# Product requirements document -Instagram Like Button

A Product Requirements doc focuses on the users' idea of what the finished product should do.

| Instagram Like Button | |
|---|---|
| Product Manager | Ankur Bansal |
| Stakeholders | Product, Tech, Design, Marketing, users |
| Document Version | v1 |
| Created | 18th October, 2023 |

## ✏️ Problem Statement And Goals

**Problem**: Instagram is a platform where people share their pictures, videos etc. However, one challenge is that Instagram lacks a dedicated “like” button for expressing preferences or dislikes **without leaving comments** on pictures/videos. This absence of a simple feedback mechanism makes it difficult for users to quickly and effortlessly convey their feelings towards content they enjoy or dislike.

**Goals (**What would success look like if you solved the customer needs you are targeting to address? )

- ✓ To improve the quality of user engagement with the app.
- ✓ To increase the audience engagement through their content.
- ✓ Over time, this will lead to an increment in **DAUs (Daily Active users).**
- ✓ To help users in creating high-quality content.
- ✓ This helps increase the reach of posts and improve feeds of other users by suggesting content similar to what they like.

## 👫 Who's it For?

This feature is for the whole user base of Instagram. For people who post their images and are looking for validation, this feedback mechanism can prove out to be better than the already present mechanism of commenting as it is quick. (it becomes challenging for people to comment on every photo/video.)

# 👤 User Type/Persona

- 1. Active User - Ankur
- **Description:** Ankur is an active Instagram user who posts photos/videos regularly. He enjoys engaging with other users' content by liking and commenting on posts.

