



THIS FORM IS FOR BOTH THE GENERAL & MEDICAL INFORMATICS PROGRAMMES

SE - I COURSE PROJECT (PHASE 1 COVER SHEET)

Discussions Scheduled for Week 8 (*more details will be announced later*).

- Print 1 copy of this cover sheet and attach it to a printed copy of the documentation (SRS, ... etc.). You must also submit softcopies of all your documents (*as PDFs*); details will be announced later.
- Please write all your names in Arabic.
- Please make sure that your students' IDs are correct.
- Handwritten Signatures for the attendance of all team members should be filled in before the discussion.
- Please attend the discussion on time (*announced separately*), late teams will lose 5 grades.

Project Name:(27) Photo and Video Sharing Social-Networking Application [inspired by Instagram]

Team Information (*typed not handwritten, except for the attendance signature*):

	ID [Ordered by ID]	Full Name [In Arabic]	Attendance [Handwritten Signature]	Final Grade
1	20210026	احمد السيد السيد محمود	احمد السيد	
2	20210082	احمد علي سالماني احمد	احمد علي	
3	20210085	احمد عمر حسين محمد	احمد عمر	
4	20210102	احمد محمد عبدالسلام محمد	احمد محمد	
5	20210108	احمد محمد متولي بيومي	احمد محمد	
6	20210115	احمد محمود رمضان محمد	احمد محمود	
7				

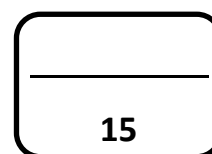
Grading Criteria:

10 Items		Grade	Notes
1. Functional Requirements	1		
2. Non-Functional Requirements	1		
3. Use-Case Diagram(s) including general use-cases for the system, and the detailed use-cases description	2		



4. System Architecture – <i>including applied Architectural Pattern(s)</i>	1		
5. Activity Diagrams	2		
6. Database Specification (<i>ERD, Tables</i>)	2		
7. Class Diagram (<i>Interfaces, Classes, Relations</i>) – An initial version based on the requirements and Use-Case/Activity diagrams.	2		
8. Object Diagrams (<i>Including object diagrams that illustrate the preconditions and the post-conditions of selected functions</i>)	1		
9. Package Diagram(s)	1		
10. Sequence Diagrams <i>including System Sequence Diagrams (SSDs)</i>	2		

Teaching-Assistant's Signature: _____



Software Engineering 1

Phase 1

Team Members:

MEMBER ID	MEMBER NAME
20210108	أحمد محمد متولي بيومي
20210102	أحمد محمد عبدالسلام
20210085	أحمد عمر حسين
20210026	أحمد السيد السيد
20210115	أحمد محمود رمضان
20210082	أحمد علي سالماني

Functional Requirements

- **User Sign Up**
 1. **Goal:** Register new user to the website.
 2. **Pre Conditions :** 1. Internet Connection 2. User accessed the website.
 3. **Post Conditions:** User Data is saved to the database.
- **User Log In**
 1. **Goal:** Give the user the ability to access the website and all it's content.
 2. **Pre Conditions:** 1. User must be signed up to the website 2. Input correct information 3. Internet Connection.
 3. **Post Conditions:** User can access their profile and the website.
- **User Log Out**
 1. **Goal:** Give the user the ability to log out to ensure the safety of their information and privacy
 2. **Pre Conditions:** : 1. User must have a pre-existing account. 2. User must be logged in 3. User must log out.
 3. **Post Conditions:** The user is redirected to the log in/sign up page.
- **Reset Password**
 1. **Goal:** Give the user the ability to reset their password incase it was lost.
 2. **Pre Conditions:** 1. User must be signed up to the website and have a pre-existing account.
 3. **Post Conditions:** User is emailed a link that gives them the ability to reset their password into a new one of their choice.
- **Editing Profile**
 1. **Goal:** Give the user the ability to edit their profile to add essential information such as DOB, Name, Gender and non-essential information like the biography.
 2. **Pre Conditions:** 1. User must be signed up to the website 2. User must be logged in using the correct information stored in database.
 3. **Post Conditions:** User can edit their profile and add the information they want.

- **Follow and Get Followed**

1. **Goal:** Give the user the ability to follow other people in order to see their posts on his news feed and also get followed by other users who might be interested in their posts.
2. **Pre Conditions:** 1. User must have a pre-existing account. 2. User must be logged in.
3. **Post Conditions:** The followers and/or following count goes up and the followed profiles shall now pop up on the user's feed.

- **Unfollow users**

1. **Goal:** Give the user the ability to remove other users from the followers/following list if they are no longer interested in their content.
2. **Pre Conditions:** 1. User must have a pre-existing account. 2. User must be logged in 3. User must follow/be followed by the user they are attempting to remove.
3. **Post Conditions:** The specified user is removed from the list.

- **Upload Post(s)**

1. **Goal:** Give the user the ability to share media with other users.
2. **Pre Conditions:** 1. User must have a pre-existing account. 2. User must be logged in 3. User must specify media files to be attached to the posts 4. Users must specify the status of the post (private/public).
3. **Post Conditions:** Post is uploaded with the option to add a caption, hashtags and/or a location tag.

- **Upload Story(ies)**

1. **Goal:** Give the user the ability to share media with other users for a certain amount of time.
2. **Pre Conditions:** 1. User must have a pre-existing account. 2. User must be logged in 3. Users must specify which users they wish to share the story with 4. Users must attach a media file to the story
3. **Post Conditions:** The story is uploaded and is to be deleted by default after 24 hours.

- **Edit Post(s)**

1. **Goal:** Give the user the ability to edit the posts they have uploaded.
2. **Pre Conditions:** 1. User must have a pre-existing account. 2. User must be logged in 3. User must have a post uploaded 4. User must save the post after editing.
3. **Post Conditions:** The post is edited with the new information the user added to it or removed from it.

- **Edit Story(ies)**
 1. **Goal:** Give the user the ability to edit the stories they have uploaded
 2. **Pre Conditions:** : 1. User must have a pre-existing account. 2. User must be logged in 3. User must have a story uploaded 4. User must save changes.
 3. **Post Conditions:** The story is edited with the information user added to it or removed from it.

- **Post Comment(s)**
 1. **Goal:** Give the user the ability to post comments on other people's posts.
 2. **Pre Conditions:** 1. User must have a pre-existing account. 2. User must be logged in 3. User must submit the comment after typing it down.
 3. **Post Conditions:** The comment pops up under the post specified including the user's name and the comment they posted.

- **Edit Comment(s)**
 1. **Goal:** Give the user the ability to edit comments they posted.
 2. **Pre Conditions:** 1. User must have a pre-existing account. 2. User must be logged in 3. User must have a comment posted. 4. User must specify the comment they want to edit and save changes after.
 3. **Post Conditions:** The changes made to the comment is visible to everyone viewing the post.

