

## Table of Contents

### [Project Part II](#)

#### [Team:](#)

#### [Project Requirements](#)

#### [Business Requirements](#)

#### [User Requirements](#)

#### [Functional Requirements](#)

#### [Non-Functional Requirements](#)

#### [Use Cases](#)

#### [Actors](#)

#### [Use Case Overview](#)

#### [Use Case Documents](#)

##### [UC-001](#)

##### [UC-002](#)

##### [UC-003](#)

##### [UC-004](#)

##### [UC-005](#)

##### [UC-006](#)

##### [UC-007](#)

##### [UC-008](#)

##### [UC-009](#)

##### [UC-010](#)

##### [UC-011](#)

##### [UC-012](#)

##### [UC-013](#)

##### [UC-014](#)

##### [UC-015](#)

##### [UC-016](#)

##### [UC-017](#)

##### [UC-018](#)

##### [UC-019](#)

##### [UC-020](#)

##### [UC-021](#)

##### [UC-022](#)

##### [UC-023](#)

##### [UC-024](#)

##### [UC-025](#)

##### [UC-026](#)

##### [UC-027](#)

#### [Activity Diagram](#)

#### [Data Storage](#)

#### [UI Mockups](#)

## Project Part II

### Team:

- Yue Zhang
- Jeannette Pepin
- Keyu Chen

**Title:** Timelines

**Project Summary:** A personal journaling app that allows a user to view life events and changes as time series graphs. Users can make quick entries to track life events, thoughts, and mood using tags, images, and text. These entries can also be shared with other users.

### Project Requirements

#### Business Requirements

ID	Requirement	Topic Area	User	Priority
BR-01	User data is completely removed upon deletion	Security	All	Medium
BR-02	Admin privileges can be added or removed when employment status changes	HR	All	High

#### User Requirements

ID	Requirement	User
	As a user, I want to...	
UR-01	Edit my profile	User

UR-02	Log in and out of the app	User
UR-03	View my past entries	User
UR-04	Edit the privacy of an entry	User
UR-05	Create a new post	User
UR-06	Add tags to a post	User
UR-07	Delete a post	User
UR-08	Delete a tag from a post	User
UR-09	Search for posts by tag	User
UR-10	View statistics of posts and activity	User
UR-11	Change my password	User
UR-12	Choose a nickname to display	User
UR-13	View posts of other users	User
UR-14	View the profiles of other users	User
UR-15	Comment on others' posts	User
UR-16	Delete my comments on posts	User
UR-17	Send a friend request	User
UR-18	Accept a friend request	User
UR-19	Reject a friend request	User
UR-20	Remove a friend	User
UR-21	Request my account be deleted	User
UR-22	View system usage statistics	Admin
UR-23	Modify user access privileges	Admin
UR-24	Receive notifications of activity relevant to me in the UI	All
UR-25	Delete a user	Admin
UR-26	Sign up	User

**Functional Requirements**

ID	Requirement
	The system should:
FR-01	Remember logins for one week
FR-02	Allow creation of a new user by anyone
FR-03	Prevent users from deleting their own account
FR-04	Prevent creation of accounts with the same username
FR-05	Prevent creation of accounts with the same email
FR-06	Prevent text content of a previous post from being edited
FR-07	Allow multiple tags to be created on a post
FR-08	Save information when offline (post creation)
FR-09	Restore saved information when internet connection restored
FR-10	Allow viewing of public posts before logging in
FR-11	Prevent users from accessing admin functionality
FR-12	Limit post text length to 140 characters
FR-13	Track counts of posts and likes
FR-14	Prevent users from viewing posts not shared with them
FR-15	Autofill tags when user begins typing them
FR-16	Block user login for 5 minutes after 5 failed logins
FR-17	Allow user to change password (once logged in)
FR-18	Allow user to reset password (before login)
FR-19	Provide Admin or User UI options based on account type
FR-20	Remove all user data upon account deletion
FR-21	Save drafts temporarily after x seconds and offer to restore in case of crash or error

FR-22	Allow User to request account deletion
FR-23	Allow Admin to delete User accounts

### Non-Functional Requirements

ID	Requirement	Category
NR-01	System should be simple to use (limited number of options for posting and viewing)	Usability
NR-02	System is responsive to mobile or desktop	Usability
NR-03	System should fail gracefully when users attempt a disallowed action (such as viewing content they don't have permissions to)	Reliability
NR-04	System should maintain 99% uptime	Reliability
NR-05	Each user can only manage the information on their own account	Operation
NR-06	The system should not lose data (draft posts) in the case of unexpected failure or a restart	Reliability
NR-07	User should be redirected to profile within 5 seconds of logging in	Performance
NR-08	System will timeout after 10 number of seconds	Performance
NR-09	75% users can be logged in simultaneously	Performance
NR-10	The system should load entries in user feed within 2 seconds	Performance
NR-11	The system should load searches or time series graph pages within 4 seconds	Performance
NR-12	Email address should be validated before account is created	Security
NR-13	Users can only log in with correct username and password	Security
NR-14	Passwords are hashed and salted in database	Security

## Use Cases

1. UC-001 User can create a new post by uploading a picture and tagging the post
2. UC-002 User can send a friend request to another user
3. UC-003 User can make a text post and add tags without image
4. UC-004 User can “like” her friend’s post
5. UC-005 User can view how many likes they have given and received
6. UC-006 User can view a graph of how many posts they have made with a specific tag
7. UC-007 User can view posts on their dashboard by tag
8. UC-008 Admin can delete any user’s account
9. UC-009 User can modify the privacy setting of a post
10. UC-010 User gets a notification when someone likes their post
11. UC-011 User can delete their old entries.
12. UC-012 User can view the notification for new friend request and accept or reject
13. UC-013 User can delete tags from an entry
14. UC-014 Admin can change account access level of another User
15. UC-015 User can edit their profile and choose a nickname
16. UC-016 User can change their password
17. UC-017 User can comment on their friend’s post
18. UC-018 User can view profiles of other users
19. UC-019 User can delete a comment they have left on a post
20. UC-020 User can remove a friend
21. UC-021 User can log in
22. UC-022 User can log out
23. UC-023 Admin can view system usage statistics
24. UC-024 User can request account deletion
25. UC-025 Admin gets notified when a User requests deletion and deletes User
26. UC-026 User can sign up
27. UC-027 Admin can view Admin Console

## Actors

Users, Admins

## Use Case Overview

[Click here](#) to view diagram

## Use Case Documents

Use Case ID:	UC-001
--------------	--------

Use Case Name:	Create a post
Description:	User can create a new post by uploading a picture and tagging the entry

Actors:	Users		
Pre-conditions:	User is logged in		
Post-conditions:	User has created an image post with a tag		
Frequency of Use:	Daily by users		
Flow of Events:		Actors:	System Response:
	1	User clicks “new post” button in menu bar	Displays “New Post” box
	3	User clicks on ‘img’ icon.	Displays “upload image” dialogue box
		User chooses image to add	Saves image to post
	4	Type a tag into the tags field	Add the tag to database if not exist
	5	Click on the ‘post’ button.	Upload the information to database and display new post in feed
Variations:	User can click cancel at anytime		
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-002
Use Case Name:	Add a friend
Description:	User can send a friend request to another user

Actors:	Users
Pre-conditions:	User is logged in, another user exists
Post-conditions:	Both user are friends to each other and view privacy posts

Frequency of Use:	Daily by Users		
Flow of Events:		Actors:	System Response:
	1	User A search user B in search bar based on their username	List the searched user of that username
	2	User A click on 'add' button.	Send user B with a request
	3	User B confirm request	Add user B's id to user A's User.Friend list. Add user A's id to user B's User.Friend list.
Variations:	User B can also reject the request		
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-003
Use Case Name:	User can make a text post and add tags without image
Description:	User can make a text post and add tags without image

Actors:	Users		
Pre-conditions:	User is logged in		
Post-conditions:	A text post has been created and it has tags		
Frequency of Use:	Daily by users		
Flow of Events:		Actors:	System Response:
	1	Click "write" icon to create new post	Show new post creation window
	2	Click tag text box and type i tags	System autocompletes already existing tags
	3	Click "send" button to publish post	System saves post and displays on feed



Variations:	
Notes and Issues:	
Developer Notes:	