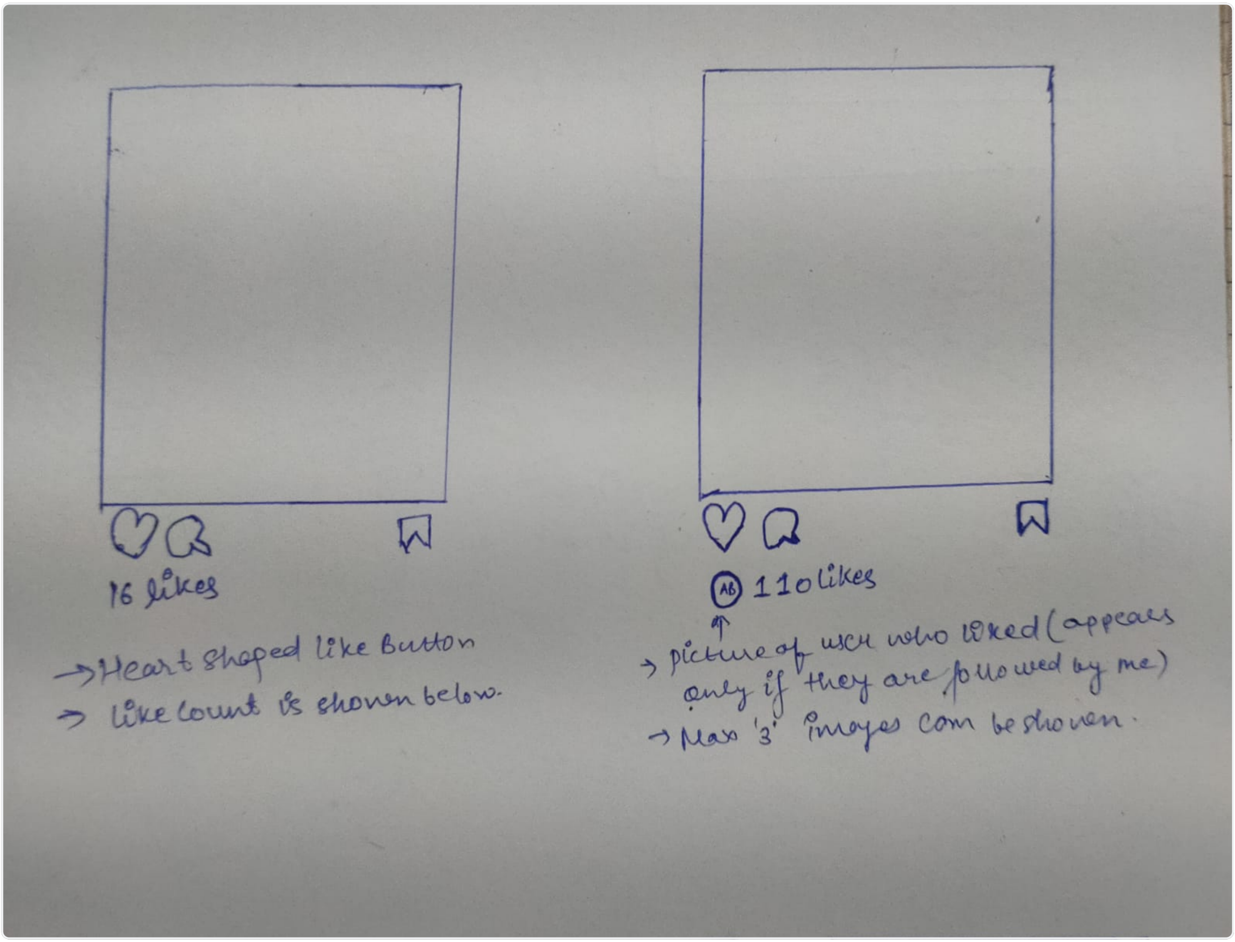
# 📱 User Stories

- **User Story 1: Liking Posts**
  - As a User, I want to be able to like posts by clicking the "Like" button so that I can show my appreciation for the content I enjoy.
- **User Story 2: Unliking Posts**
  - As a User, I want to be able to unlike posts by clicking the "Like" button again to change my mind if I accidentally like a post.
- **User Story 3: Viewing Notifications**
  - As a User, I want to receive notifications when someone likes my posts so that I can stay updated on user engagement with my content.

# ✨ Feature Functional Requirements

## 1. Like Button

- **Description:** A heart-shaped "Like" button will be added to each post on the user's feed.


16 likes
→ Heart shaped like Button
→ like count is shown below.
AB
110 likes
→ picture of user who liked (appears only if they are followed by me)
→ Max '3' images can be shown.

Like Button

- **User Acceptance Criteria:**
    - The "Like" button should be visually appealing and easily identifiable.
    - Users can click/tap the "Like" button to show their appreciation for the post.
    - The button should respond with a visual animation and convert the heart to red colour to indicate the like action. (decide with the design team).
    - A user can toggle the "Like" status by clicking/tapping the button again.
- **Technical Implementation:**
    - Add the "Like" button UI element to each post.
    - Implement click/tap functionality to handle like/unlike actions.
    - Implement visual animation for button interaction.