- **Delete Post(s)**
 1. **Goal:** Give the user the ability to delete posts they have uploaded.
 2. **Pre Conditions:** 1. User must have a pre-existing account. 2. User must be logged in 3. User must have a post uploaded. 4. User must specify the post they wish to delete.
 3. **Post conditions:** The post specified is no longer visible to the user or any other user.

- **Delete Story(ies)**
 1. **Goal:** Give the user the ability to delete stories they have uploaded.
 2. **Pre Conditions:** : 1. User must have a pre-existing account. 2. User must be logged in 3. User must have a story uploaded 4. User must specify the story they wish to delete
 3. **Post Conditions:** The story specified is no longer visible to the user or any other user.

- **Delete Comment(s)**
 1. **Goal:** Give the user the ability to delete comments they have posted.
 2. **Pre Conditions:** 1. User must have a pre-existing account. 2. User must be logged in 3. User must have a comment posted 4. User must specify the comment they wish to delete
 3. **Post Conditions:** The comment is no longer visible to the user or any other user.

- **Upload Multiple Media in one post**
 1. **Goal:** To allow the user to upload more than one picture in the same post to avoid making many posts.
 2. **Pre Conditions:** 1. User must have a pre-existing account. 2. User must be logged in 3. User must specify more than 1 media file. 4. User must post file
 3. **Post Conditions:** A normal post is uploaded with more than one media file attached to it.

- **Apply Filters to uploaded media**
 1. **Goal:** Give the user a wide variety of filters to apply on chosen media.
 2. **Pre Conditions:** : 1. User must have a pre-existing account. 2. User must be logged in 3. User must have atleast 1 media file uploaded 4. User must specify the filter they wish to apply
 3. **Post Conditions:** The specified filter is applied to the media file.

- **Count posts**
 1. **Goal:** Keep track of the number of posts uploaded by user.
 2. **Pre Conditions:** : 1. User must have a pre-existing account. 2. User must be logged in 3. User must have post(s) uploaded
 3. **Post Conditions:** The post counter on the user's profile is incremented by 1 upon each post upload.

- **Count Followers/Following**
 1. **Goal:** Keep track of the user's count of followers and following users.
 2. **Pre Conditions:** 1. User must have a pre-existing account. 2. User must be logged in 3. User must be following/followed by other users.
 3. **Post Conditions:** Followers and/or Following list may be incremented depends on the situation.

- **Send/Receive Messages**
 1. **Goal:** Allow another way of communication between users in a more private manner.
 2. **Pre Conditions:** : 1. User must have a pre-existing account. 2. User must be logged in 3. User must be following the user they wish to text 4. User must be followed by the user they wish to text
 3. **Post Conditions:** A message is sent from the sender user to the receiver user and vice versa.

- **Delete Messages**
 1. **Goal:** Give the user the ability to unsend messages sent to the receiver user.
 2. **Pre Conditions:** 1. User must have a pre-existing account. 2. User must be logged in 3. User must be following the user they wish to text 4. User must be followed by the user they wish to text. 5. User must specify the message they wish to delete.
 3. **Post Conditions:** The message is unsent and no longer visible to both users.

- **Receive Notifications**
 1. **Goal:** Allow user to keep track with the latest news on their feed.
 2. **Pre Conditions:** 1. User must have a pre-existing account. 2. User must be logged in 3. User must be mentioned in a post 4. User must receive a like on their post
 3. **Post Conditions:** A small pop up message appears notifying the user.

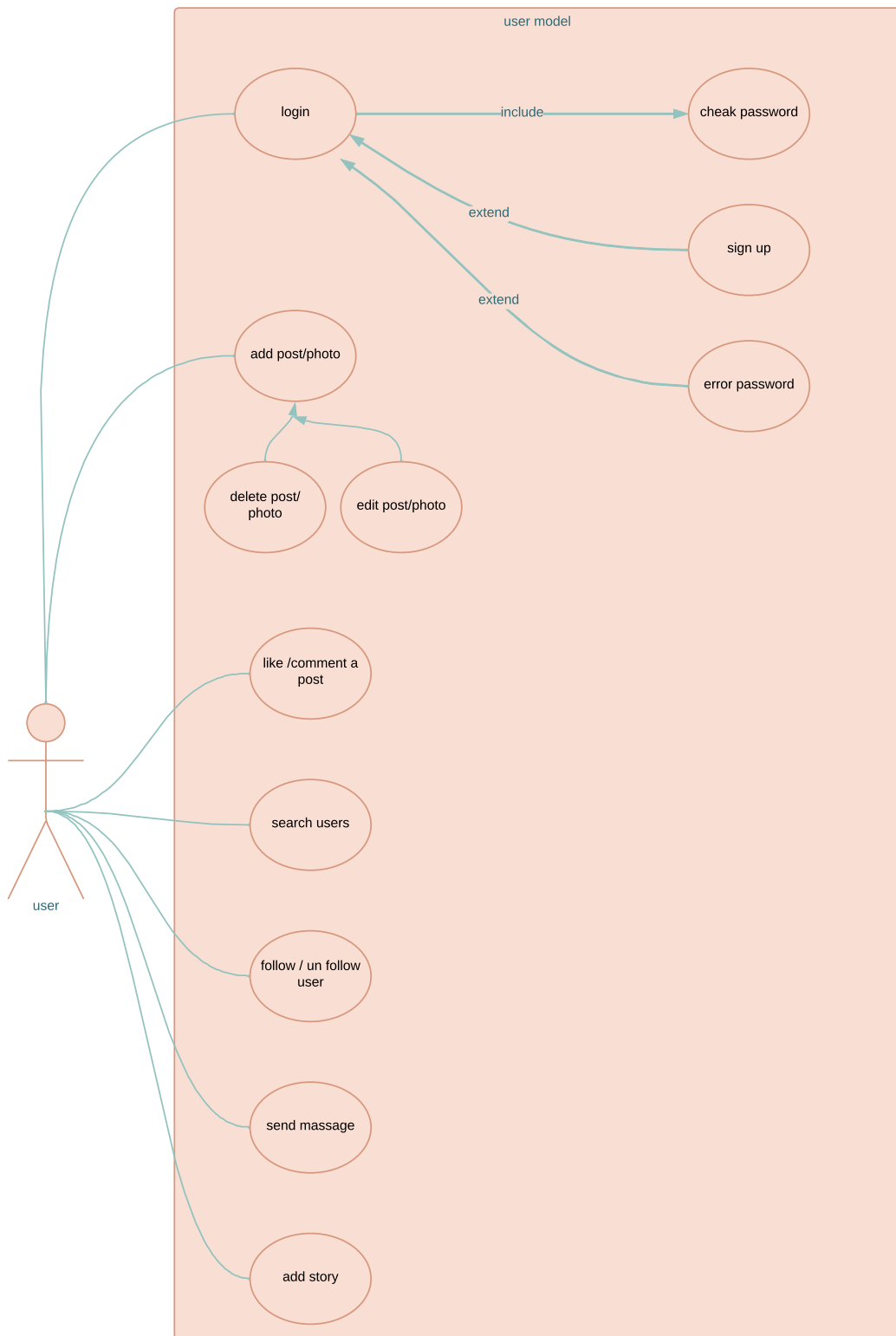
- **Information Encryption**
 1. **Goal:** Preserve the security and the privacy of the user by encrypting their information
 2. **Pre Conditions:** 1. User must have a pre-existing account in the database.
 3. **Post Conditions:** The user password is encrypted to avoid security issues.