Use Case ID:	UC-004
Use Case Name:	User can “like” her friend’s post
Description:	User can click likes button if he / she finds the post interesting

Actors:	Users		
Pre-conditions:	User is logged in		
Post-conditions:	Database record the likes action		
Frequency of Use:	Depends on individual user’s behavior and preference		
Flow of Events:		Actors:	System Response:
	1	Click the “world” button	Go to the main page
	2	Browse others’ post	
	3	Click “likes” button	Record the likes action
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-005
Use Case Name:	User can view how many likes they have given and received
Description:	User views a graph of how many posts they made last month, and filters the view by a specific tag

Actors:	Users
---------	-------

Pre-conditions:	User is logged in		
Post-conditions:	User can view graphs		
Frequency of Use:	Daily by Users		
Flow of Events:		Actors:	System Response:
	1	User clicks "Stats" button in menu bar	Display the statistics page
	2	User see the statistics (How many likes you have got and how many likes you have sended) once get into the statistics page.	
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-006
Use Case Name:	Filter graph view by tag
Description:	User can view a graph of how many posts they have made with a specific tag

Actors:	Users		
Pre-conditions:	The user is logged in. The user has made at least one post and the posts have been tagged.		
Post-conditions:	User views graph that shows a visual of posts that were tagged with the specified tag		
Frequency of Use:	Daily by users		
Flow of Events:		Actors:	System Response:
	1	Click on "Stats" button in the menu bar	Displays stats page
	2	Select "posts" graph and type name of tag in	Autocomplete if tag exists in database for

		"tags" field to filter	user
		Click "filter" button	Display graph of posts tagged with specified tag
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-007
Use Case Name:	Filter dashboard posts by tag
Description:	User can view posts on their dashboard by tag

Actors:	Users		
Pre-conditions:	The user is logged in. There are posts that the user has access to view (public or shared) and that are tagged with the specified tag		
Post-conditions:	User views posts in feed that are tagged with specified tag		
Frequency of Use:	Daily by users		
Flow of Events:		Actors:	System Response:
	1	User types tags into search bar in main menu bar	Autocomplete tags if they exist in the database already on posts user can view
	2	User hits "enter" or search button when done typing	Displays posts tagged with specified tag
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-008
Use Case Name:	Admin deletes user account
Description:	An admin user can delete another user's account

Actors:	Admin		
Pre-conditions:	The user exists and the Admin is logged in		
Post-conditions:	User account and data is removed from the system		
Frequency of Use:	Daily or less often by Admins		
Flow of Events:		Actors:	System Response:
	1	Open settings menu and click "Admin Panel" button	Displays Admin dashboard
	2	Select User tab	Display User section
	3	Search for user by name in search bar	Displays matching results
		Select User	Display User Admin controls panel
	4	Click "Remove Account" section from Users Admin Controls menu	Display Account Removal section
	5	Click "Delete Account" button	System displays confirmation prompt
	6	Confirm deletion of user	System deletes user account and data
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-009
Use Case Name:	User modifies post privacy
Description:	A user can modify the privacy setting of a post they own

Actors:	User		
Pre-conditions:	The user is logged in and has previously made at least one post		
Post-conditions:	The post has updated privacy settings		
Frequency of Use:	Daily or more by users		
Flow of Events:		Actors:	System Response:
	1	Scroll to post in feed	Displays post
	2	Click privacy dropdown menu on post	Shows privacy dropdown menu
	3	Select desired privacy level for post	Post icon updates to reflect new setting
Variations:	1B User can search for post by tag		
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-010
Use Case Name:	Notification of user activity
Description:	User gets a notification when someone likes their post

Actors:	User
Pre-conditions:	The user is logged in. Someone has liked their post.
Post-conditions:	The user gets a notification in the UI display
Frequency of Use:	Multiple times daily by users

Flow of Events:		Actors:	System Response:
	1	User B likes User A's post	Saves "like" and triggers notification
	2		Displays "new notification" icon in UI menu tray to User A
	3	User clicks notification icon to view notification	Displays notification
	4		Marks notification as seen
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-011
Use Case Name:	Delete an entry
Description:	Users can delete their old entries

Actors:	Users		
Pre-conditions:	Users is logged in and target entry exists		
Post-conditions:	The entry is removed from database.		
Frequency of Use:	Daily by users.		
Flow of Events:		Actors:	System Response:
	1	User navigates to post in feed	
	2	Click on 'delete' button on post	Removes post from database along with tags
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-012
Use Case Name:	User can view the notification for new friend request and accept or reject
Description:	User can view the notification for new friend request and accept

Actors:	Users		
Pre-conditions:	Users must login, target entry must exist		
Post-conditions:	The two users are now friends		
Frequency of Use:	Daily or hourly by users		
Flow of Events:		Actors:	System Response:
	1	Click the notification icon	Displays notification page
	2	Click the "friend request" notification	Display the request with the profile of the requester.
	3	Click "accept" button	Show request as read
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-013
Use Case Name:	Delete tags from a post
Description:	User can delete tags from a post

Actors:	Users
Pre-conditions:	Users must login, target entry must exist
Post-conditions:	The tag has been removed from the entry

Frequency of Use:	Daily or hourly by users		
Flow of Events:		Actors:	System Response:
	1	Navigate to the post	
	2	Click in the tags field	Displays cursor in tag field
	3	Backspace or delete to remove tag	Tag is removed from post
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-014
Use Case Name:	Admin changes User access level
Description:	An Admin can change the access level of any user

Actors:	Admin		
Pre-conditions:	Admin must be logged in. Target User account must already exist		
Post-conditions:	User account access level has been updated		
Frequency of Use:	Daily or hourly by users		
Flow of Events:		Actors:	System Response:
	1	Open settings menu and click "Admin Panel" button	Displays Admin dashboard
	2	Select User tab	Display User section
	3	Search for User by name in search bar	Displays User
	4	Click "Account access" section from Users menu	Display Account Access section
	5	Select access level drop-down and choose desired level	



	6	Click “save” button	User access level is updated
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-015
Use Case Name:	Edit profile
Description:	User can edit their profile and choose a nickname

Actors:	User		
Pre-conditions:	User is login, being on the landing page		
Post-conditions:	User's profile is updated		
Frequency of Use:	Daily by user		
Flow of Events:		Actors:	System Response:
	1	Click “profile” section	Redirecting the user to the page displaying user's personal profile.
	2	Click “Edit” button.	Popup a window with input fields of each item allowing user to edit
	3	User can change what they want including the nickname area in input fields	
	4.	User click ‘submit’ button	Update the data in database
Variations:	User can also click ‘cancel’ to decide not to make any change. The data will not change in database.		
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-016
Use Case Name:	Change password

Description:	User can change their password
--------------	--------------------------------

Actors:	User		
Pre-conditions:	User forgot password when trying to login		
Post-conditions:	User's password is reset		
Frequency of Use:	Daily by user		
Flow of Events:		Actors:	System Response:
	1	Click "reset" button	Sending an email to user's email address.
	2	Open the link	Open the page allowing user to enter new password
	3	Enter new password	Reset password in database
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-017
Use Case Name:	Comment
Description:	User can comment on their friend's posts

Actors:	User		
Pre-conditions:	User is login, user has friend(s), there exists some post to be commented		
Post-conditions:	A comment is added		
Frequency of Use:	Daily by user		
Flow of Events:		Actors:	System Response:
	1	Click 'comment' icon	Expand an input field
	2	Enter some content	
	3	Click 'send' button	Comment on post is saved

Variations:	User can also cancel to add a comment.
Notes and Issues:	
Developer Notes:	

Use Case ID:	UC-018
Use Case Name:	View Other User's Profile
Description:	User can view the profile of other users

Actors:	User		
Pre-conditions:	User is login, user has friend(s)		
Post-conditions:	User is redirected to the profile page of other user		
Frequency of Use:	Daily by user		
Flow of Events:		Actors:	System Response:
	1	Type a user name in search bar	List the user that be searched.
	2	Click on that user	Redirect to that user's public profile
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-019
Use Case Name:	User can delete a comment they have left on a post
Description:	User can delete a comment under a post if he / she wants

Actors:	User
Pre-conditions:	User must login, and have comment under specific post