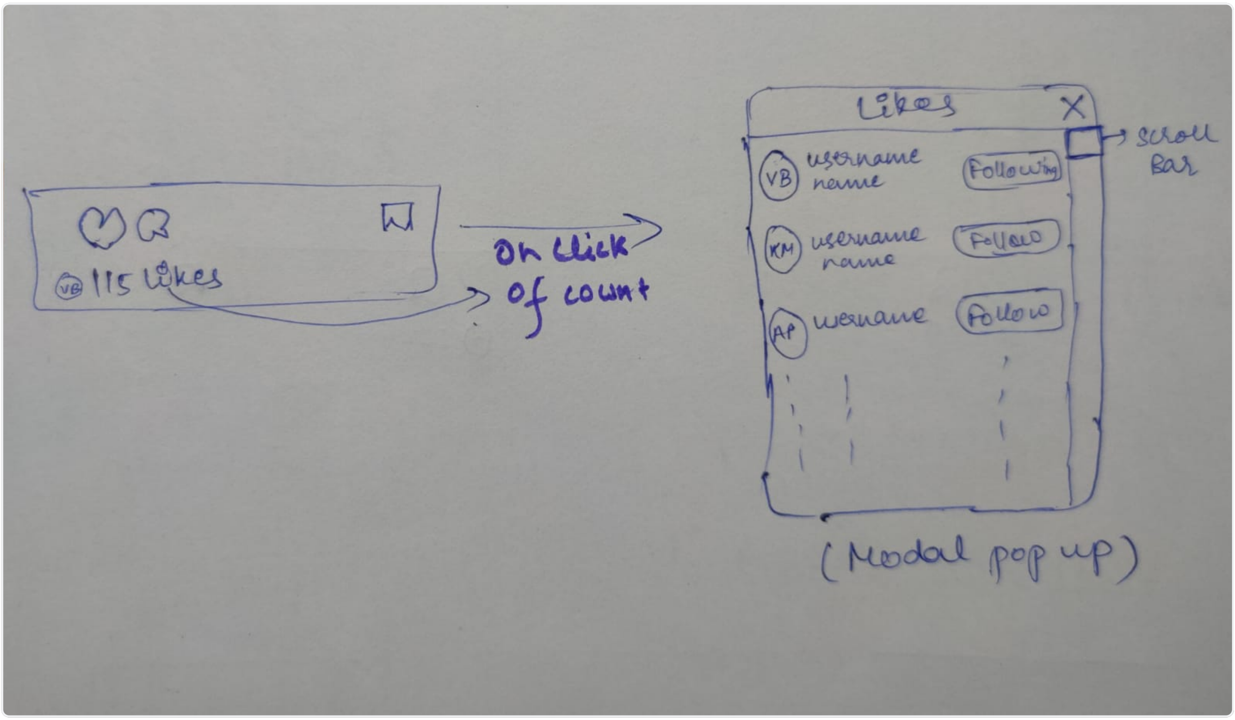
## 2. Like Count

- **Description:** Each post will display the total count of likes it has received.
- **User Acceptance Criteria:**
    - The like count should be displayed below the post.
    - The count should be updated in real time when a user likes or unlikes a post.
- **Technical Implementation:**

  - Integrate a real-time update mechanism for like counts.
  - Ensure the count is displayed accurately and consistently.

### 3. Like List

- **Description:** Users can view a list of profiles who have liked a post.



Like List

- **User Acceptance Criteria:**
  - A click/tap on the like count should pop up a modal or page with the list of users who liked the post.
  - Each user's profile picture should be displayed in  the list.
  - The list should be scrollable for posts with a large number of likes.
- **Technical Implementation:**
  - Create a UI element for the like list.
  - Implement functionality to display the list when the like count is clicked/tapped.

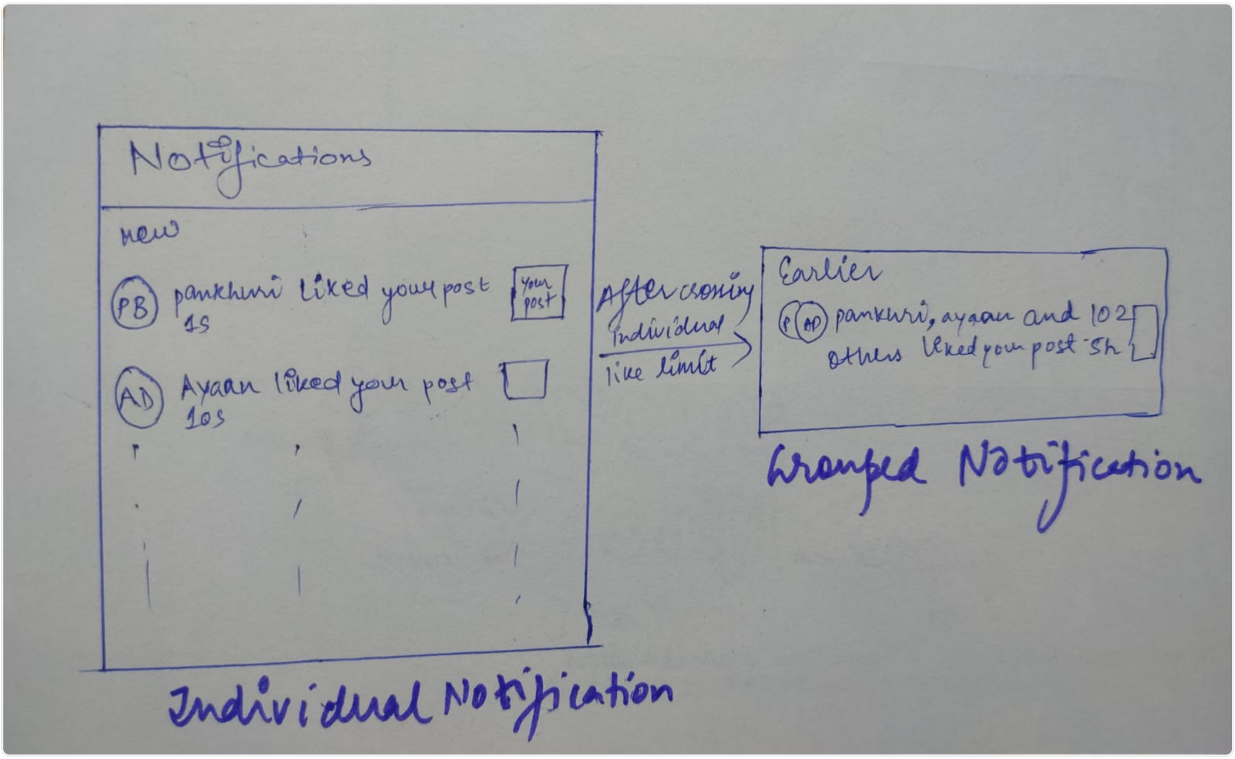
### 4. Notifications

- **Description:** Users should receive notifications when their posts are liked.
- **User Acceptance Criteria:**
  - Users should receive a notification in the notifications tab when someone likes their post.
  - Notifications should be real-time and displayed consistently.
- **Technical Implementation:**

    - Integrate a notification system to trigger notifications when a like action(onClick) occurs.