- **Access News Feed**
 1. **Goal:** Give the user access to news feed to see the posts they uploaded and/or posts uploaded by other users.
 2. **Pre Conditions:** 1. User must have a pre-existing account. 2. User must be logged in 3. User must have uploaded posts 4. User may or may not follow/be followed by other users
 3. **Post Conditions:** Posts may appear over the news feed depending on the situation.
- **Access Stories**
 1. **Goal:** Give the user access to view stories uploaded by users that they follow or are followed by.
 2. **Pre Conditions:** 1. User must have a pre-existing account. 2. User must be logged in 3. User must be followed/following another user 4. The story must be set to public.
- **Search Posts by Hash/Location tags**
 1. **Goal:** Give the user the ability to look up posts depending on their interests and/or their location.
 2. **Pre Conditions:** 1. User must have a pre-existing account. 2. User must be logged in 3. User must access the search page 4. User must specify a valid hash/location tag.
 3. **Post Conditions:** The user should see posts depending on the hash/location tag they specified.

Non-Functional Requirements

1. High Availability
2. High Reliability
3. Ease of use
4. User-Friendly UI
5. High Accessibility.
6. Highly Scalable
7. Durability of stored data
8. Up To Date Security and Encryption Methods
9. A Latency of no more than 150 ms for generating news feed

Use Case Diagrams



User model description:

Function :	login
Description :	User login in application using name and password.
Success scenario:	user able to login into the system successfully
unsuccessful scenario:	the user cannot login because he is not registered to the system or the password is not correct

Function :	Add post / photo
Description :	User can add post or photo on application.
Success scenario:	User able to add post on application successfully.
unsuccessful scenario:	User cannot add post on application.

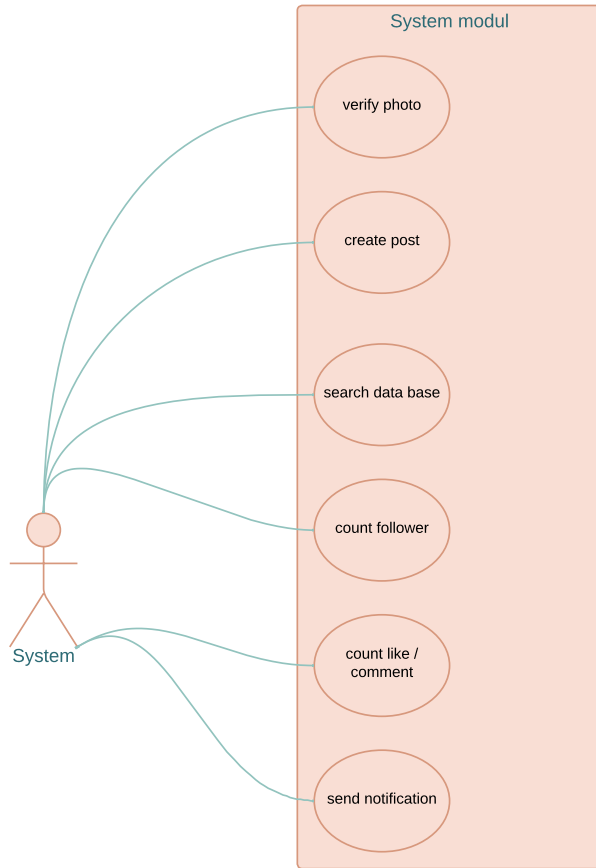
Function :	Like / comment a post
Description :	User can add like or comment on the any post he want.
Success scenario:	User able to add like or comment successfully.
unsuccessful scenario:	User cannot add like on the post.

Function :	Search users.
Description :	User search in her follower or a profile anther user.
Success scenario:	User able to search about any profile successfully
unsuccessful scenario:	User fails to search about any profile he want.

Function :	Follow / un follow
Description :	User can follow anther account.
Success scenario:	The user follow anther user successfully.
unsuccessful scenario:	The user fails to follow anther user.

Function:	Message.
Description :	User can send message to her friends or any account he want.
Success scenario:	User able to send message successfully.
unsuccessful scenario:	User fails to send message.

Function:	Add story.
Description :	User add post for a short time to express his status.
Success scenario:	User able to add story successfully.
unsuccessful scenario:	User fails to add story.



- **Function:** Verify Photo
- **Description:** Ensure the photo uploaded is not damaged/corrupted and that it is linked to the correct user through the user ID/Post ID/Story ID etc.
- **Successful Scenario:** The photo is validated and is assigned to the correct user
- **Unsuccessful Scenario:** The System fails to verify the picture due to invalid media ID/URL.

- **Function:** Create post.
- **Description:** Create a new post with multiple media files attached to it and optionally include Location tags/Hashtags, Captions and Comments.
- **Successful Scenario:** The post is successfully posted, linked to the correct user through the user ID and shows up on the user news feed and other users following/followed by the user.
- **Unsuccessful Scenario:** The post is not successfully posted due to being linked to an invalid user ID or a corrupt post ID or a corrupt media URL/ID.