Post-conditions:	The specific comment is removed from the post		
Frequency of Use:	Daily by user		
Flow of Events:		Actors:	System Response:
	1	Click Personal main page	Display the main page
	2	Find the specific post, and click comment button	Shows all the comment under the post
	3	Click “delete” icon on the top right of the specific comment	Remove the comment
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-020
Use Case Name:	User can remove a friend
Description:	User can edit to remove a friend if he / she wants to remove

Actors:	User		
Pre-conditions:	User has signed up and has friends		
Post-conditions:	The users are no longer shown as friends		
Frequency of Use:	Daily by user		
Flow of Events:		Actors:	System Response:
	1	Click “Friends” button in the main navigation bar	Displays the “Friends” page
	2	Click “delete” icon on the top right of the specific friend’s synopsis item	Page show up a prompt “Are you sure to delete this friend? Yes / Cancel”
	3	Click “Yes” button	The friendship relationship is removed from both users’ “Friends” page and they no longer see each others’ posts

Variations:	User can choose Yes or Cancel to decide if is sure enough to delete a friend
Notes and Issues:	
Developer Notes:	

Use Case ID:	UC-021
Use Case Name:	User login
Description:	User can login

Actors:	User		
Pre-conditions:	User is not logged in but is at "login" page		
Post-conditions:	User is logged in		
Frequency of Use:	Daily by user		
Flow of Events:		Actors:	System Response:
	1	Input correct username and password.	
	2	Click "login"	Verify username and password
	3	User is online	Display the landing page with user profile and posts after logging in
Variations:	If the password is wrong, popup an alert and reject the user.		
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-022
Use Case Name:	User logout
Description:	User can logout

Actors:	User
---------	------

Pre-conditions:	User is logged in		
Post-conditions:	User is logged out. User profile and non-public posts are hidden.		
Frequency of Use:	Daily by user		
Flow of Events:		Actors:	System Response:
	1	Click “logout” button	Redirect the landing page without user profile and posts after logging out
	2	User is logged out	
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-023
Use Case Name:	Admin can view system usage statistics
Description:	An Admin can check the system usage statistics if needed

Actors:	Admin		
Pre-conditions:	Admin must be logged in to Admin Console		
Post-conditions:	Admin can see the usage statistic page		
Frequency of Use:	Daily and may be more frequently when in special cases.		
Flow of Events:		Actors:	System Response:
	1	Select “Stats” section on menu bar	Display Usage Statistics page
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-024
--------------	--------

Use Case Name:	User requests account deletion
Description:	A User can request that their account be deleted

Actors:	User		
Pre-conditions:	User must be logged in.		
Post-conditions:	User account deletion request has been queued		
Frequency of Use:	Daily or less by users		
Flow of Events:		Actors:	System Response:
	1	Click Settings panel and select "settings"	Displays Settings pane
	2	Select "Request Account Deletion" button	Display confirmation prompt
	3	Confirm	Adds user to deletion queue
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-025
Use Case Name:	Admin confirms user deletion request
Description:	Admin is notified of new user deletion request and deletes user

Actors:	Admin		
Pre-conditions:	Admin is viewing the Admin console. A User has requested deletion		
Post-conditions:	User account has been deleted		
Frequency of Use:	Daily or less by Admins		
Flow of Events:		Actors:	System Response:
	1	Click "notifications" icon	Displays Notifications

	2	Click “User Deletion Requests” notification	Display User console
	3	Click “Delete User” button next to user 1 in queue	Display confirmation prompt
	4	Click “yes”	Deletes User account
Variations:	2B Admin may go straight to User console and see deletion queue		
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-026
Use Case Name:	User can sign up
Description:	User can create a new account

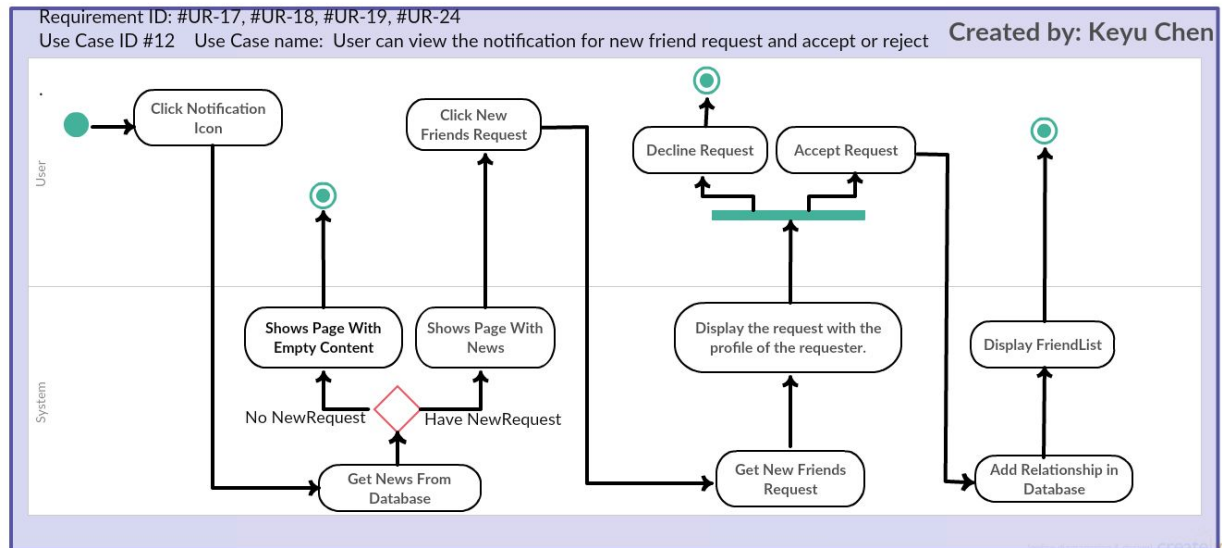
Actors:	Admin		
Pre-conditions:	The user does not already have an account. The user is on the main page.		
Post-conditions:	User's account is created		
Frequency of Use:	Usually once for each user		
Flow of Events:		Actors:	System Response:
	1	Click “Sign Up” icon in navigation bar	Displays Display sign up page
	2	Type in First Name, Last Name, Email, re-enter Email, password, re-enter password and birthday.	Check if the email is valid and if the password and the re-enter password matched and if Email and re-enter Email matched
	3	Choose gender	
	4	Click “Sign Up” button	redirect to the Login page
Variations:	Email may not be valid, Email and re-enter email may not match, password and re-enter password may not match.		
Notes and Issues:			
Developer Notes:			



Use Case ID:	UC-027
Use Case Name:	Admin can view Admin Console
Description:	Admin can view Admin Console

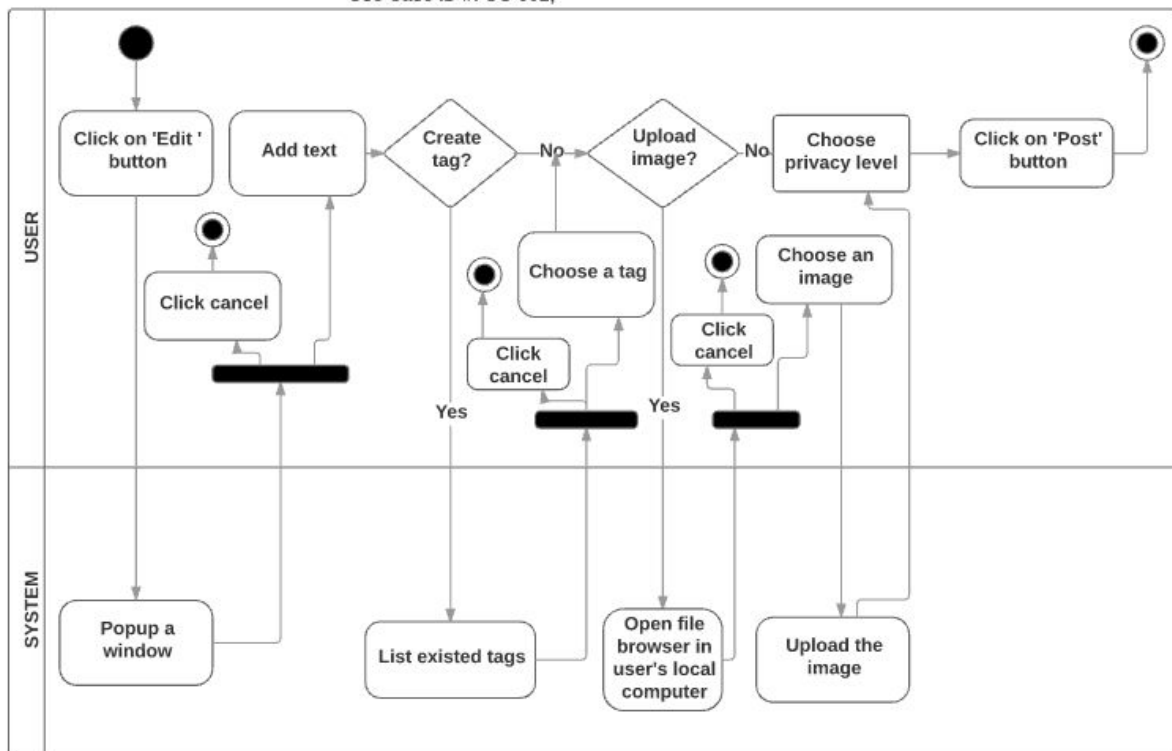
Actors:	Admin		
Pre-conditions:	User has Admin role. Admin must be logged in.		
Post-conditions:	Admin can see the Admin Console		
Frequency of Use:	Daily and may be more frequently when in special cases.		
Flow of Events:		Actors:	System Response:
	1	Select "settings" menu button	Display settings popout
	2	Click "view admin console"	Redirect to Admin Console
Variations:			
Notes and Issues:			
Developer Notes:			

