	
ID: BN001	
Actors: User	
Preconditions: The user is logged in and clicks on the create article button	
Flow of events:	
1. User types/pastes his post in the text box.	
2. If the article does not exceed the word limit (1500 characters)	
3. User types for available tags.	
4. The system auto suggests tags.	
5. The user selects tag from suggestion.	
6. The system attaches a tag from the article.	
7. If the user uploads a media less than 2MB in image box.	
7.1 The system attaches media to the article.	
8. User clicks on create button.	
Postconditions: The system will display confirmation message	
Alternative flow:	
The user cancels the post without submitting.	
Post Conditions: The system saves it as a draft and redirects to home page.	
Alternative flow: 2.1 User writes an article exceeds word limit.	
Post Conditions: The system will display a warning message and make create button invisible.	
Alternative flow:	
The user uploads media greater than 2MB.	
Post Conditions: The system will display a warning message.	
Alternative flow:	
3.1 User writes tag not available in the system.	
3.2 User clicks on create tag icon.	
Post Conditions: The system attaches a new tag to the article.	

3.2 Use Case-2: Create Event

Use Case: Create Event
ID: BN0002
Actors: User
Preconditions: The user is logged in and clicks on the create an event button
Flow of events:
<ol style="list-style-type: none"> 1. User types/pastes title to the post in the text box. 2. The user sets a specific date for the event. 3. The user writes description of the event(1500 characters) 7. If the user uploads a picture in image box. <ol style="list-style-type: none"> 7.1 if the user uploads media less than 2 MB. 7.1.1 The system attaches media to the event. 4. User clicks on create event button.
Postconditions: The system will display confirmation message
Alternative flow: The user cancels without submitting.
Post Conditions: Redirects to the home feed.
Alternative flow:
<ol style="list-style-type: none"> 2.1 User sets a date that is in the past.
Post Conditions: The system will display a warning message.
Alternative flow:
<ol style="list-style-type: none"> 3.1 User writes a description that has exceeded the word limit.
Post Conditions: The system will display a warning message.

3.3 Use Case-3: View Post

Use Case: View Article
ID: BN003
Actors: User
Preconditions: The user must be logged in
Flow of events:
<ol style="list-style-type: none"> 1. The user clicks on the view full article button. 2. The system redirects loading the article with the list of comments. 3. If the user writes a comment in a text box and clicks on the add comment button. <ol style="list-style-type: none"> 2.1 The system will display a confirmation message. 4. If the user clicks on the add to favorite icon. <ol style="list-style-type: none"> 3.1 The system will change the color of the icon. 3.2 The system will add to user's favourites list. 5. If the user clicks on the like button. <ol style="list-style-type: none"> 5.1 The system will change the color of the icon.
Postconditions:
Alternative flow: The user closes the article without doing anything.
Post Conditions: Redirects to the home feed.
Alternative flow: 2.1 the user clicks on the delete icon.
Post Conditions: The system will display delete confirmation text.

3.4 Use Case-4: Notification

Use Case: Notification
ID: BN004
Actors: User
Preconditions: User is logged in
Flow of events:
1. The user clicks on the icon.
2. The system redirects to the notification page.
3. The Notification page will show all the previews of the notifications.
4. If the user clicks one of the articles in the list .
4.1 Redirects the user to that view page that loads the corresponding article.
5. If the user clicks on the mark as read button.
5.1 System will make the read icon visible.
6. If the user clicks on delete notification icon.
6.1 System removes the notification from the list.
Postconditions: Change the appearance of the notification icon
Alternative flow: User does not click on any notification, clicks on the home button.
Post Conditions: The system redirects to home.

3.5 Use Case-5: Search By Tag

Use Case: Search By Tag
ID: BN005
Actors: User
Preconditions: User is logged in the system
Flow of events:
1. User writes the text from the tag search.
2. The system auto suggests tags.
3. User selects tag from suggestions.
4. User clicks on the search icon
Postconditions: User is redirected to the page loads preview of the articles.
Alternative flow:
4. User writes a tag that doesn't exist in system.
5. User selects the create tag option.
Post Conditions: The system will redirect to the create tag page.
Alternative flow: User might not click the search icon.
Post Conditions: The system will keep the user in the current page.

3.6 Use Case-6: User Authentication

Use Case: User Authentication	
ID: BN006	
Actors: User	
Preconditions: The user goes to the login page.	
Flow of events:	
1. The user enters his/her name and password.	
2. Press Login	
3. If the system validates credentials.	
4.1 The system redirects the user to the home feed.	
4. Else the credentials are not validated the system will show a warning message.	
Postconditions: The system shows the personalization icon.	
Alternative flow:	
The user interacts with the forgot password.	
Post Conditions: The system will redirect to the reset password page.	
Alternative flow:	
The user leaves the login page.	
Post Conditions:	

3.7 Use Case-7: Forgot Credential

Use Case: Forgot Credential	
ID: BN007	
Actors: User	
Preconditions: If the user clicks forgot password icon	
Flow of events:	
1. The user enters his/her email id.	
2. Press submit icon.	
3. If the email id exists with the system.	
3.1 The user will get his/her OTP in the email id.	
3.2 A text box will be shown to enter OTP.	
3.3 The user enters the OTP in the text box.	
3.4 If the OTP is correct.	
3.4.1 A new password prompt is shown.	
3.4.2 User types new password and re-enter new password.	
3.4.3 If both the password and reentered are matched show no warning and make the submit button visible.	
3.4.4 Else the system shows warning.	
3.5 Else	
The system shows warning the OTP is mismatched.	
3. Else	
The system shows warning email does not exist.	
Postconditions:	
Alternative flow:	
The user requests for login.	
Post Conditions: The system redirects to the login page.	

3.8 Use Case-8: Change Information on User Profile

Use Case: Change Information on User Profile
ID: BN008
Actors: User
Preconditions: The user is logged in.
Flow of events: <ol style="list-style-type: none">1. The user clicks on the personalization icon.2. If the user enters new details in the fields.3. If the user uploads a profile picture in image box.<ol style="list-style-type: none">3.1 The user uploads media less than 2 MB.3.2 The system attaches media to the profile.4. If the user includes his/her social media profile.
Postconditions: The system will display a confirmation message
Alternative flow: 2.1 User uploads media more than 2 MB.
Post Conditions: The system will display a warning message.

3.9 Use Case-9: Add Tags to the article

Use Case: Add Tags to the article
ID: BN009
Actors: User
Preconditions: User must be logged on and clicks the add tag field.
Flow of events: <ol style="list-style-type: none">1. User writes text for the tags.2. System auto-suggests the existing tags.3. The user selects a tag from the existing tags.4. The system attaches a tag to the article.
Postconditions: Tags are in the rounded box.
Alternative flow: The user creates a new tag not registered in the system.
Post Conditions: System redirects to a create new tag page

3.10 Use Case-10: Create Tags

Use Case: Create Tags
ID: BN010
Actors: User
Preconditions: User must be logged on and clicks the create tag button.
Flow of events:
1. User writes text for the tag.
2. System auto-suggests existing tags of the systems.
3. The user writes a tag, not in the system.
4. The user clicks on the submit button.
4.1 Shows pop-up success messages.
5. Else
Shows warnings and stays on the page.
Postconditions: The system adds the tag to the list of already created tags by the user.
Alternative flow: The user can leave the page at any time.
Post Conditions: The system redirects to the desired page.

3.11 Use Case-11: Add to favorites

Use Case: Add to favorites
ID: BN0011
Actors: User
Preconditions: User must be logged on and in the view article page.
Flow of events:
1. User clicks on the add to favorites icon.
Postconditions: The icon's color is changed.
Alternative flow: User can leave the view article page.
Post Conditions: System redirects to the desired view.

3.12 Use Case-12: Share to social media

Use Case: Share to social media
ID: BN012
Actors: User
Preconditions: User is logged in and also in the view page of a post.
Flow of events:
<ol style="list-style-type: none"> 1. User clicks the social media icons. 2. System generates the shareable post. 3. The system redirects users to the desired social media platform. 4. If the browser has a valid login session then the user can share it on their desired social media. 5. Else user logs in the desired social media and then share the article.
Postconditions: User has valid login session in their browser for the desired social media.
Alternative flow: User clicks the social media icon but chooses not to share the post in their social media.
Post Conditions:

3.13 Use Case-13: Create Profile

Use Case: Create Profile
ID: BN013
Actors: User
Preconditions: User has to click on the create profile button.
Flow of events:
<ol style="list-style-type: none"> 1. System loads the create profile page. 2. User enters email, password, re enters password, username. 3. User clicks on the sign up button. 4. If the credentials are validated <ol style="list-style-type: none"> 4.1 The system redirects the select interest page. 4.2 User selects multiple options from the interest tags. 4.3 User is redirected to the personalized home feed. 5. Else the credentials are not validated the system will show a warning message.
Postconditions: The system will load the username, profile icon on the home page.
Alternative flow: User can leave from the create profile.
Post Conditions: System redirects to the generic home feed.
Alternative flow: User can click on the already have a profile link.
Post Conditions: System redirects to the login page.