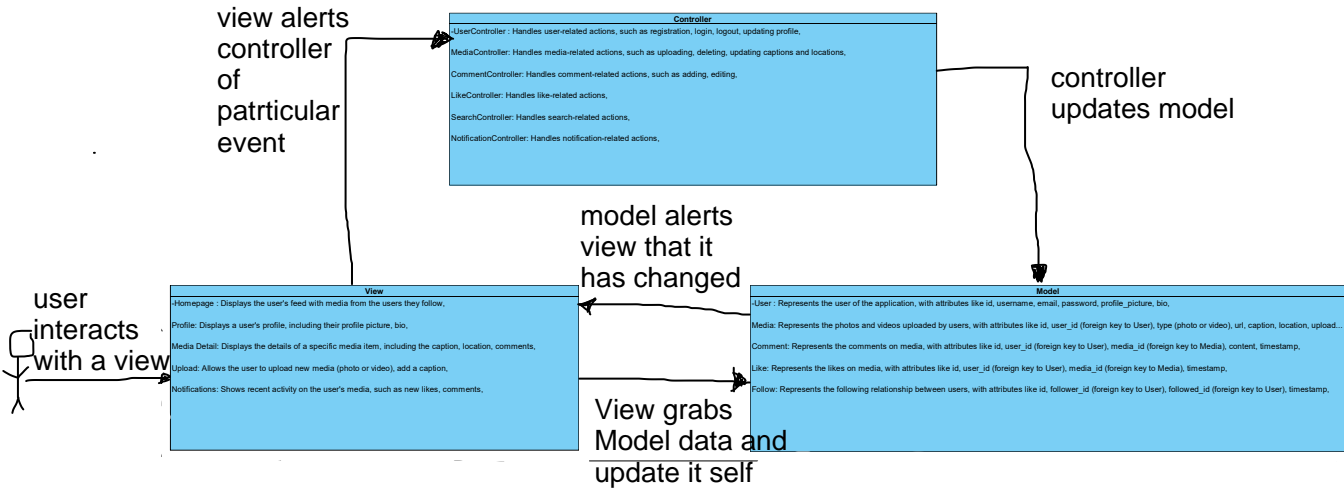
- **Function:** Search database.
- **Description:** Search the database for different types of information, User ID, Post ID, Story ID for example.
- **Successful Scenario:** The system successfully returns with the value it's intended to return.
- **Unsuccessful Scenario:** The system fails to return any value/returns a garbage value due to a conflict in the database or because of an invalid request.

- **Function:** Count Follower
- **Description:** Counts the number of followers of a user, increments the counter by 1 everytime the user follows a user.
- **Successful Scenario:** The counter is incremented by 1 upon following a user.
- **Unsuccessful Scenario:** The counter is incremented by 1 but does not refer to a valid user ID or the counter is not incremented at all despite following a valid user.

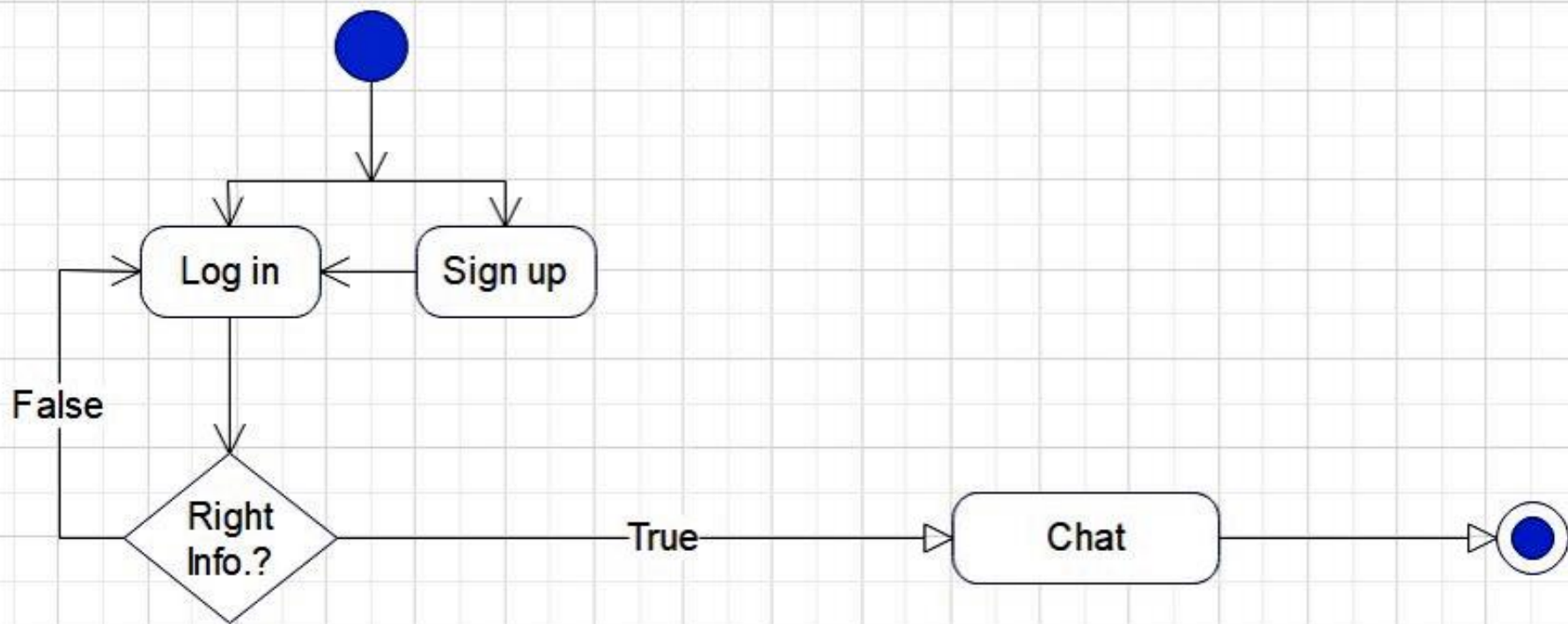
- **Function:** Count Like/Comment
- **Description:** Counts the number of likes and comments on a post posted by a user.
- **Successful Scenario:** The counter is incremented by 1 upon liking or commenting on a post.
- **Unsuccessful Scenario:** The counter is incremented by 1 but does not refer to a valid user ID or the counter is not incremented at all despite following a valid user.