## Activity Diagram

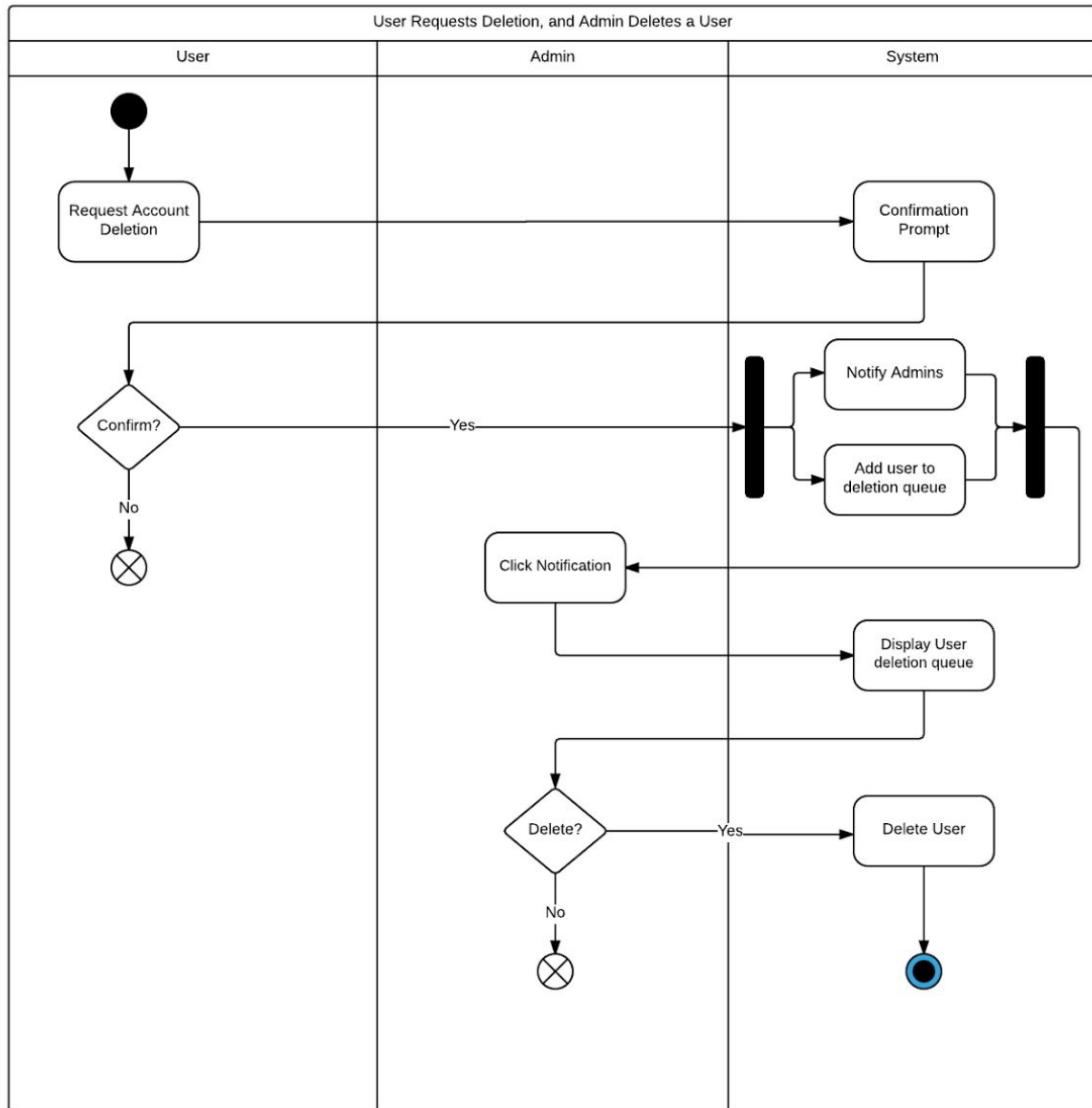


YUE ZHANG

Use Case: Create a post[User can create a new post by uploading a picture and tagging the entry ];  
Requirement ID #: UR-05,UR-06,FR-14 ;  
Use Case ID #: UC-001;