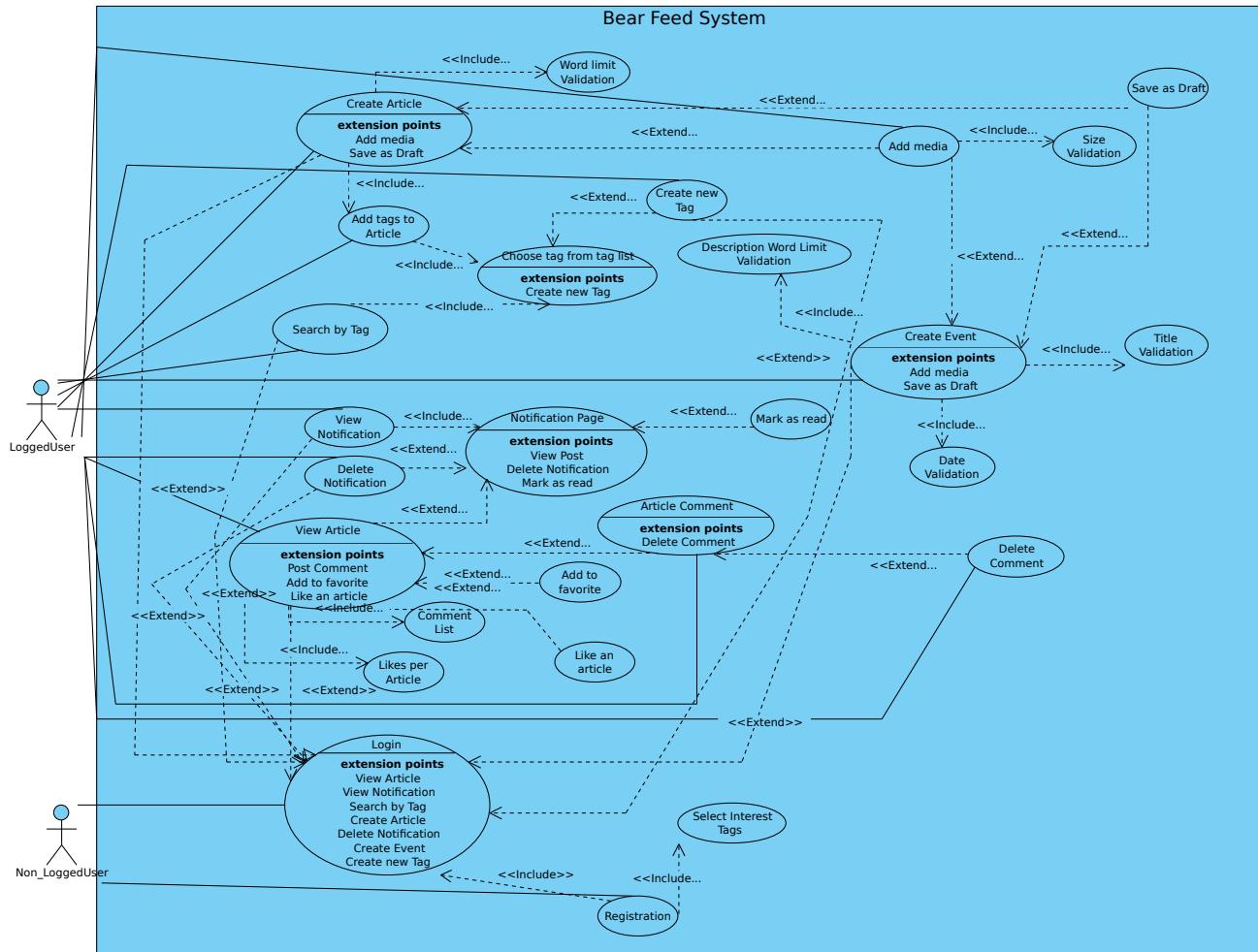
3.14 Use Case-14: Comment creation

Use Case: Comment creation
ID: BN014
Actors: User
Preconditions: User is logged in the system and in the view full article page.
Flow of events: 1. The user writes text to the comment. 2. The user clicks on the comment button. 3. The system pops up confirmation message.
Postconditions: The comment is attached to the post.
Alternative flow: The user chooses to delete the created comment.
Post Conditions: The system removes the comment from the list.

3.15 Use Case-15: Filter by Tags

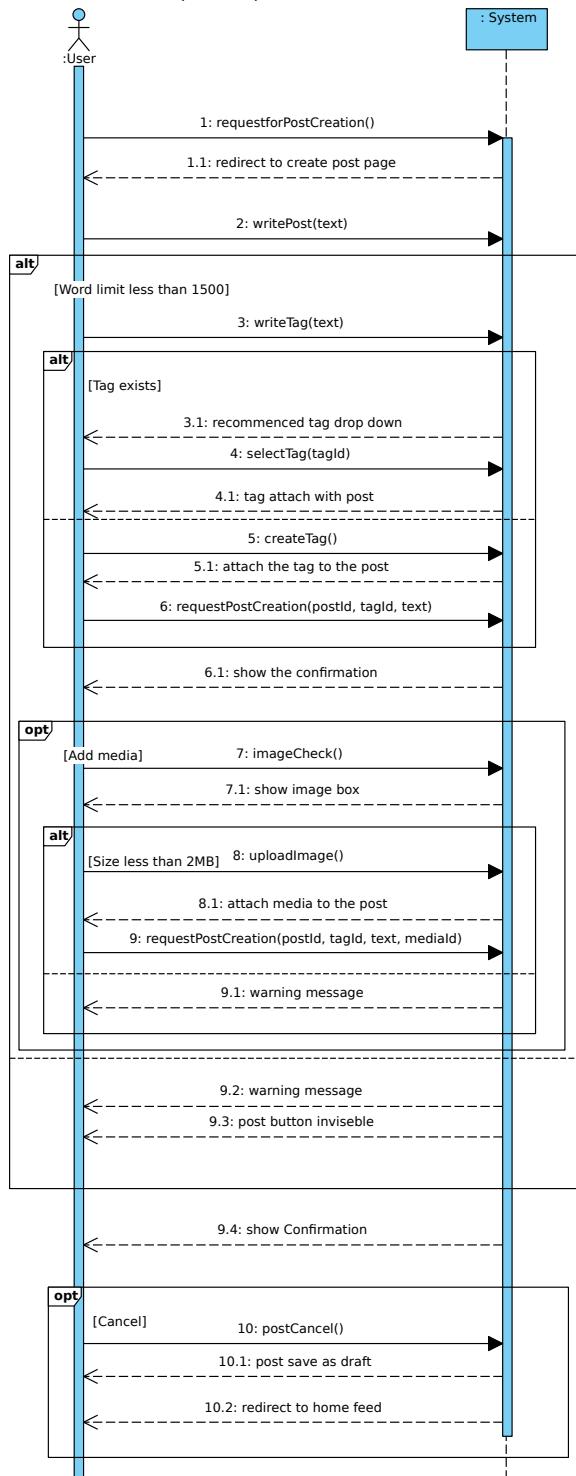
Use Case: Filter by Tags
ID: BN015
Actors: User
Preconditions: User has to be logged in and click on the filter icon
Flow of events: 1. User chooses tags. 2. Click on Save button. 3. System shows only related articles.
Postconditions:
Alternative flow:
Post Conditions:

4 Use Case Diagram

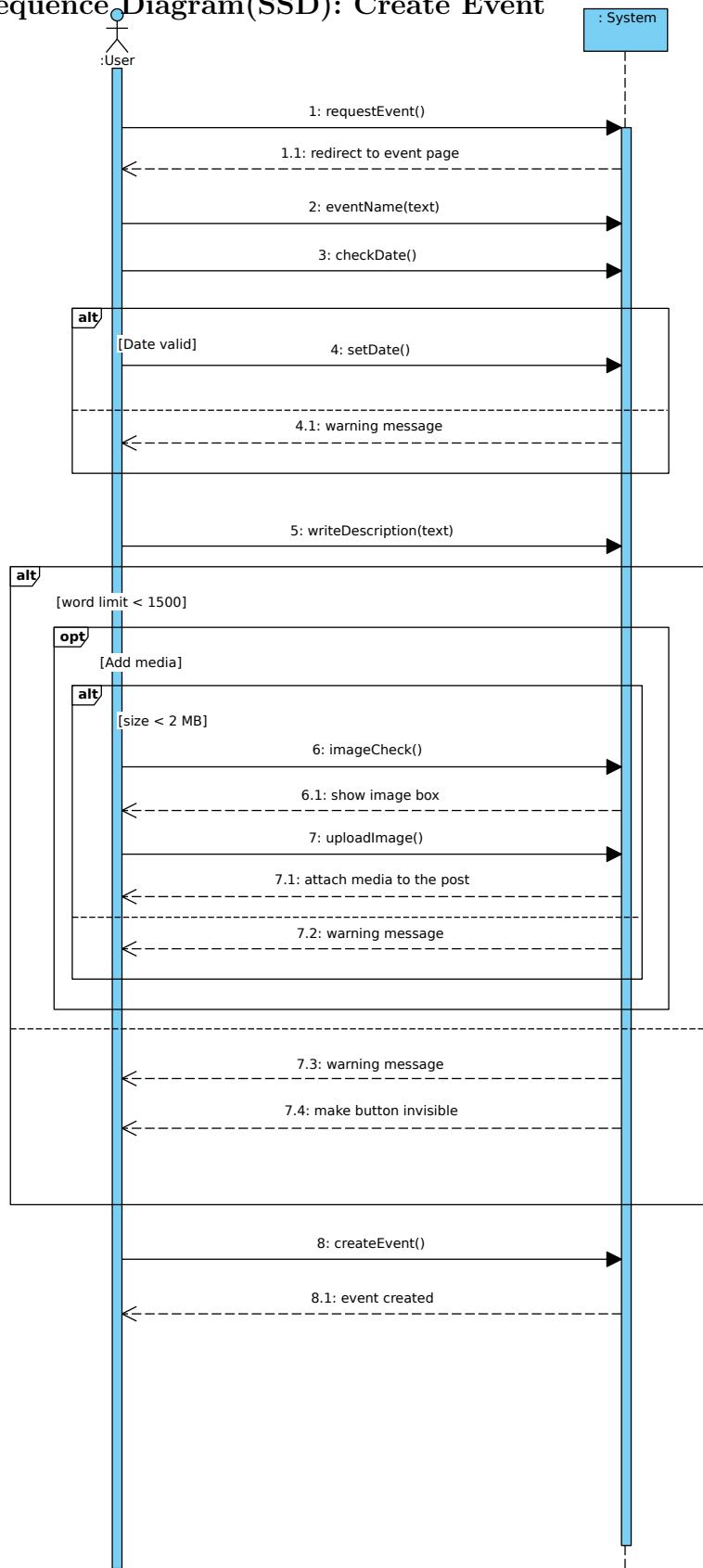


5 System Sequence Diagram(SSD):

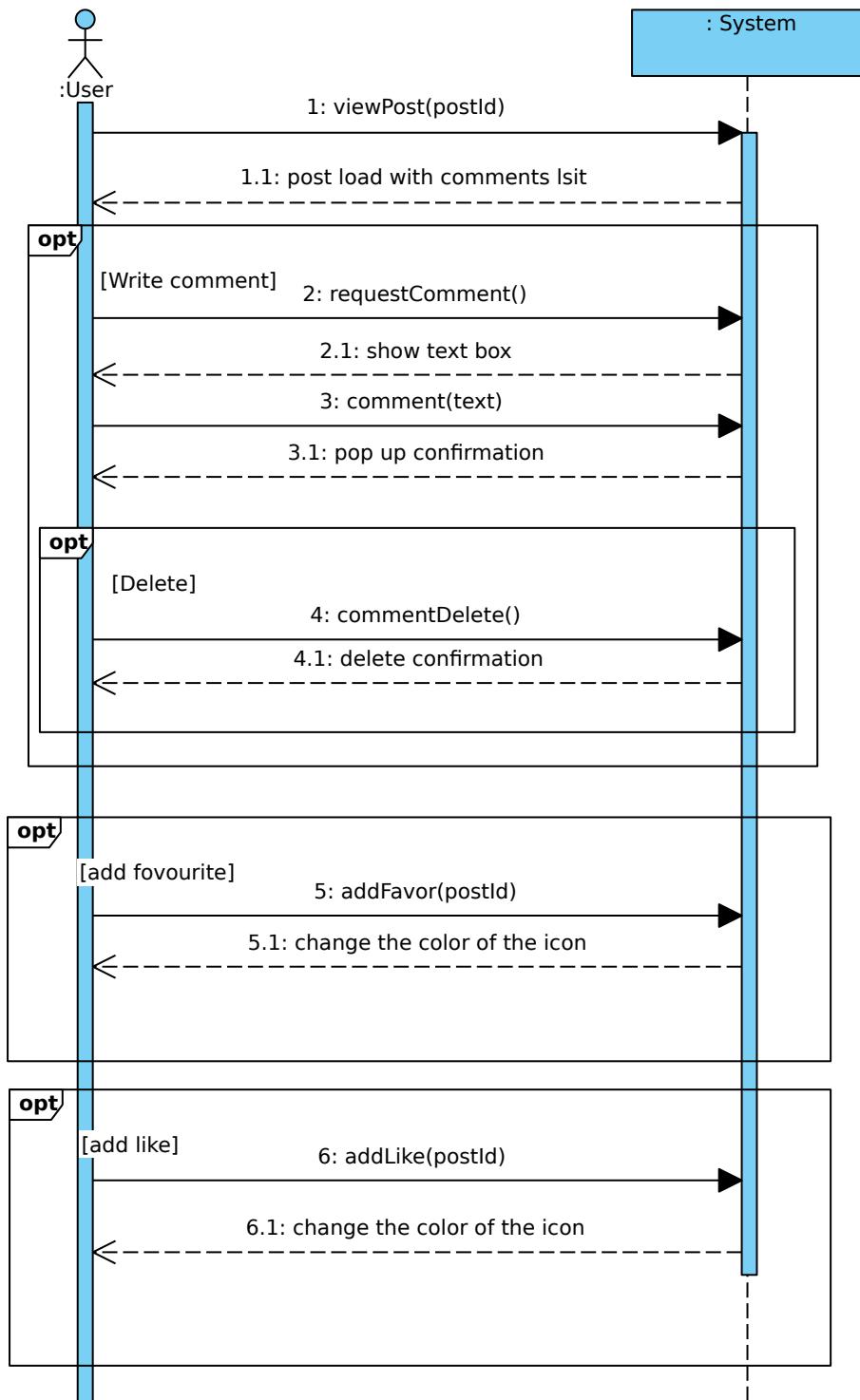
5.1 System Sequence Diagram(SSD): Create Post