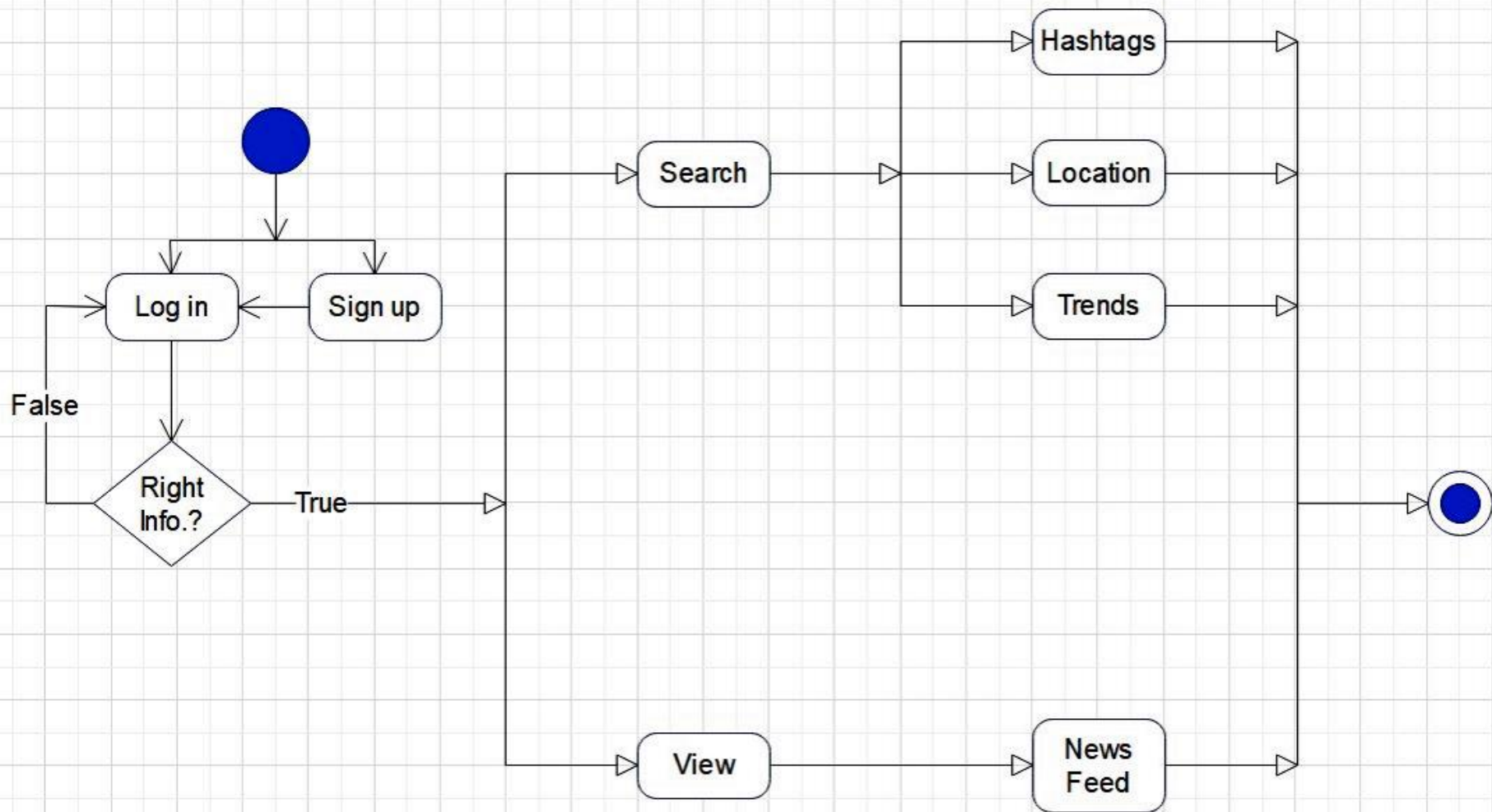
- **Function:** Send Notification
- **Description:** Sends a pop up message to a user to notify them of a certain event.
- **Successful Scenario:** The message successfully pops up the same moment an event happens and refers to a valid event.
- **Unsuccessful Scenario:** The message fails to pop up causing the user to miss an event or The message pops up referring to an invalid event.