UR-21, UR-25  
UC-024, UC-025  
FR-22, FR-23  
Jeannette Pepin

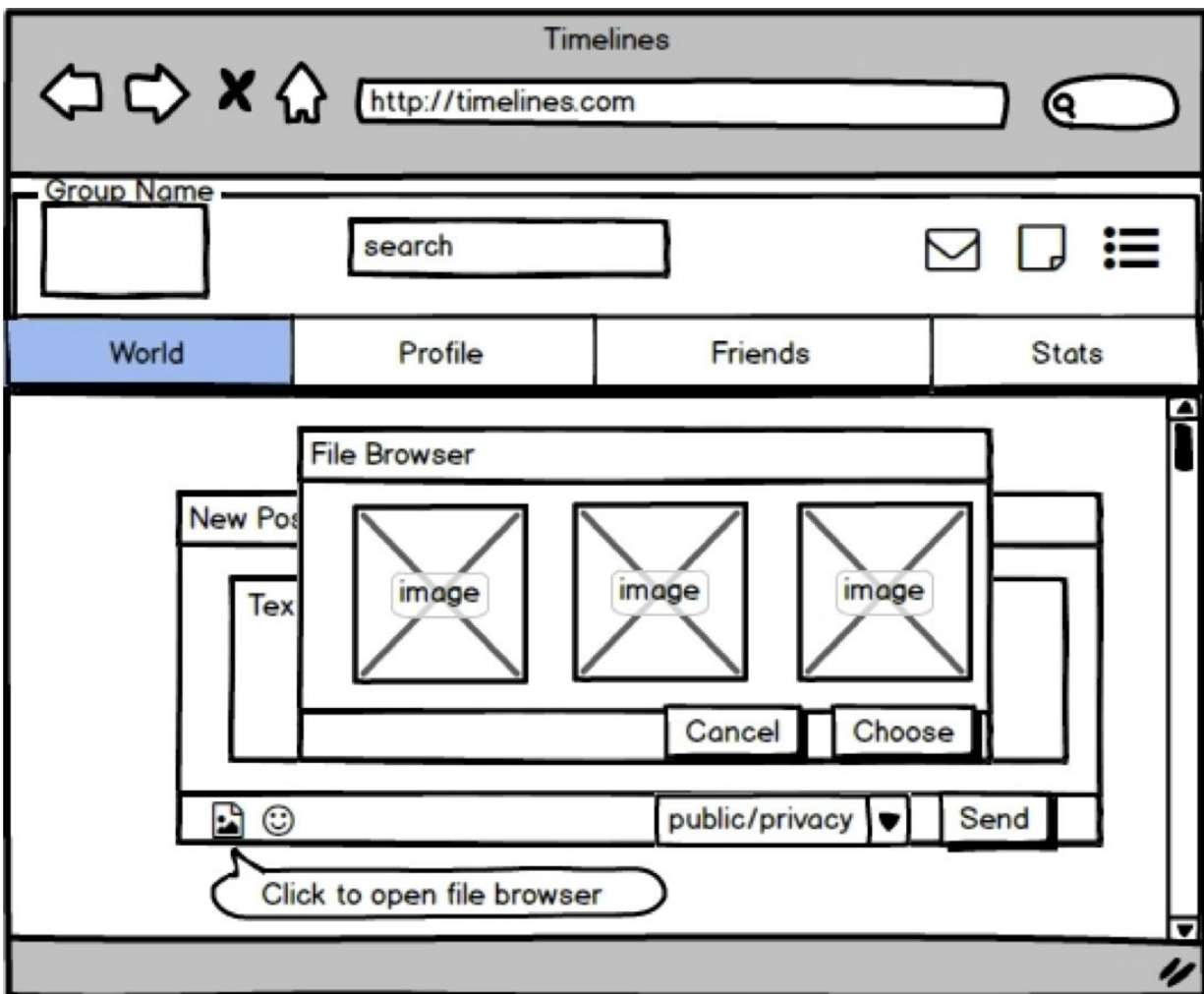


## Data Storage

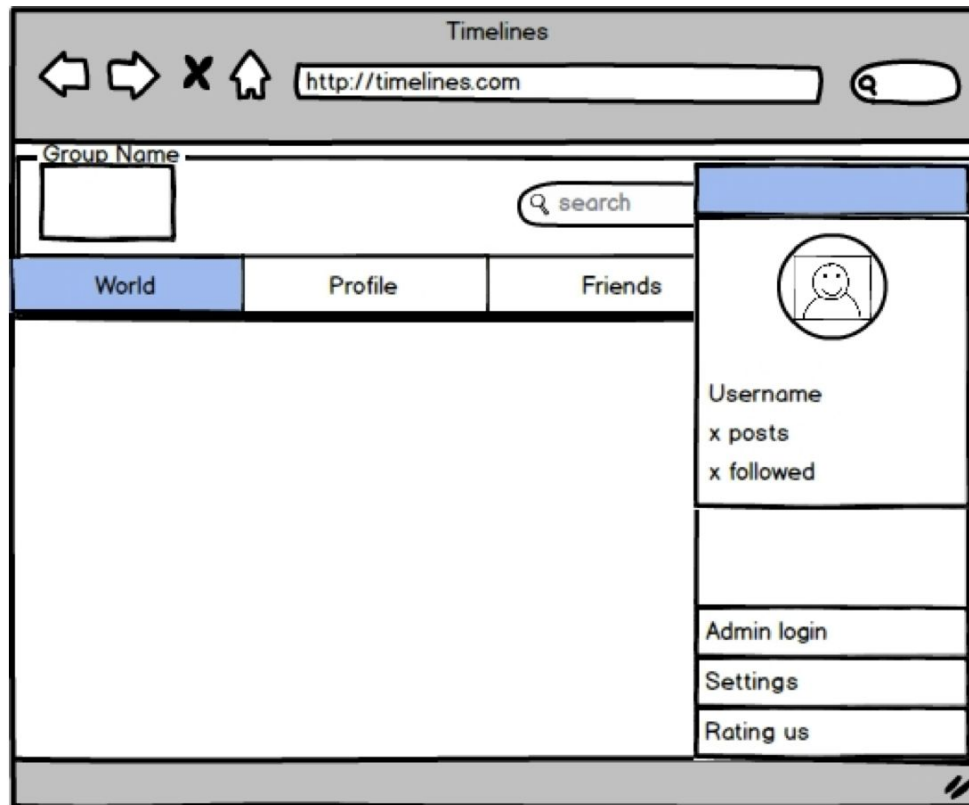
- Data Storage: MySQL
- Classes:
  - User to store information about users (admins are users)
  - Post to store information about posts
  - Comments to store comments on posts
  - Tags to store tags on posts
  - Likes to store the users that liked a given post
  - NotifyComment to store comment notifications and handle notifications
  - NotifyFriend to store friend request notifications and handle notifications
  - NotifyLikes to store Like notifications and handle notifications
  - NotifyDeletion to store user deletion requests and handle notifications to admins
  - Friends to store friendship relationships between users
  - Profile to store user bio information and userpic
  - UserDeletion to store when user accounts were flagged for deletion, and when they were deleted

## UI Mockups

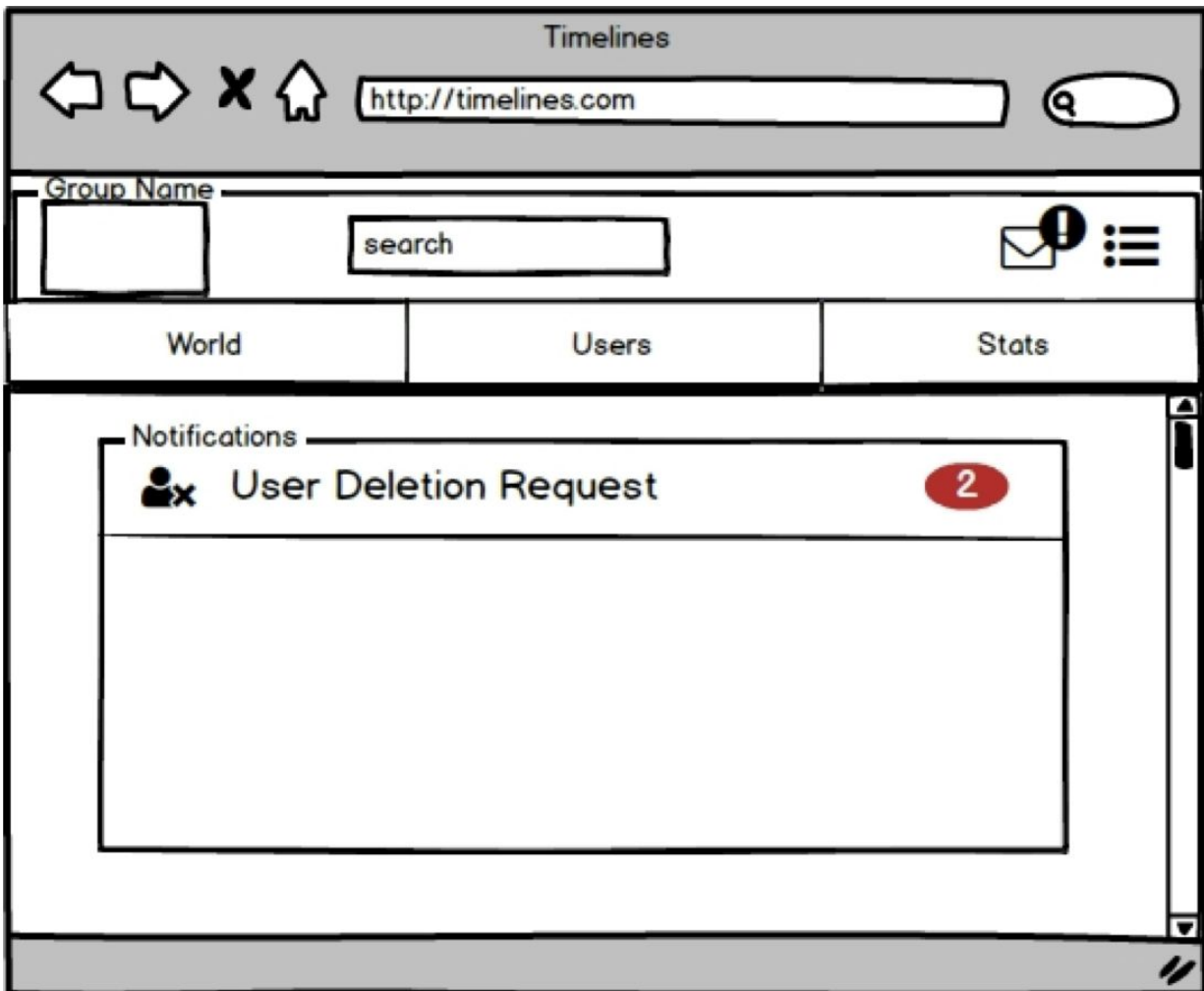
See Timelines.pdf for full set



[\[back to top\]](#)