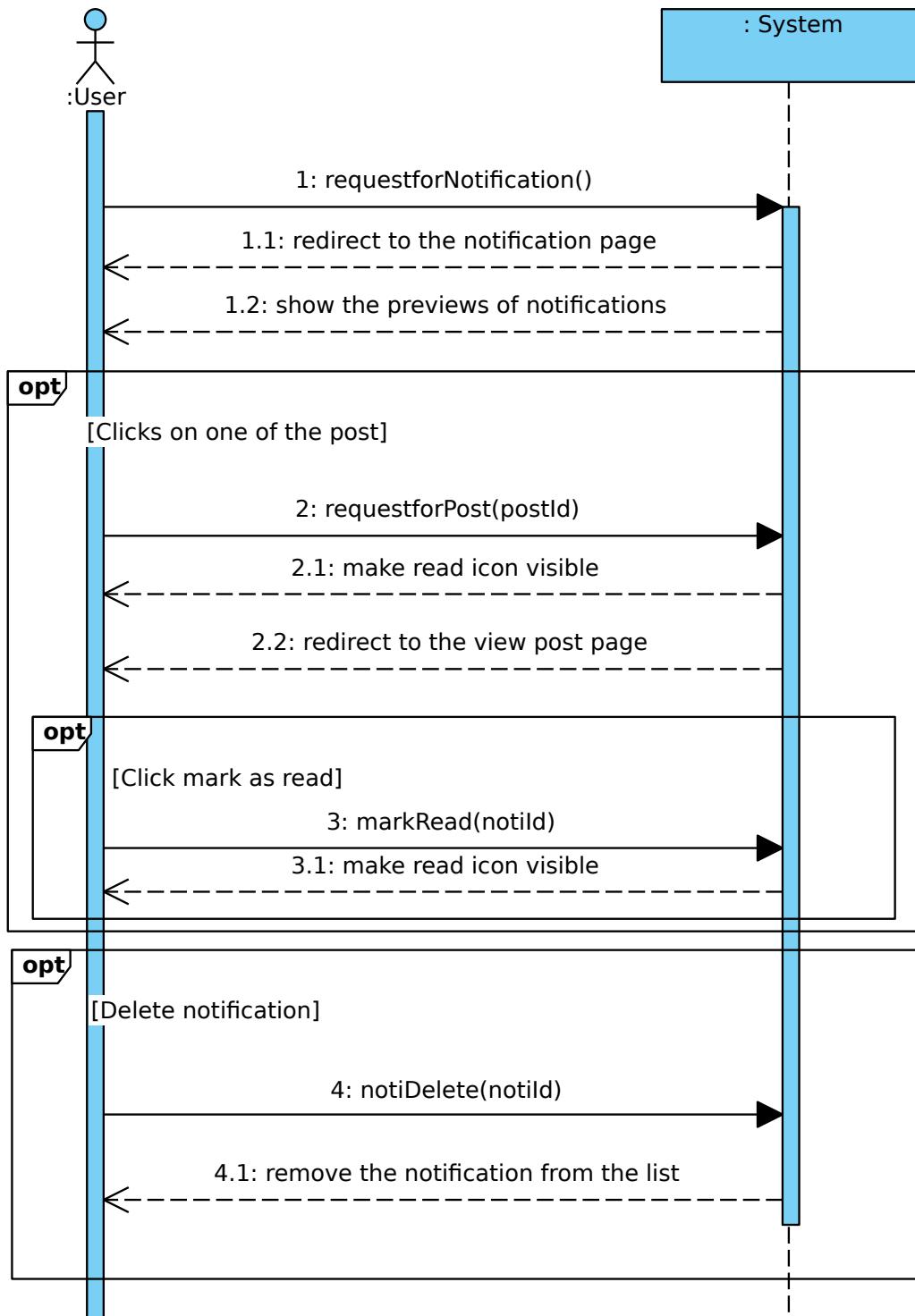
5.2 System Sequence Diagram(SSD): Create Event



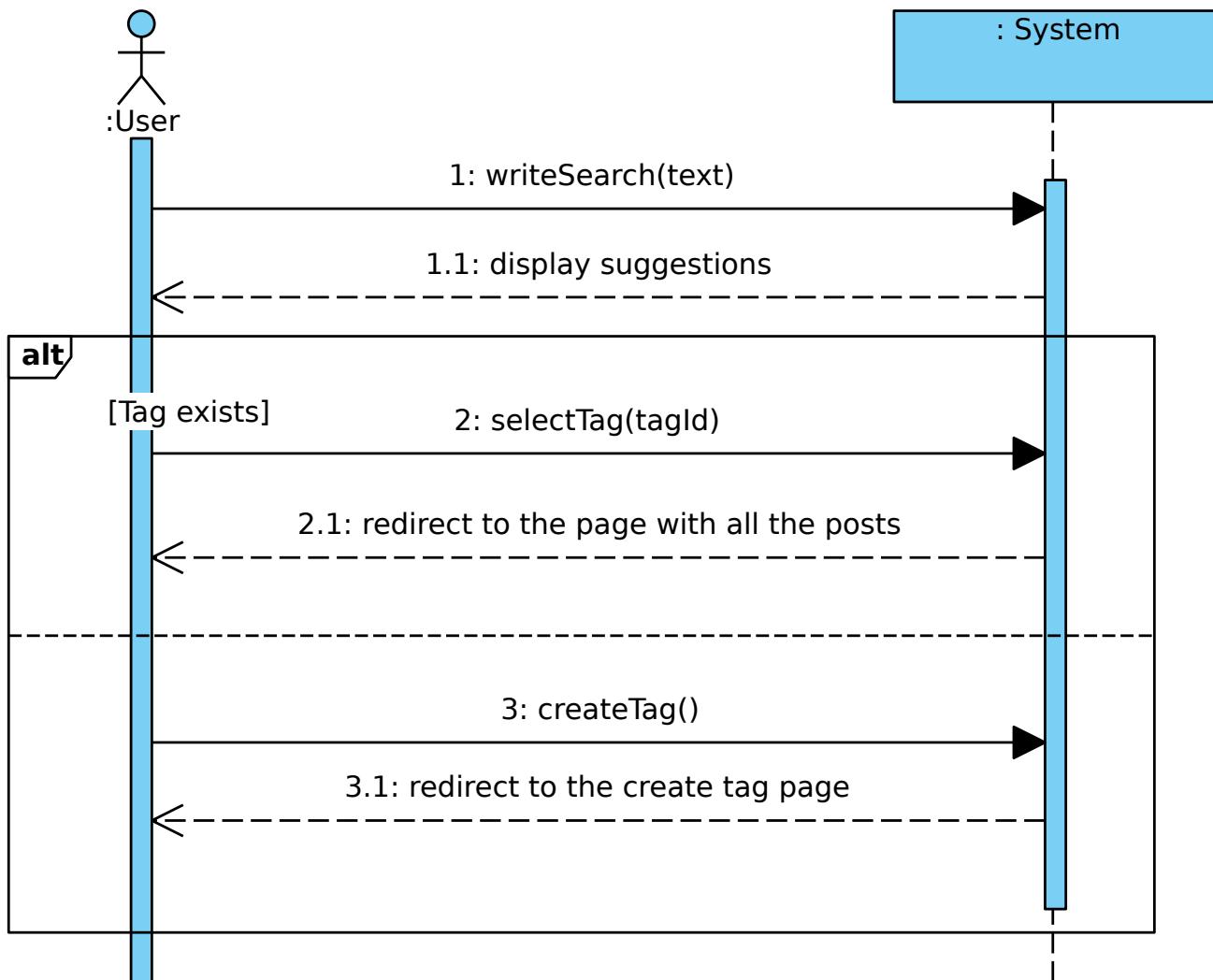
5.3 System Sequence Diagram(SSD): View Post