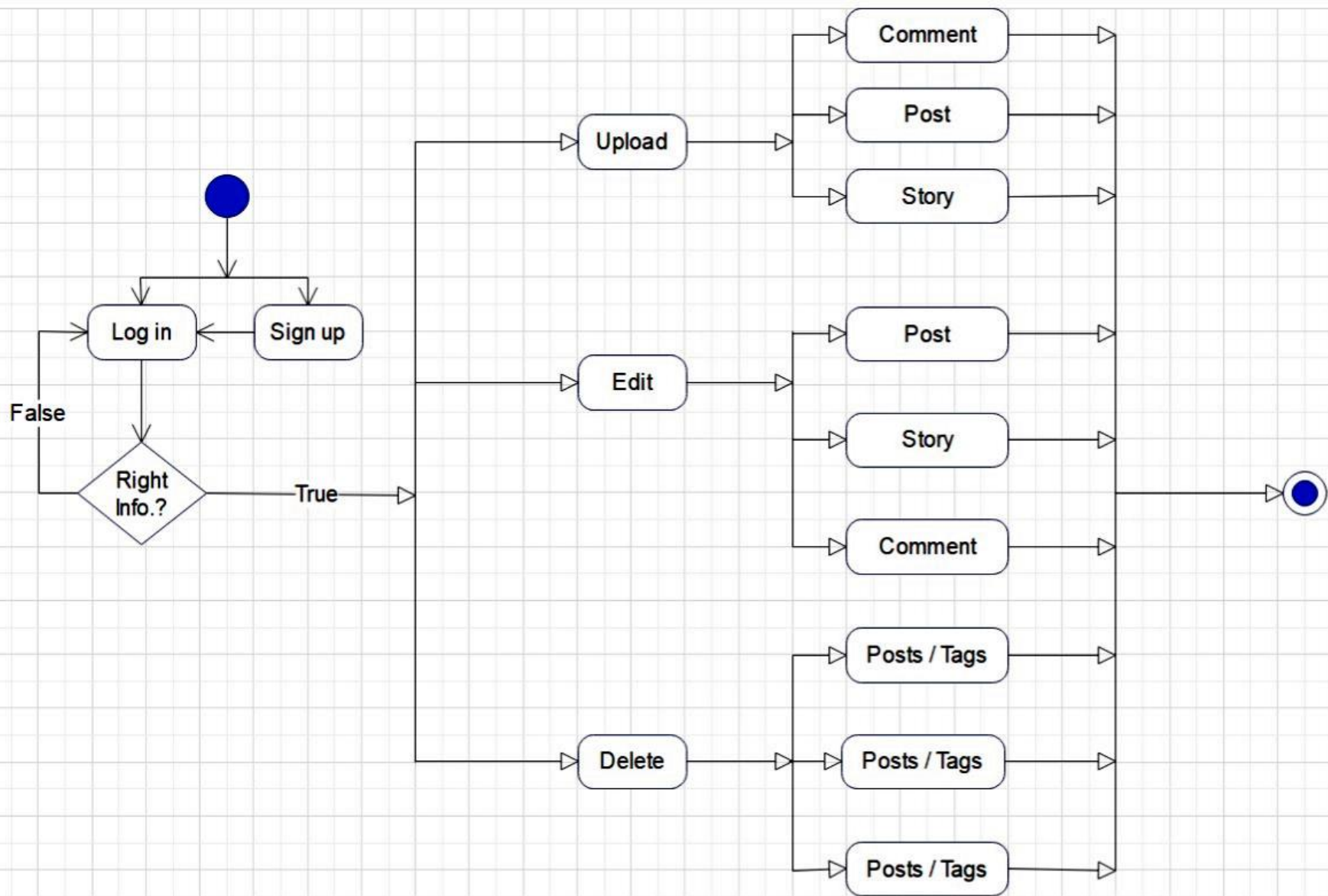
System Architecture



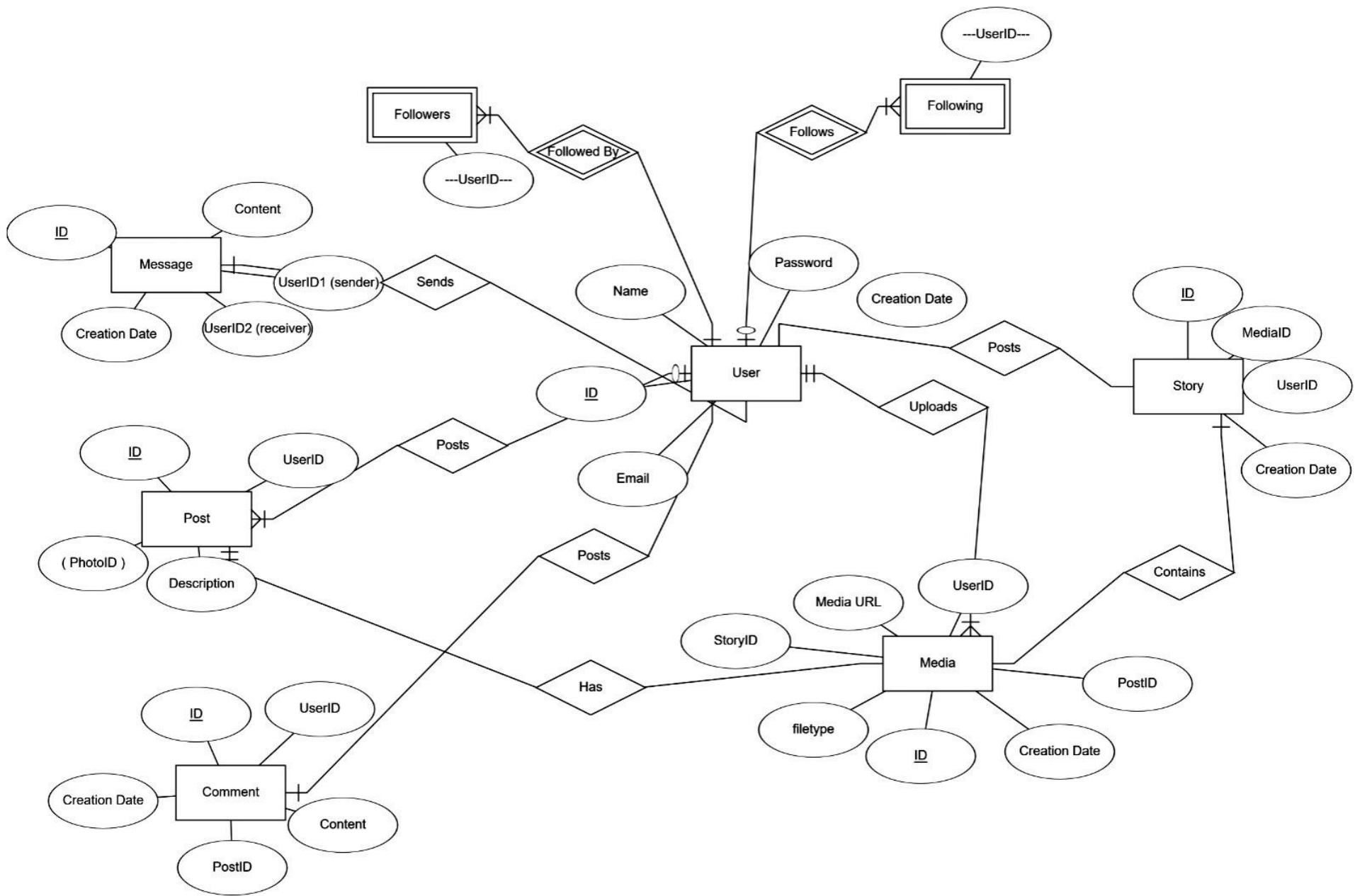
Activity Diagrams

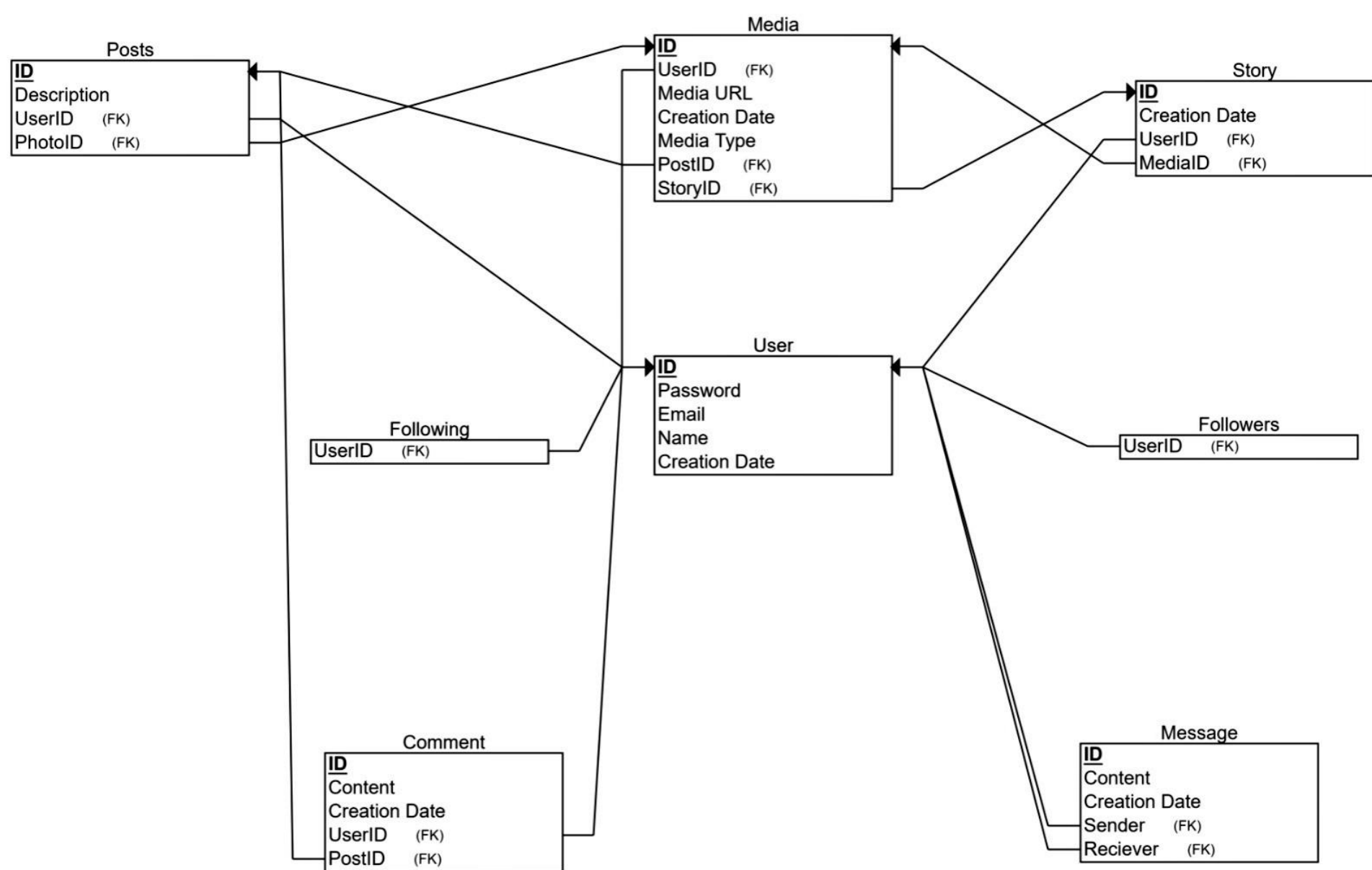