Side-nav bar  
popup when  
menu icon  
clicked



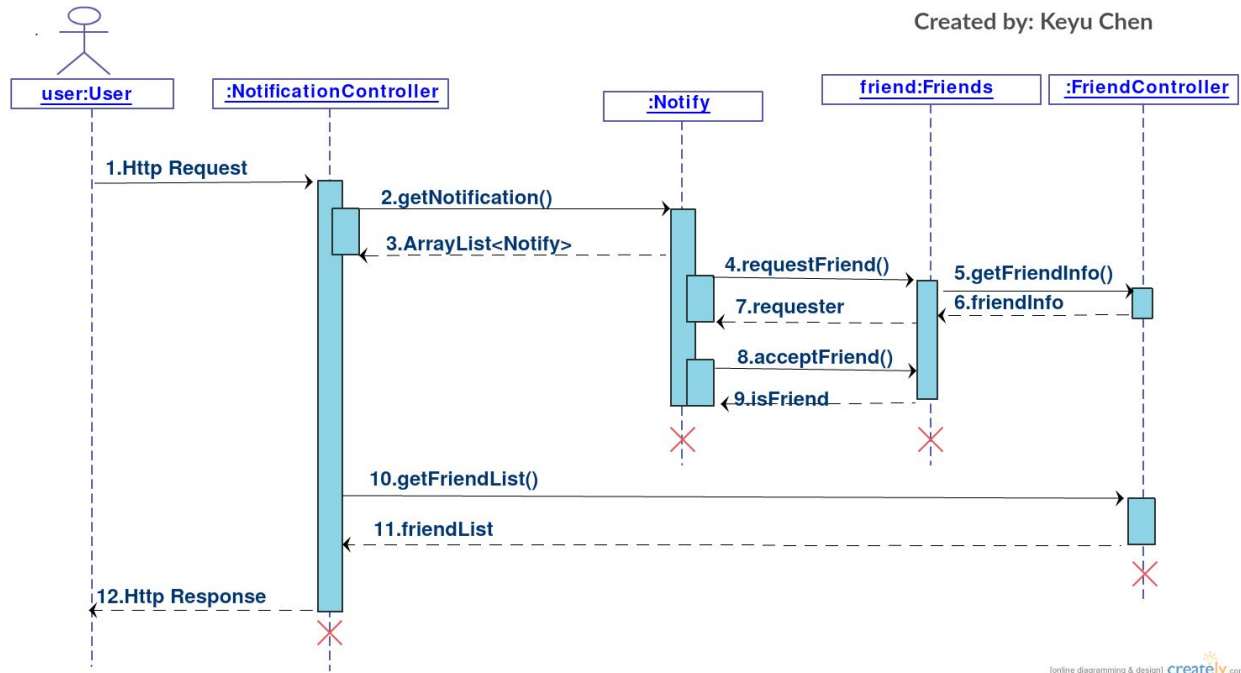
## User Interaction

### UC-12(Condition: acceptFriend() )

Requirement ID: #UR-17, #UR-18, #UR-24

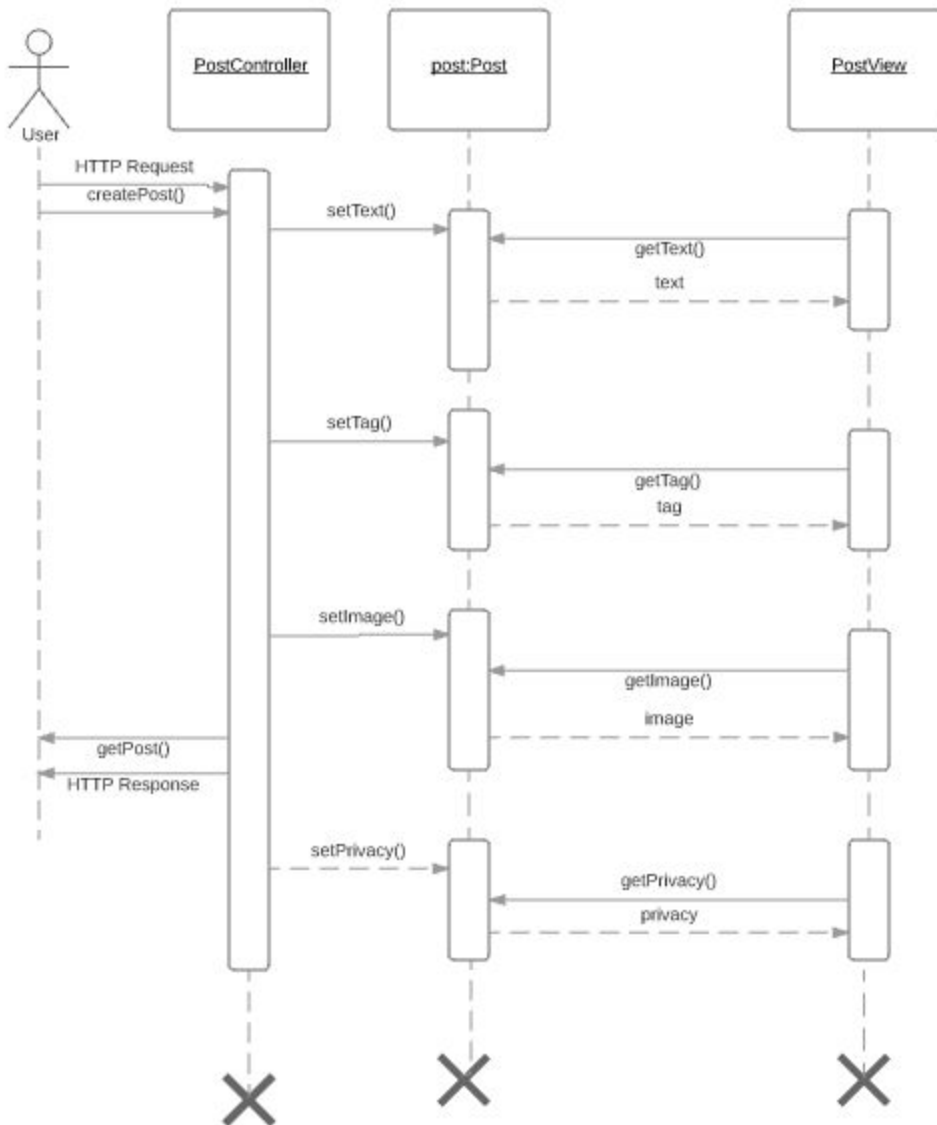
Use Case ID #12 Use Case name: User can view the notification for new friend request and accept or reject

Created by: Keyu Chen



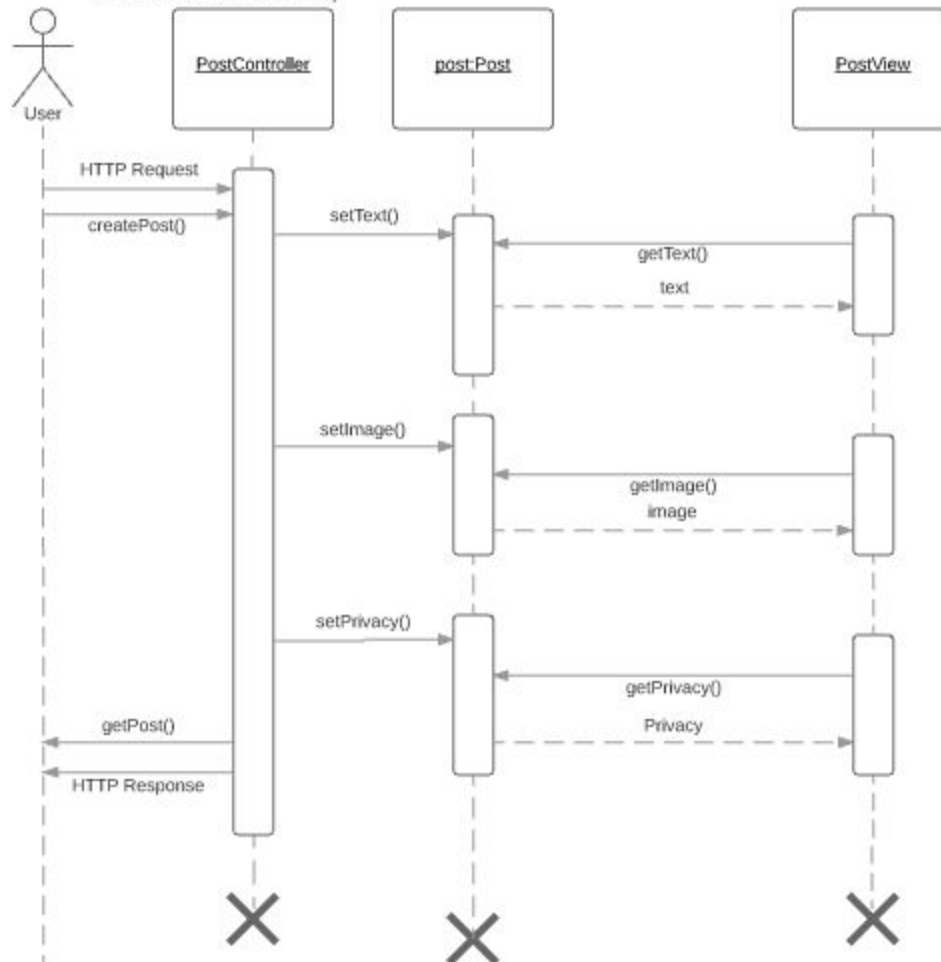


Use Case: Create a post;  
Description: A post with tag, text, image  
Requirement ID #: UR-05,UR-06,FR-14 ;  
Use Case ID #: UC-001;