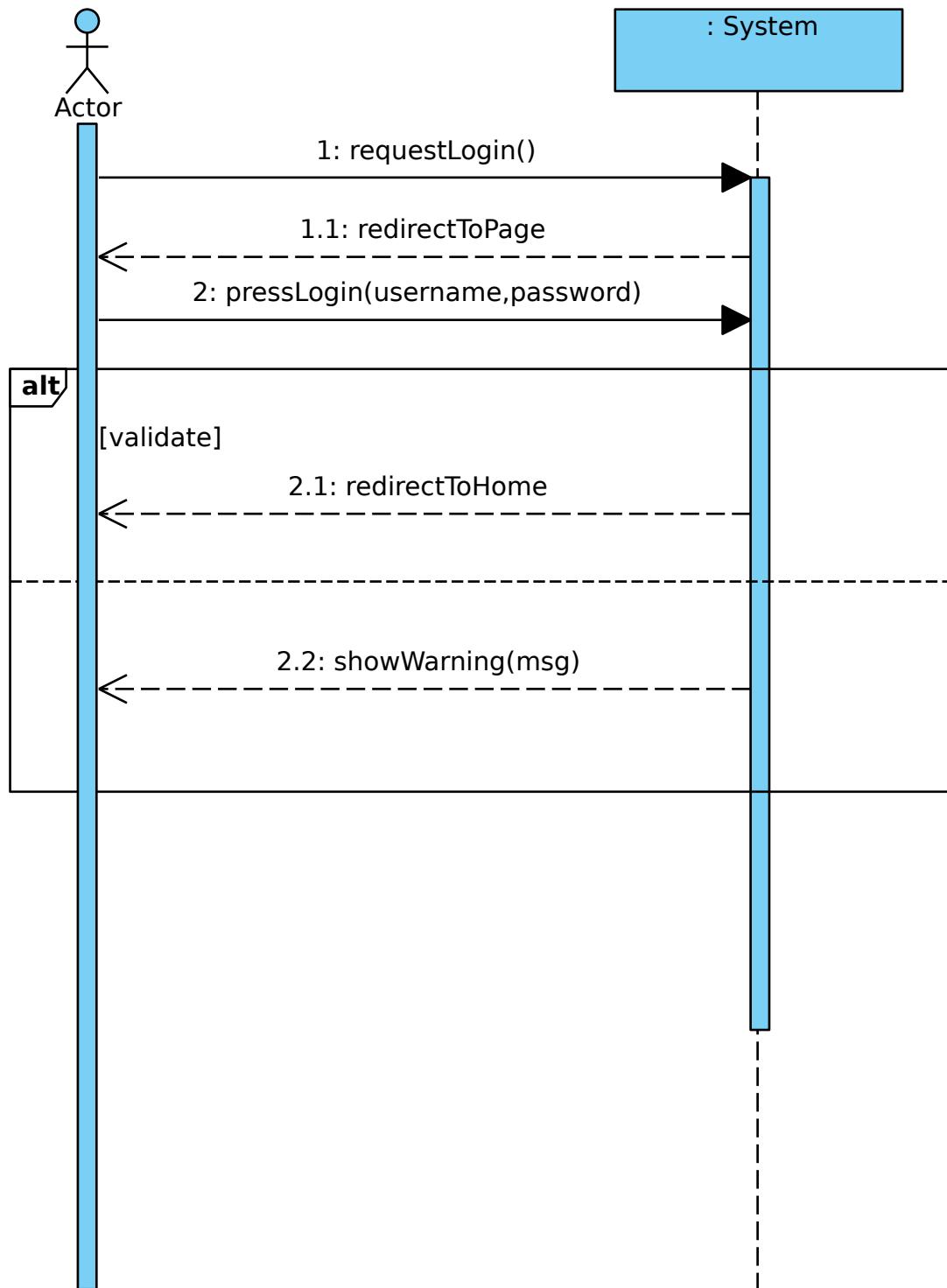
5.4 System Sequence Diagram(SSD): Notification



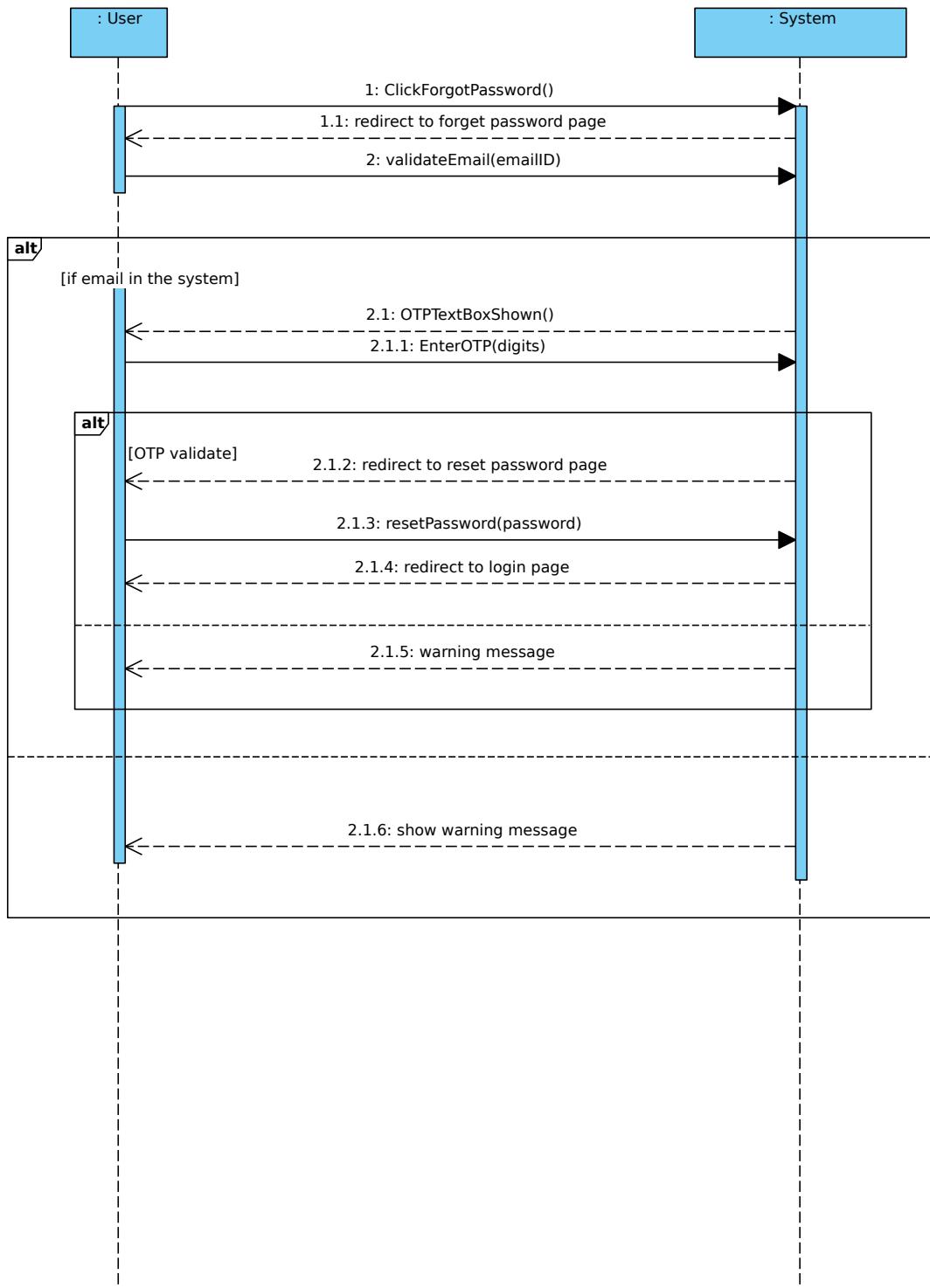
5.5 System Sequence Diagram(SSD): Search By Tag



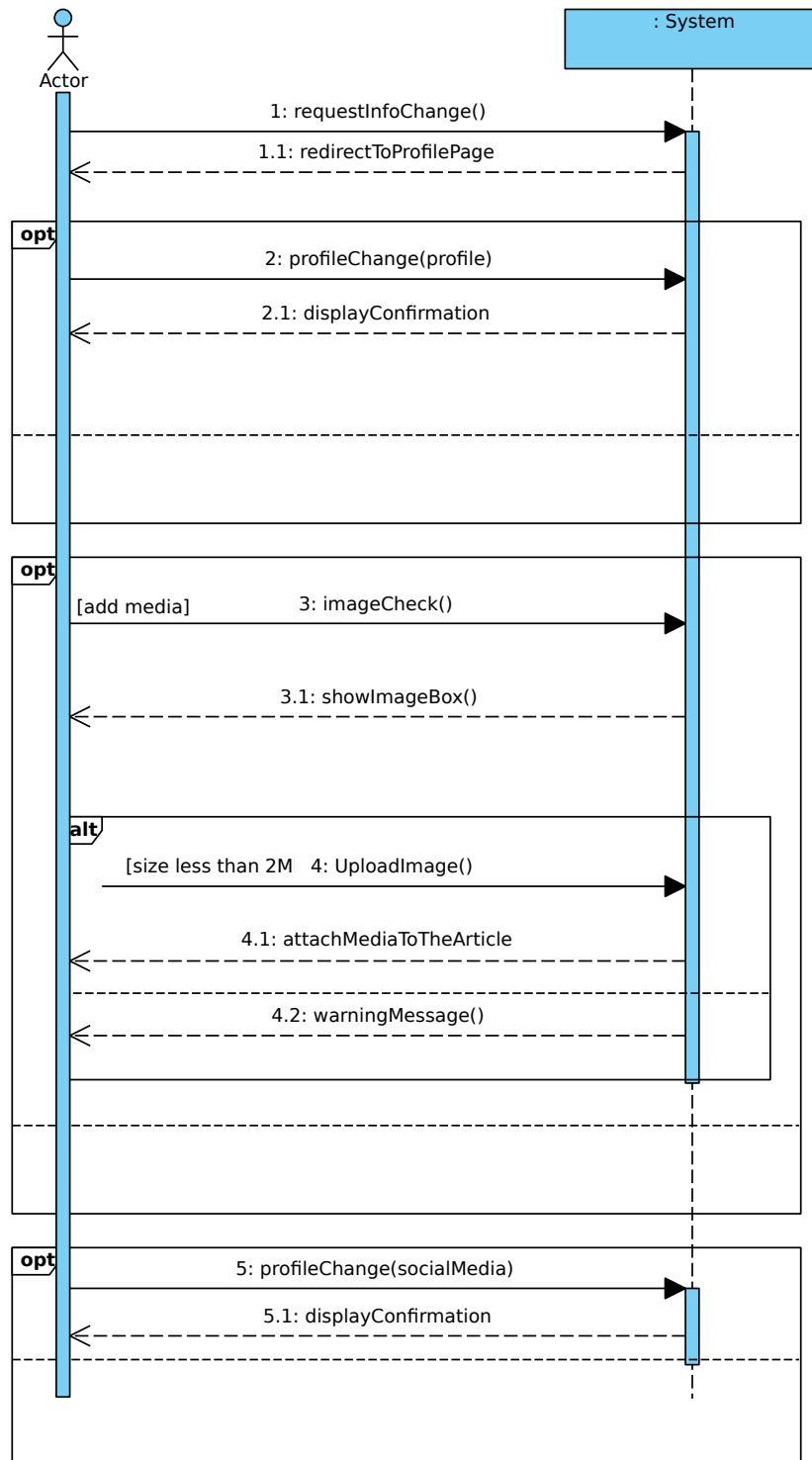
5.6 System Sequence Diagram(SSD): User Authentication