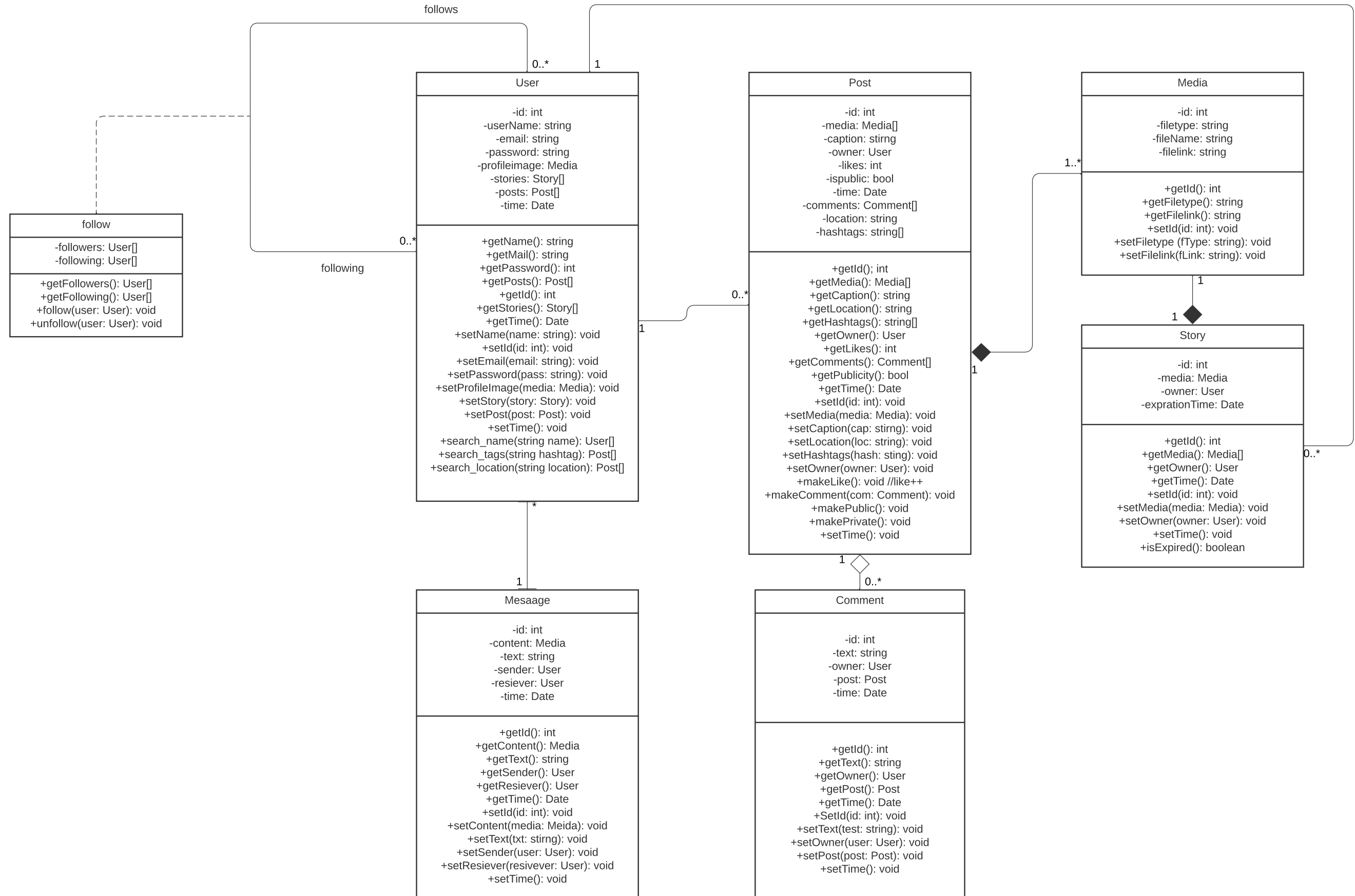


Database Specification



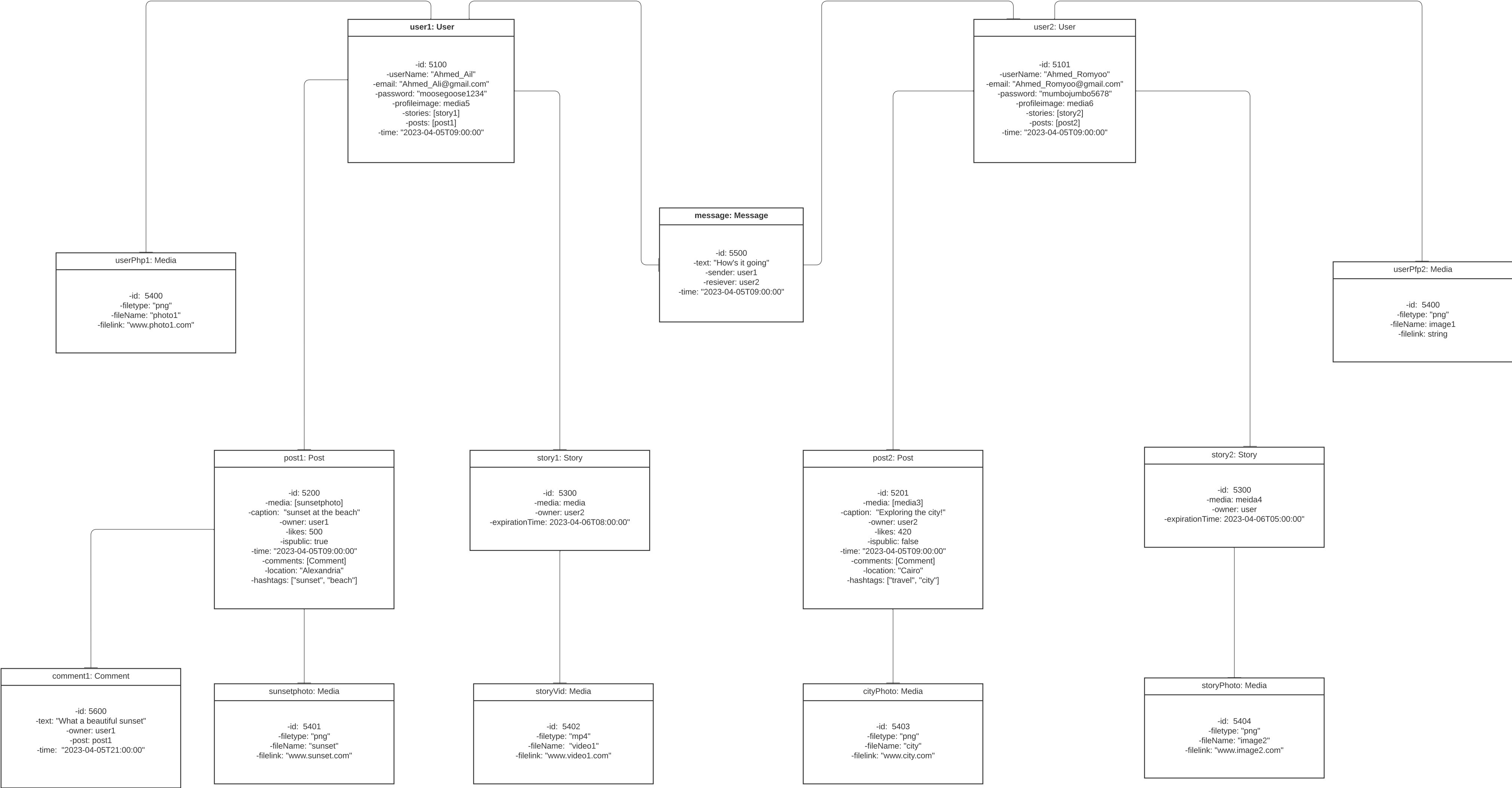


Class & Object Diagram