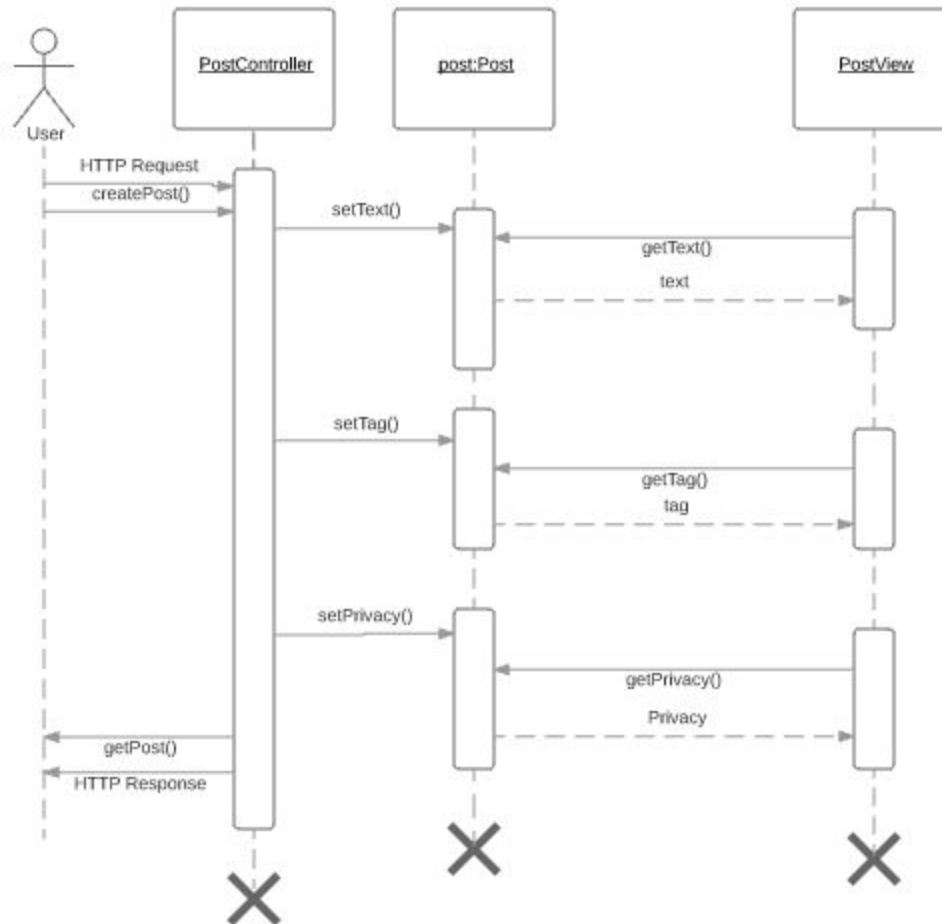
YUE ZHANG

Use Case: Create a post;  
Description: A post with image, text  
Requirement ID #: UR-05,UR-06,FR-14 ;  
Use Case ID #: UC-001;



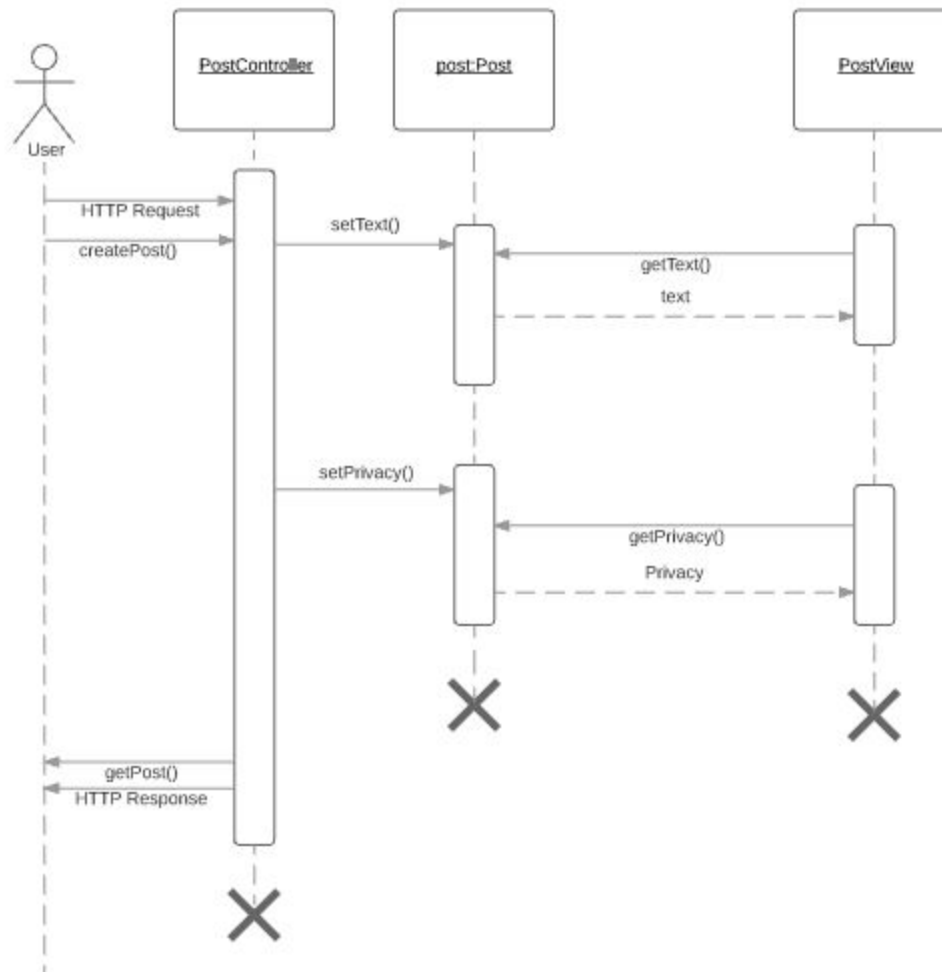
YUE ZHANG

Use Case: Create a post;  
Description: A post with tag, text  
Requirement ID #: UR-05,UR-06,FR-14 ;  
Use Case ID #: UC-001;



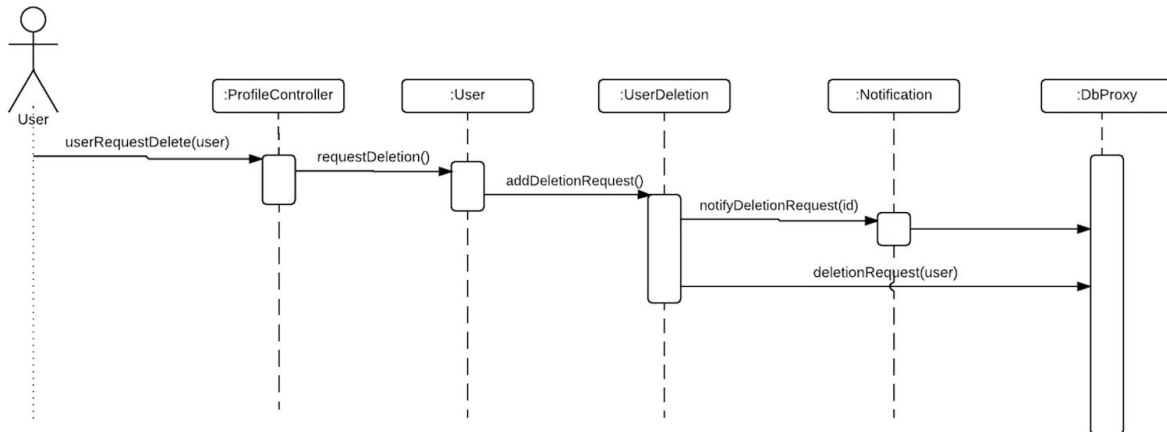
## YUE ZHANG

Use Case: Create a post;  
Description: A post with plain text  
Requirement ID #: UR-05,UR-06,FR-14 ;  
Use Case ID #: UC-001;