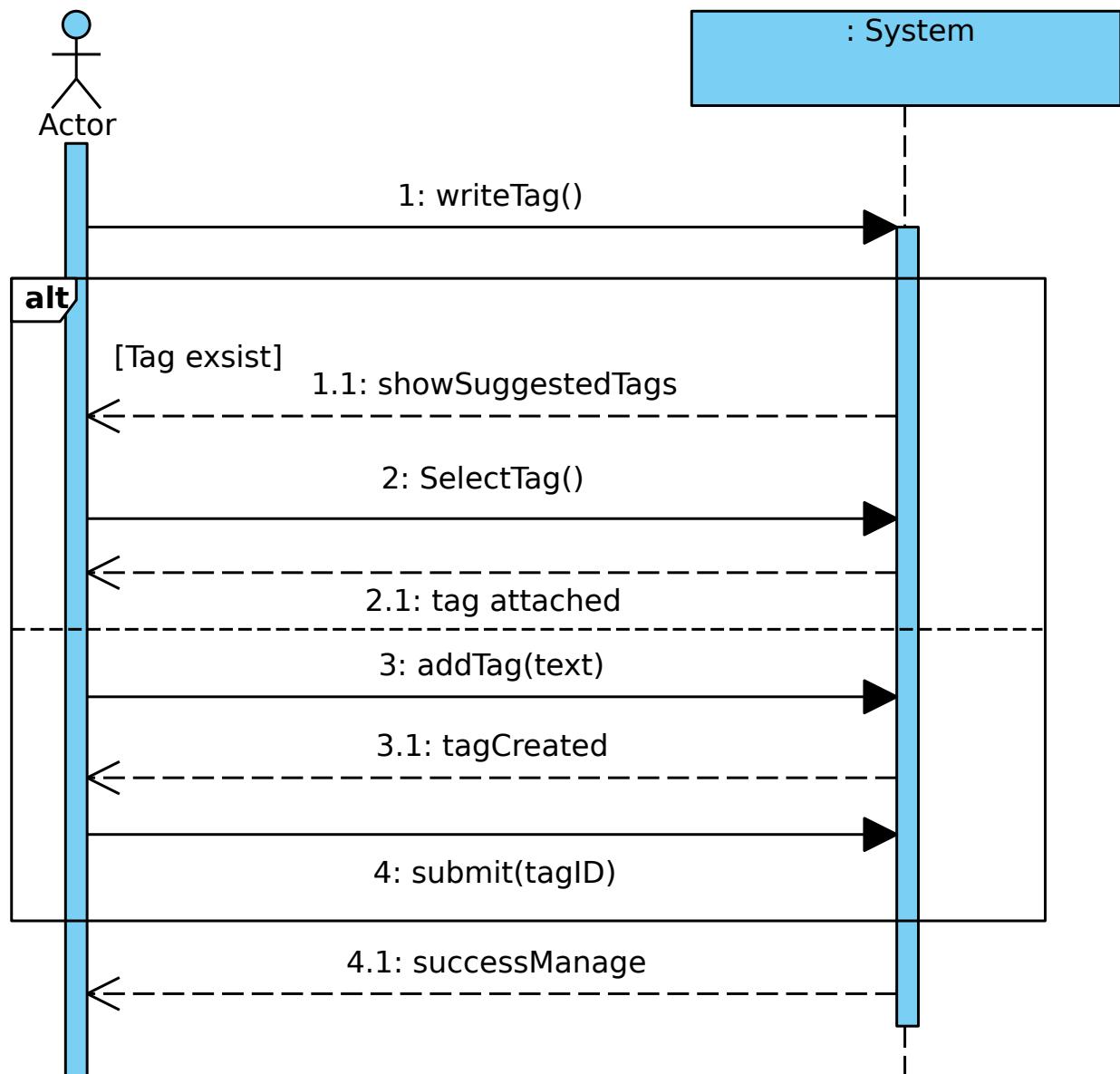
5.7 System Sequence Diagram(SSD): Forgot Credential



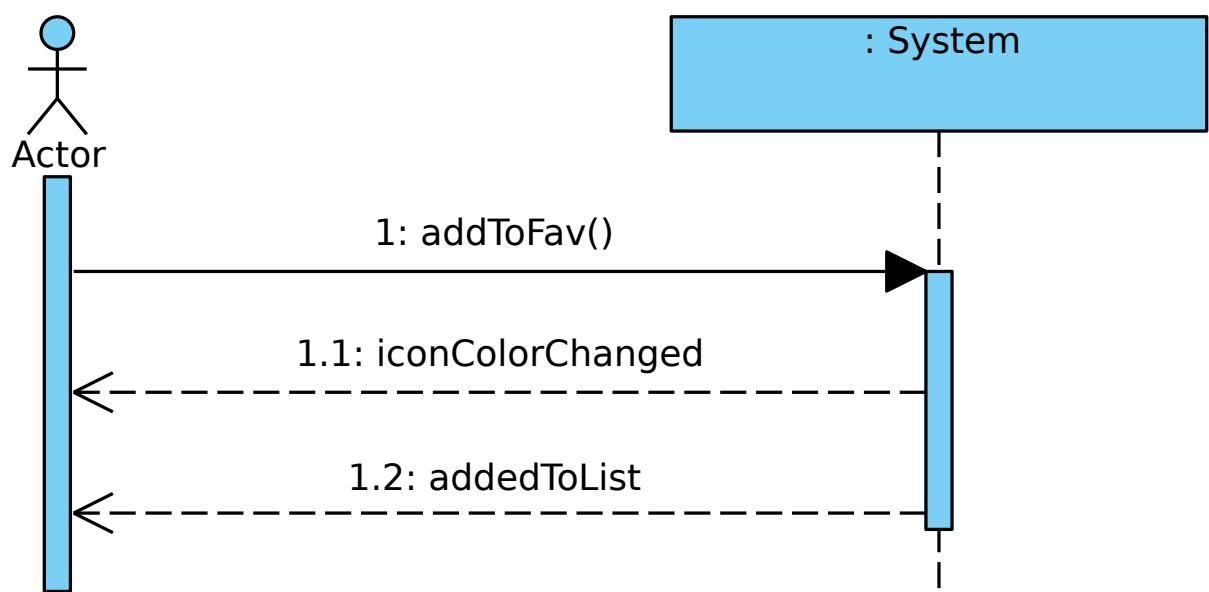
5.8 System Sequence Diagram(SSD): Change Information on User Profile



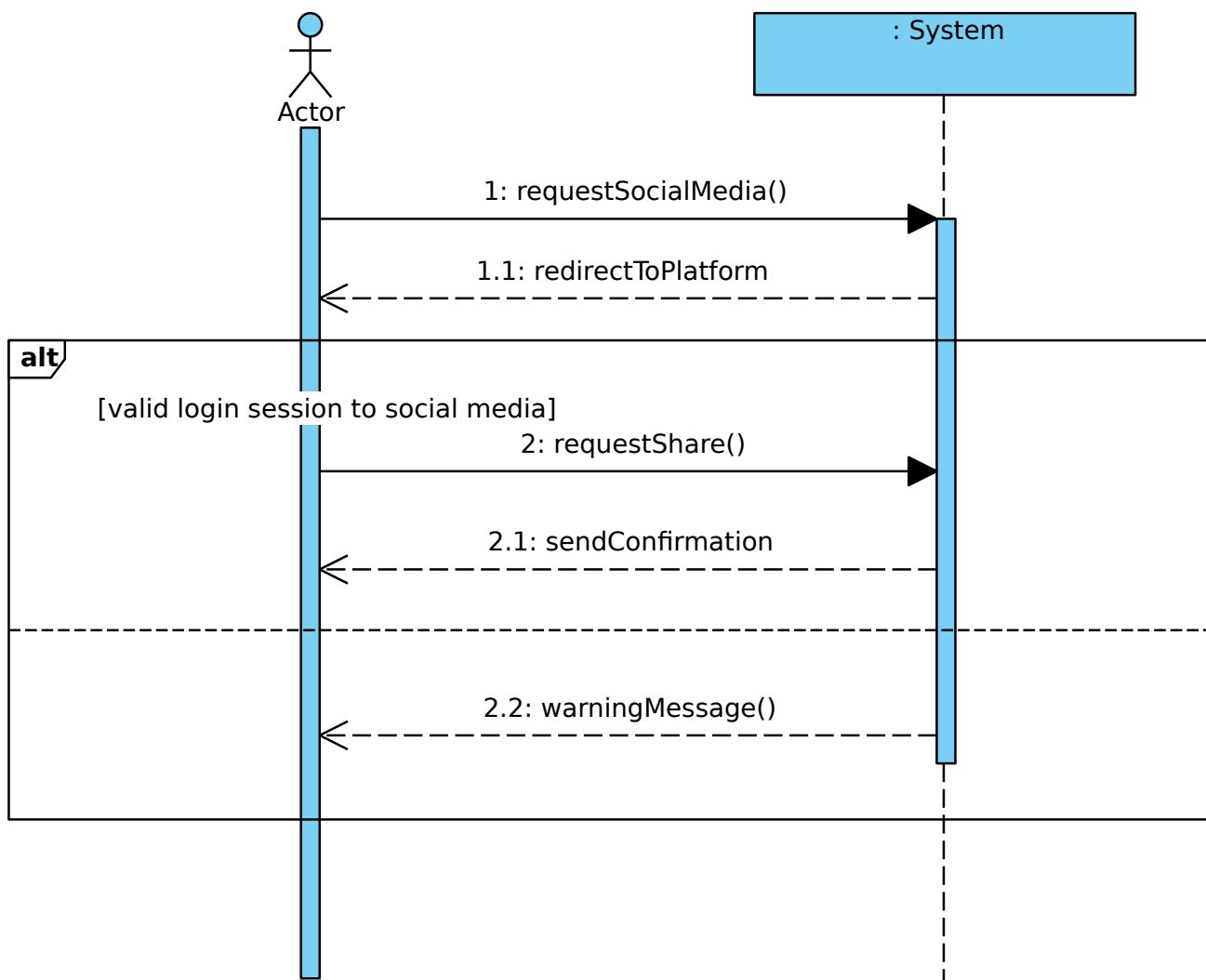
5.9 System Sequence Diagram(SSD): Add Tags to Article



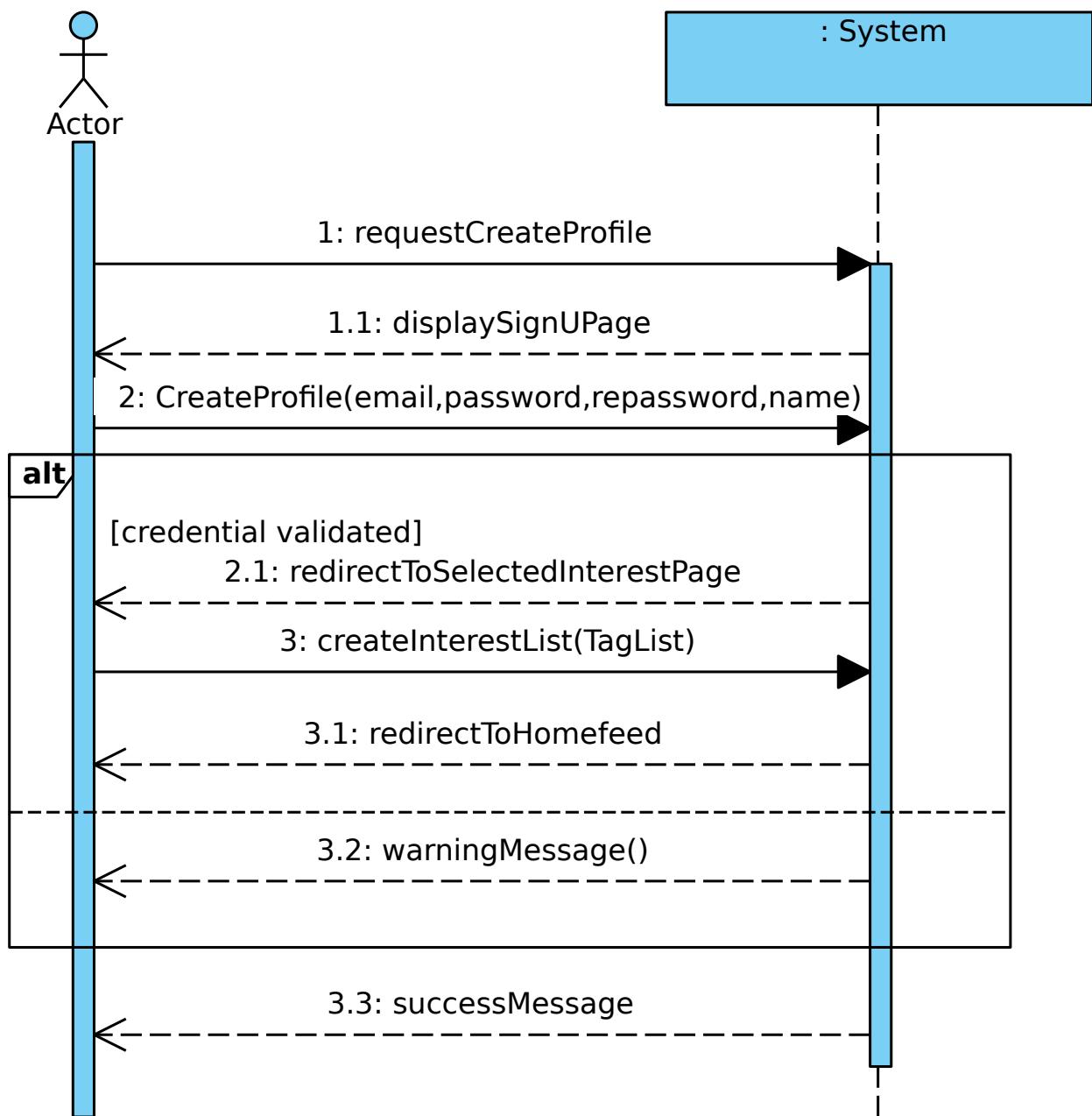
5.10 System Sequence Diagram(SSD): Add to the Favourite