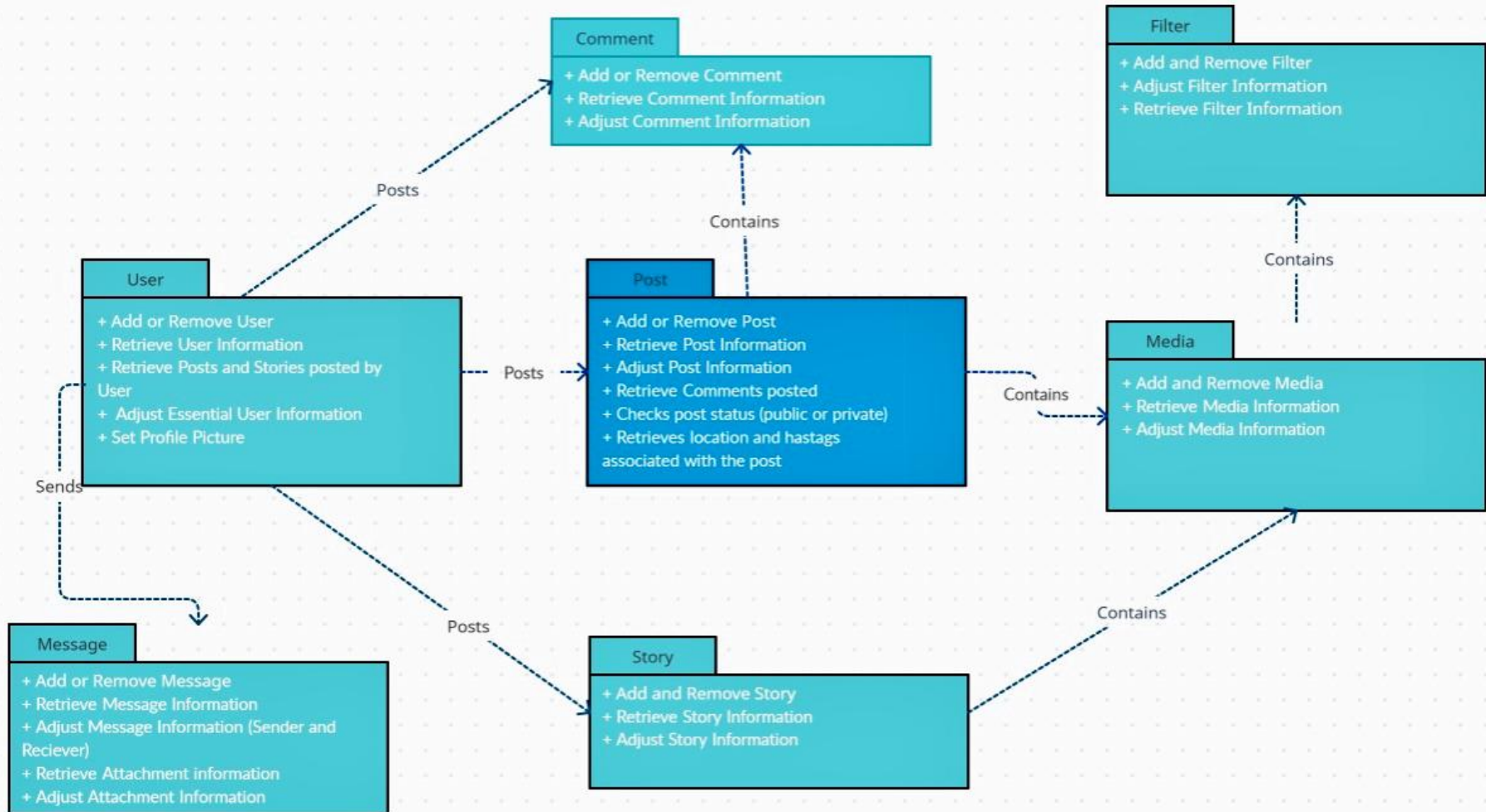


object diagram

A Person | April 8, 2023



Package Diagram



Sequence Diagrams

sd user diagram