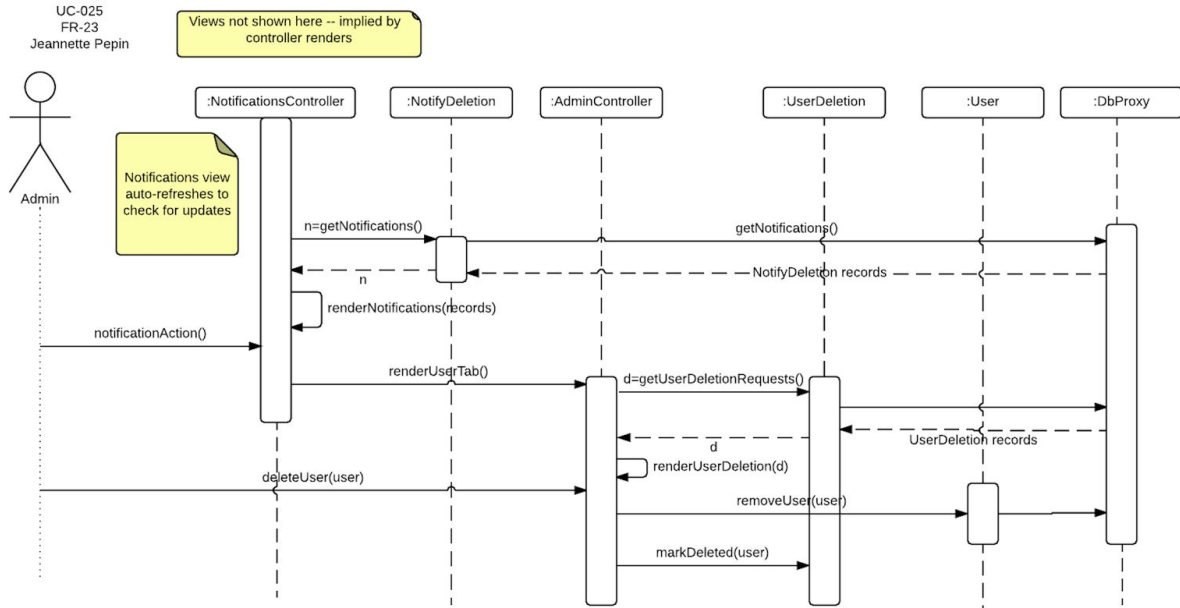


## Jeannette Pepin

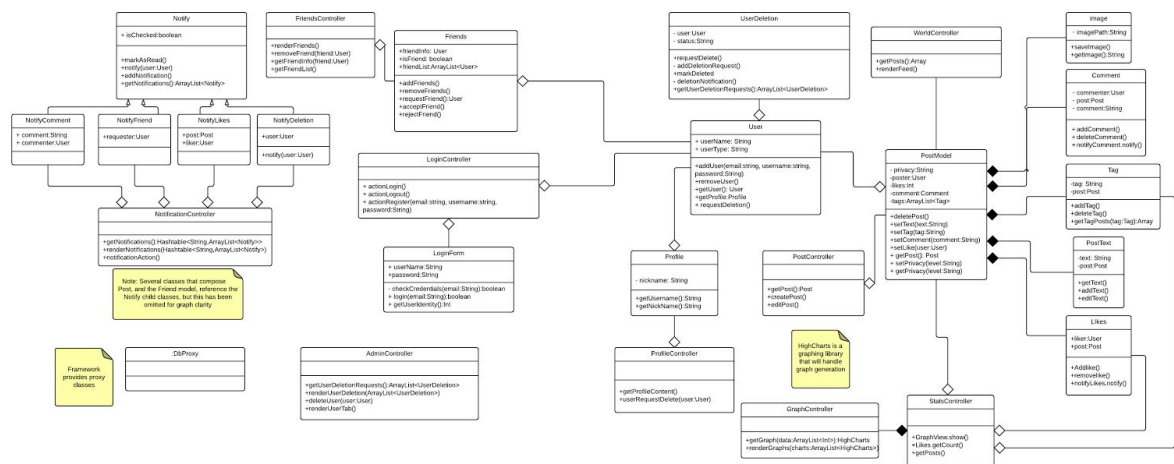
UR-021  
UC-024  
Jeannette Pepin



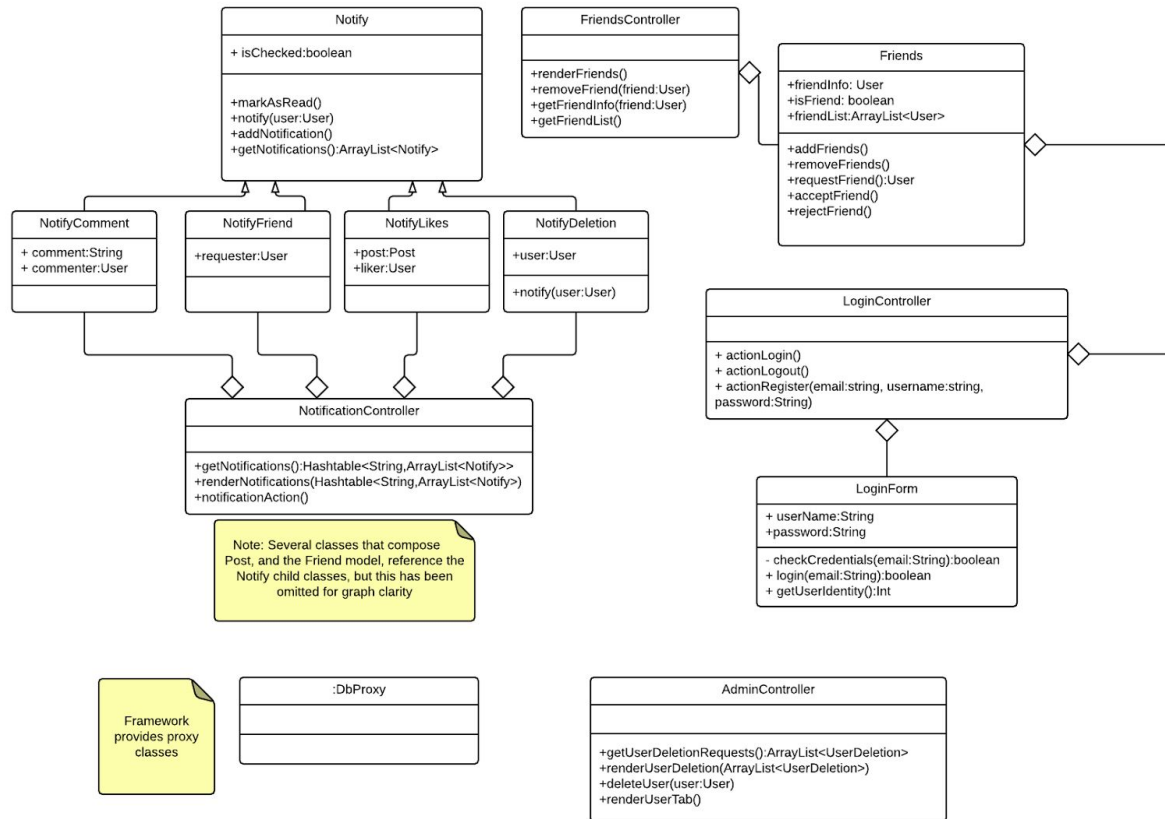
UR-025  
UC-025  
FR-23  
Jeannette Pepin



## Class Diagram



### Closeup One



Closeup Two