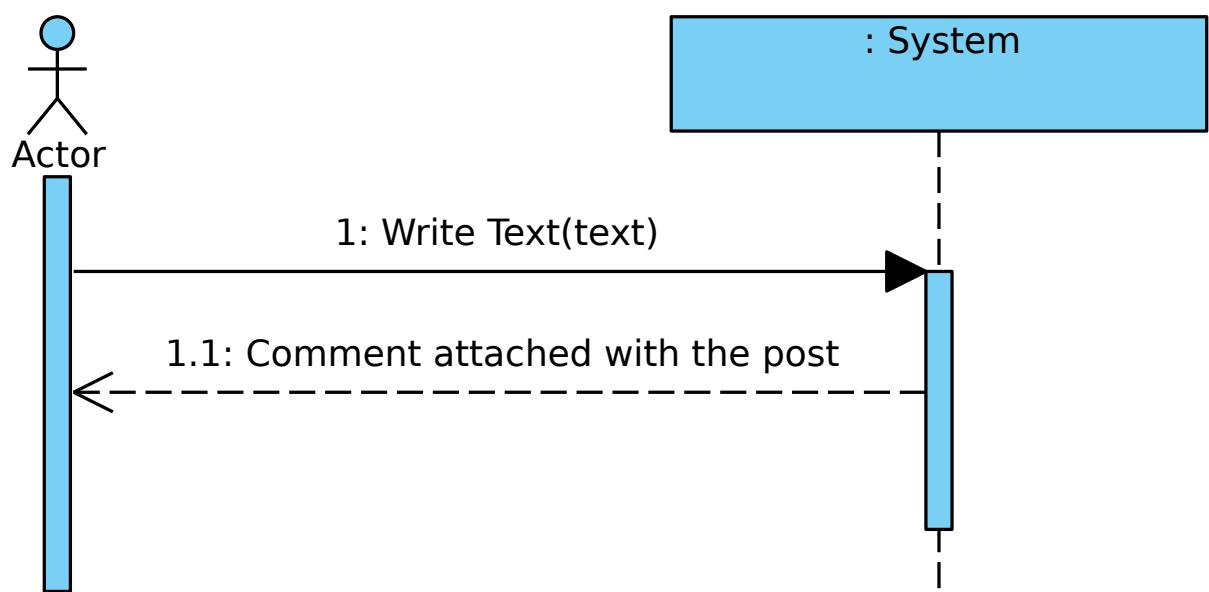
5.11 System Sequence Diagram(SSD): Share to Social Media



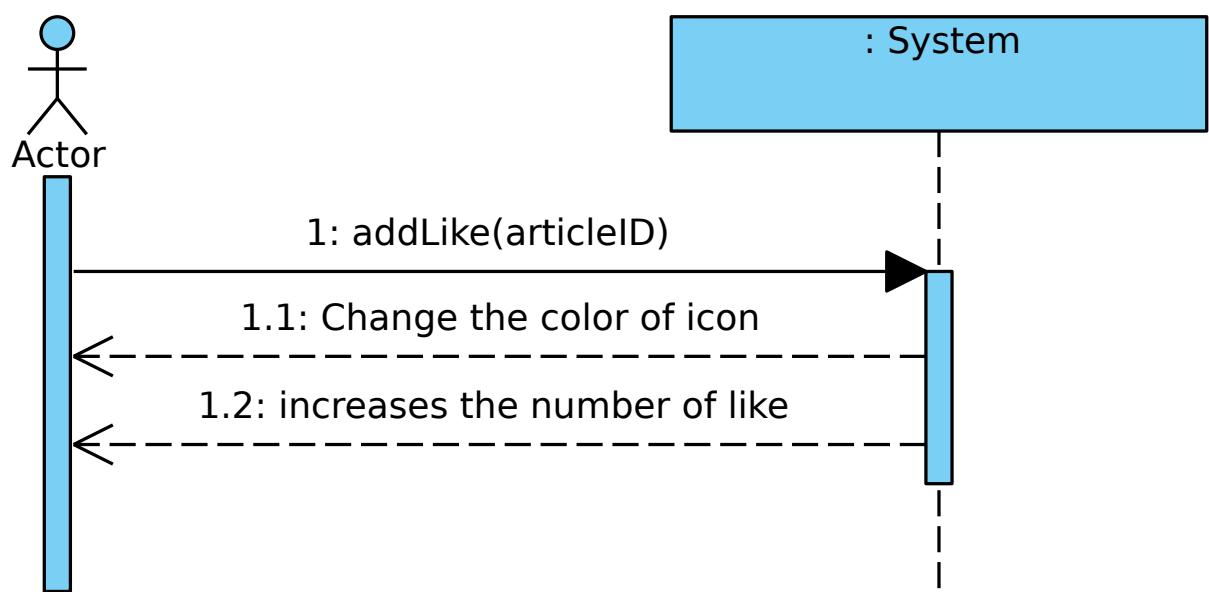
5.12 System Sequence Diagram(SSD): Create Profile



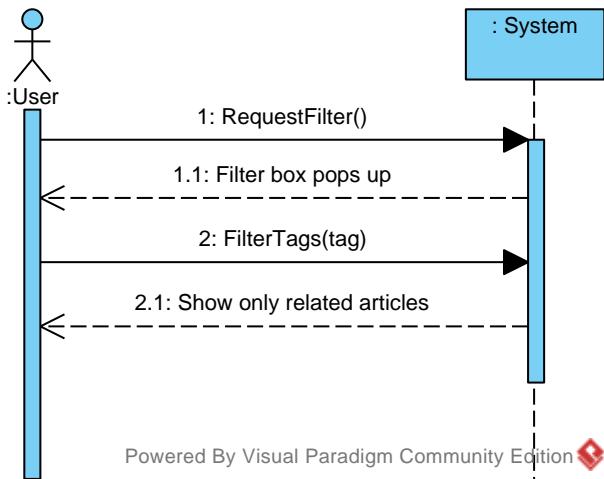
5.13 System Sequence Diagram(SSD): Add Comment to the post



5.14 System Sequence Diagram(SSD): Add like to the Article



5.15 System Sequence Diagram(SSD): Filter by Tags



6 System Operation

System
<pre>+writeSearch(text) +selectTag(tagId) +createTag() +viewNotification(notiID) +markRead(notiID) +notiDelete(notiID) +createArticle(article_id, text, tag) +addmedia(mediaID) +ArticleCancel() +checkDate(date) +setDate(date) +createEvent(text) +viewArticle() +createComment(title, description) +commentDelete(commentID) +favoriteList(articleID) +addLike(articleID, count) +login(email, password) +forgetPassword(email) +signup(userEmail, password, firstName, lastName, tagInteres +createProfile() +updateProfile(firstName, lastName, dateOfBirth) +credentialRecovery(userEmail, password) +authenticateUser(userEmail, password) +addFavorite(articleID)</pre>

7 Operation Contract

Contract CO: writeSearch
Operation: writeSearch(text: string) Cross Reference: Use cases: Search by Tag, create article Pre-conditions: user logged in and creating an article Post-conditions: <ul style="list-style-type: none">• A instance search was created• search was associated with tag.(association formed)• Search.text was set to text.
Contract CO: selectTag
Operation: selectTag(tagID: tagID) Cross Reference: Use cases: Create article Pre-conditions: user logged in and creating an article Post-conditions: <ul style="list-style-type: none">• A instance stag was created• stag was associated with user.(association formed)• stag was associated with article.(association formed)
Contract CO: viewNotification
Operation: viewNotification(notiID: notiID) Cross Reference: Use cases: Notification Pre-conditions: user logged in. Post-conditions: <ul style="list-style-type: none">• A instance noti was created• noti was associated with user.(association formed)
Contract CO: markRead
Operation: markRead(notiID: notiID) Cross Reference: Use cases: Notification Pre-conditions: user logged in and in the view notification page Post-conditions: <ul style="list-style-type: none">• A instance noti_mark was created• noti_mark was associated with user.(association formed)

Contract CO: notiDelete
Operation: notiDelete(notiID: notiID)
Cross Reference: Use cases: Notification
Pre-conditions: user logged in and in the view notification page
Post-conditions:
<ul style="list-style-type: none">• A instance noti_del was created• noti_del was associated with user.(association formed)

- A instance media was created
- media was associated with user(association formed)
- media was associated with article(association formed)
- media was associated with event(association formed)

Contract CO: checkDate
Operation: checkDate(Date: Date)
Cross Reference: Use cases: Create Event
Pre-conditions: user logged in and in the event creation page
Post-conditions:
<ul style="list-style-type: none">• A instance cdate was created• cdate was associated with event

- A instance sdate was created
- sdate was associated with event

Contract CO: viewArticle
Operation: viewArticle Cross Reference: Use cases: View Article Pre-conditions: user in the main page Post-conditions: <ul style="list-style-type: none">• A instance varcticle was created• varcticle was associated with article• Vertical was associated with the user

| **Contract CO: addFavourite** |
| **Operation:** addFavourite(articleID: articleID) **Cross Reference:** Use cases: Add to favourites **Pre-conditions:** user logged in and view the article |
| **Post-conditions:** |

- | |
| --- |
| - A instance fav was created - fav was associated with article - Vertical was associated with the user |

Contract CO1: CreateComment
Operation: CreateComment(title:String, description:String) Cross References: Use Cases:Comment event Pre-conditions: User request to create a event Post-conditions: <ul style="list-style-type: none">• A Comment instance evet was created. (instance creation)• Comt was associated with the Article, based on ArticleID match (association formed)

| **Contract CO1: CreateEvent** |
| **Operation:** CreateEvent(text:String) **Cross References:** Use Cases:Comment creation **Pre-conditions:** User request to write comment for an article |
| **Post-conditions:** |

- | |
| --- |
| - A Event instance evet was created. (instance creation) - evet was associated with the Notification (association formed) - evet was associated with the User (association formed) |

Contract CO2: CreateTag

Operation: CreateTag(name:String)

Cross References: Use Cases:Comment creation

Pre-conditions: User request to write comment for an article

Post-conditions:

- A Tag instance tag was created. (instance creation)
- Tag was associated with the User (association formed)

Contract CO2: ArticleCancel

Operation: PsotCancel()

Cross References: Use Cases:Create Article

Pre-conditions: User request to cancel the process of article creation

Post-conditions:

- A Article instance arti was created. (instance creation)
- arti was associated with the User (association formed)

Contract CO3: authenticateUser

Operation: UserValidation(useEmail :String, password: String)

Cross References: Use Case: User Authentication

Pre-conditions: User request to logged into the system

Post-conditions:

- Login instance LIU has been created.
- LIU was associated with the User (association formed)

Contract CO3: signUp

Operation: UserSignUp(useEmail :String, password: String, firstName: String, Last Name: String, tagInterest:<List>)

Cross References: Use Case: Sign Up

Pre-conditions: User request to be signed up into the system

Post-conditions:

- SignUp instance SIU has been created.
- SIU was associated with the User (association formed)
- SIU was associated with Tags(association formed)

Contract CO3: credentialRecovery
Operation: forgotPassword(useEmail :String, password: String) Cross References: Use Case: Forgot credential Pre-conditions: User gives wrong credentials and request for password reset Post-conditions: <ul style="list-style-type: none">• Credential Recovery instance CIU has been created.• CIU was associated with the User (association formed)

Contract CO3: updateProfile
Operation: updateInformation(firstName: String, lastName: String, dateOfBirth: Date, profilePicture: Blob) Cross References: Use Case: Profile Update Pre-conditions: User has to be logged on to the system Post-conditions: <ul style="list-style-type: none">• Profile instance PIU has been created.• PIU was associated with the User (association formed)• PIU was associated with media (association formed)• BirthDate attribute was set• Profile picture attribute was set

Contract CO3: favouritesList
Operation: addtoFavourites(articleId: Long) Cross References: Use Case: Add to favorites Pre-conditions: User has to be logged on to the system and click on the icon Post-conditions: <ul style="list-style-type: none">• Favorite instance FIU has been created.• FIU was associated with the User (association formed)• Add to favorites list was modified

Contract CO: CommentDelete
Operation: CommentDelete(commentID: commentID)
Cross Reference: Use cases: View Article
Pre-conditions: user logged in and try to deletes the comment he wrote
Post-conditions:
<ul style="list-style-type: none">• A instance comm_del was created• comm_del was associated with user

Contract CO: AddLike
Operation: Addlike(articleID: articleID, count:Integer)
Cross Reference: Use cases: View Article
Pre-conditions: user logged in and request to like an article
Post-conditions:
<ul style="list-style-type: none">• A instance liki was created• liki was associated with user• liki was associated with article• liki.count was set to count

8 CLASS RESPONSIBILITY COLLABORATOR (CRC) Model:

User	
Knows Email	Article
Article	
Knows First Name	Event
Knows Last Name	Tag
Knows Username	Notification
Have a Birth Date	Comment
Have an Image URL	Like
Has social media link	
Has favorites list	
Can create an article	
Create a new tag	
Can view another user's article	
Can comment on article	
Can delete comment from article	
Can like on an article	
Can view notification	
Can read notification	
Delete notification	
Can search for tags	
Can create an event	

Article	
Has content	User
Can contain media	Media
Must have a tag	Tag
Has number of likes	Comment
Number of comments	Notification
Has creation date	Article crawler
Has comments	Twitter crawler
Article creation can be notifying to user	
Can come from outside source	
Can drafted by user	

Event	
Has title	User
Has description	Notification
Has starting date	Media
Has ending date	
Has time frame	
Has location	
Can have media	
Event creation can be notifying to user	

Tag	
Has name Has creation date Tags can be auto suggested to user	Article User

Notification	
Has preview Can be marked as read Can be deleted Has a date	Article User Event

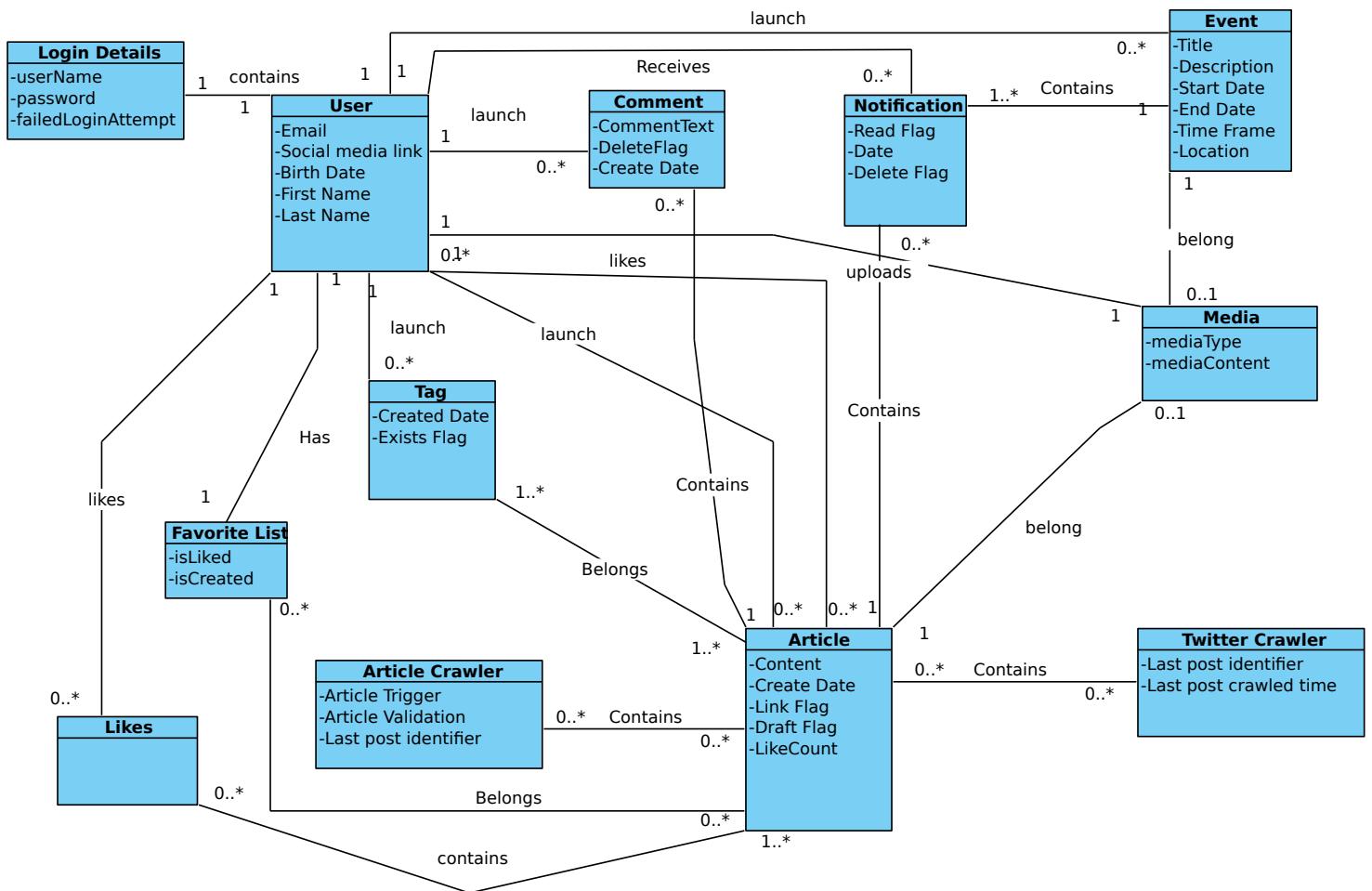
Comment	
Has text Can be deleted Has a creation date knows the article	Article User

Media	
Has file path	Article Event

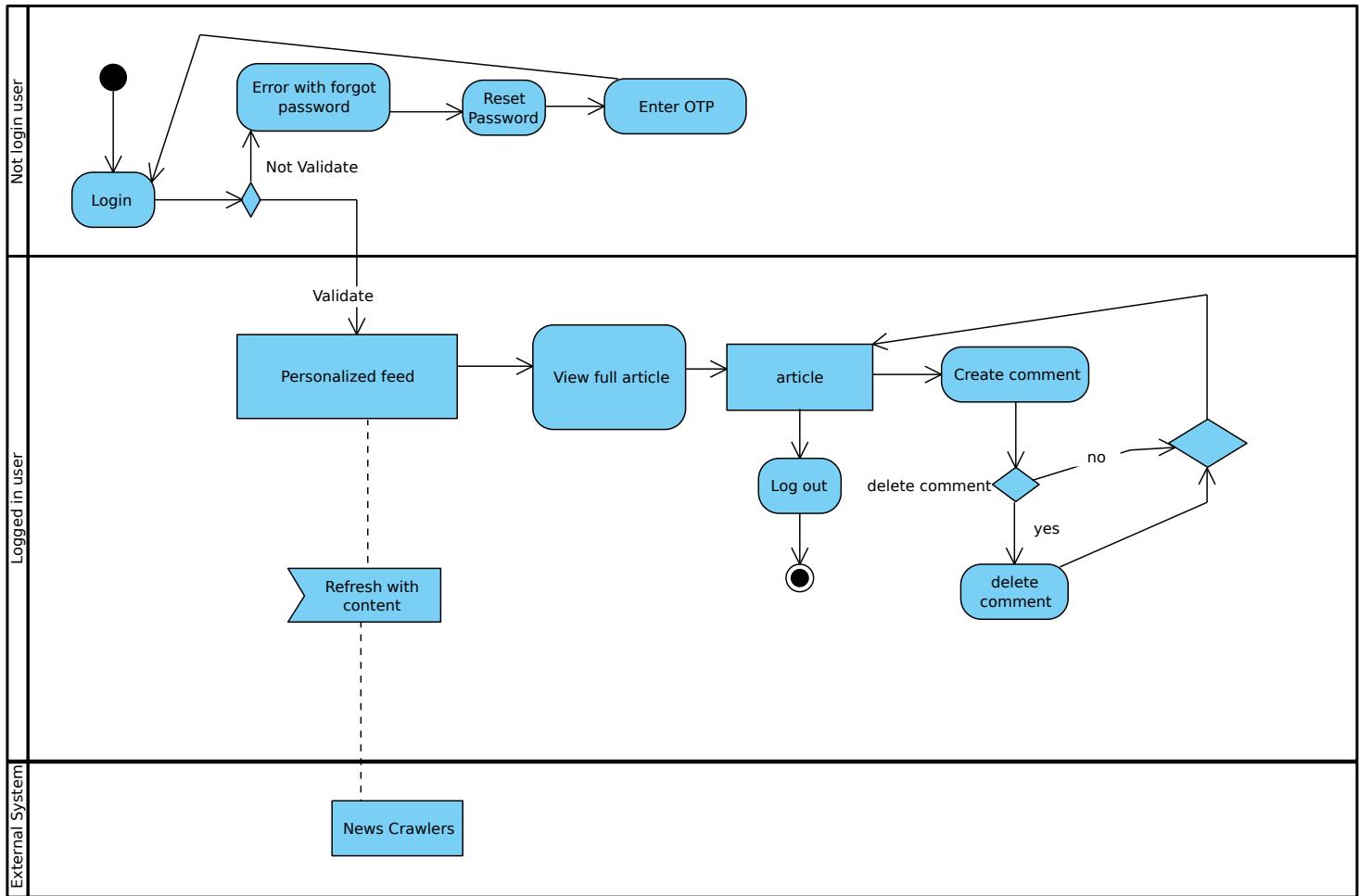
Article crawler	
Can search for article with a specific # tag Can filter article Can save article	Article

Twitter crawler	
Can trigger article Articles can be validated Can save article	Article

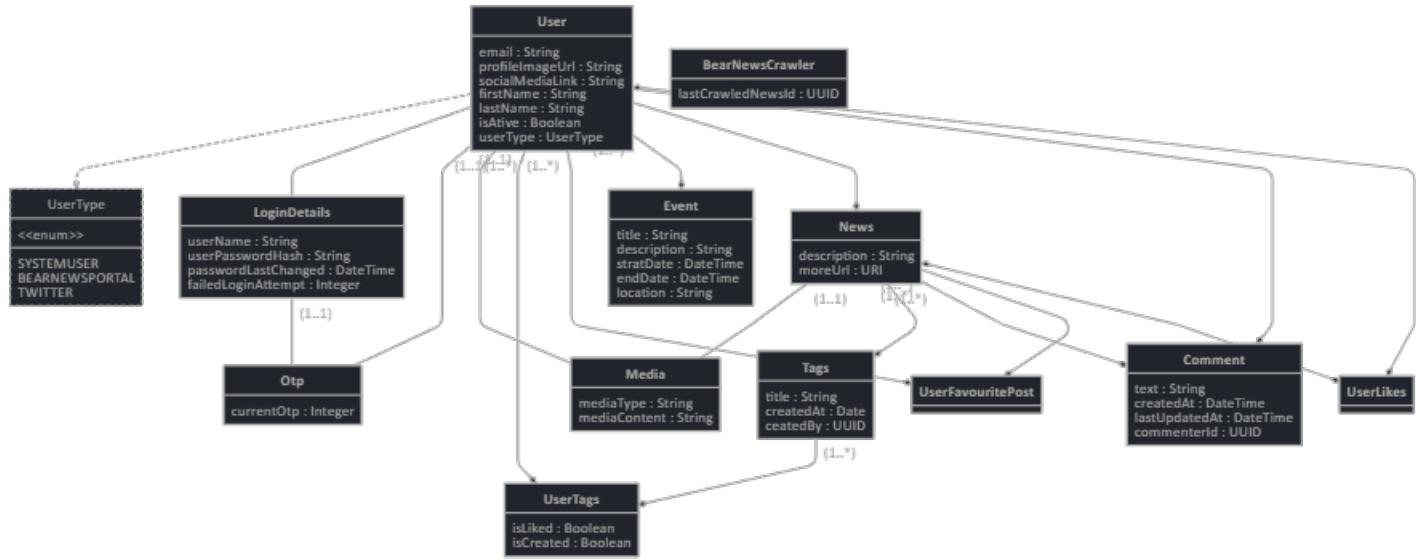
9 Domain Model



10 Activity Model(Create comment)



11 Data Model



12 Trello and Gantt chart

12.1 Trello

Beer Feed (A01)

- Product Backlog
- Spring Backlog
- In Progress
- 1.0 Sprint Complete

Beer Feed (A02)

- Product Backlog
- Spring Backlog
- In Progress
- 2.0 Sprint Complete

Documentation of the analysis phase

in list [5.0 Sprint Complete](#)

Members



Dates

Sep 21 - Sep 23 at 3:53 PM complete

Labels



Add to card

- Members
- Labels
- Checklist
- Dates
- Attachment
- Cover
- Custom Fields

Add dropdowns, text fields, dates, and more to your cards.

Upgrade

Subtasks

100% 

- presentation
- work-on-review
- redraw-activity-diagram
- domain-change
- use-case-change
- use-case-diagram-change

Add an item

Log Work

Log Work

Add Power-Ups

Automation

Add button

Activity

Show details

Design Modeling (Diagramming)

in list [5.0 Sprint Complete](#)

Members Labels

Dates [complete](#)

Description [Edit](#)
find the input to activities in implementation and test.

Subtasks [Hide checked items](#) [Delete](#)

100%

- tools
- assign
- draw
- jpa-modeling
- git-repo-add

[Add an item](#)

Activity [Show details](#) [Automation](#)

Add to card

- Members
- Labels
- Checklist
- Dates
- Attachment
- Cover
- Custom Fields

Add dropdowns, text fields, dates, and more to your cards. [Upgrade](#)

Power-Ups

- Log Work
- Log Work

[+ Add Power-Ups](#)

12.2 Trello Work hour

Name	Arju	Sadia	Noorah	Mushfika	Tingshuo
Time	57	58.5	42	58.1	57.7

12.3 Gantt Chart

Bear Feed

Sep 23, 2022

<http://>

Project manager

Aug 24, 2022 - Dec 15, 2022

Completion

9%

Tasks

30

Resources

5

Bear Feed

Sep 23, 2022

Tasks

Name	Begin date	End date
Analysis	8/24/22	9/21/22
Project Vision	8/24/22	8/24/22
Team Assembly	8/25/22	8/25/22
Infrastructure initialization	8/26/22	8/29/22
Requirements analysis	8/30/22	9/1/22
Use cases/Traceability matrix	9/1/22	9/7/22
System operations	9/5/22	9/7/22
Use cases	9/1/22	9/7/22
System sequence diagrams	9/1/22	9/7/22
Wireframes	9/8/22	9/9/22
Domain model	9/12/22	9/13/22
Activity Diagram	9/14/22	9/20/22
Presentation and reporting	9/21/22	9/21/22
Design	9/22/22	10/5/22
Sequence diagrams	9/22/22	9/23/22
Design model	9/26/22	9/28/22
Package diagrams	9/29/22	10/3/22
Test coverage	9/28/22	9/28/22
Components and deployment	10/3/22	10/3/22
Presentation and reporting	10/4/22	10/5/22
Implementation	10/6/22	12/14/22
Backend	10/6/22	12/7/22
User Interface	10/6/22	12/7/22
Unit-Testing	10/6/22	12/7/22
Real data testing	12/1/22	12/7/22
User input validation	12/1/22	12/5/22
Imports/exports	12/1/22	12/5/22

Tasks

Name	Begin date	End date
System Test (with production deployment)	12/5/22	12/13/22
Documentation	12/1/22	12/9/22
Presentation and reporting	12/12/22	12/14/22

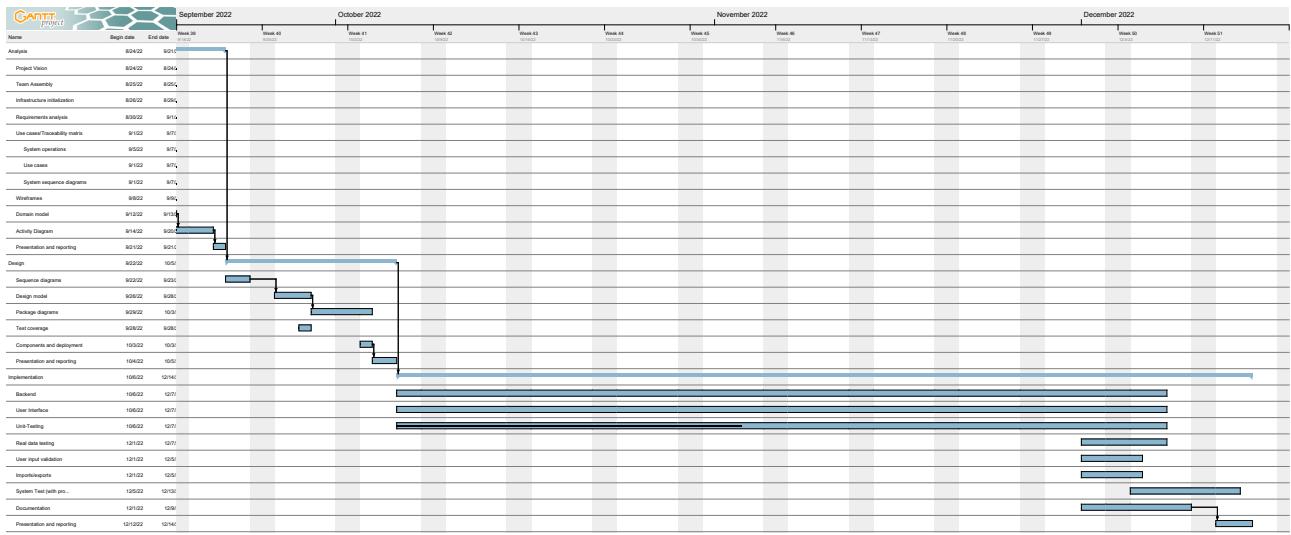
Resources

Name	Default role
Arju	Team Leader
Scott	Analyst
Tisha	Developer
Rahman	Developer
Noorah	Tester

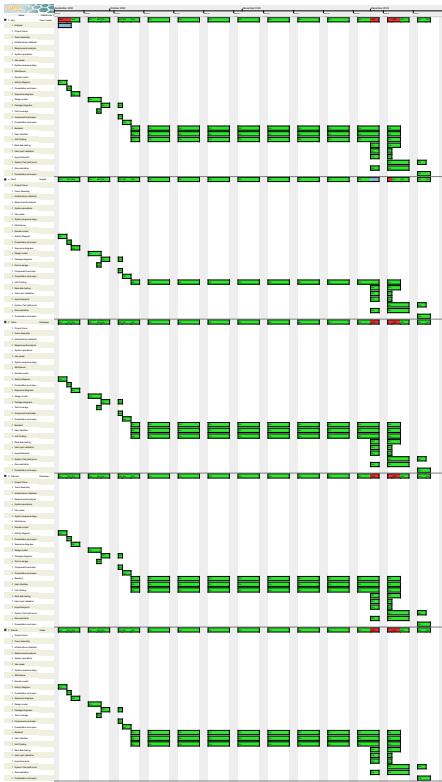
Bear Feed

Sep 23, 2022

Gantt Chart



5

Resources Chart

13 Github

GitHub repository: [↗](#)