## 5. Like Notifications Aggregation

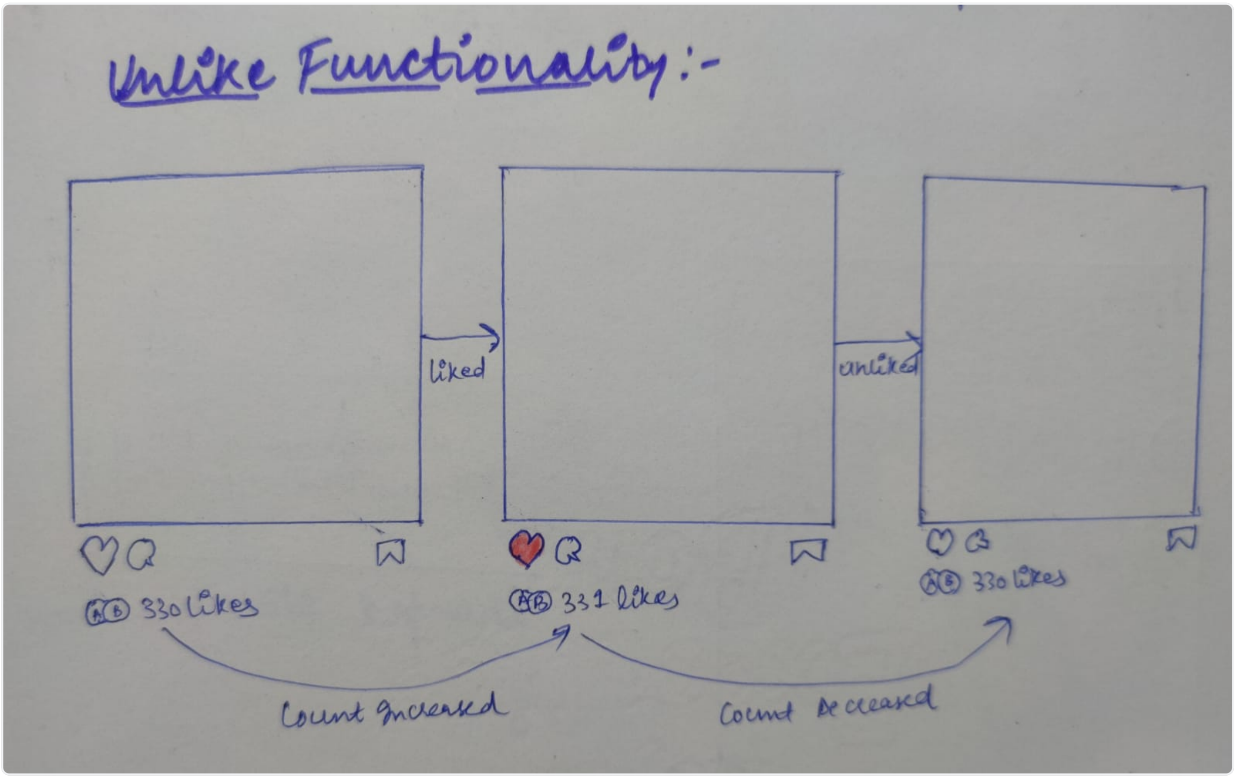
- **Description:** To improve the user experience, we will group like notifications to avoid overwhelming users with multiple individual notifications.
- **User Acceptance Criteria:**
    - When a post receives a significant number of likes (e.g., more than 10), notifications should be grouped and display a message like "30 others liked your post" instead of separate notifications for each like..
    - The grouping logic should be efficient and trigger notifications in real time.
- **Technical Implementation:**
    - Implement a server-side grouping function that groups like notifications for a single post.
    - Consider using a sliding time window to group likes within a certain timeframe (e.g., 5 minutes) and display the aggregated notification.
    - When the user views an aggregated notification, the system should fetch individual like details from the database (so that user can view individual likes as well).



## 6. Unlike Functionality

- **Description:** Users should have the ability to "unlike" a post they had previously liked.
- **User Acceptance Criteria:**

  - Users should be able to click/tap the "Like" button on a post they've already liked to remove their like.
  - The "Like" count should decrease when a user "unlikes" a post.
  - The user's like status for the post should be stored and displayed consistently.
- **Technical Implementation:**
  - Implement client-side functionality to toggle the like/unlike action.
  - Update the "Like" count in real-time when a user unlikes a post.
  - Ensure that the user's like status is reflected accurately in the database.
-



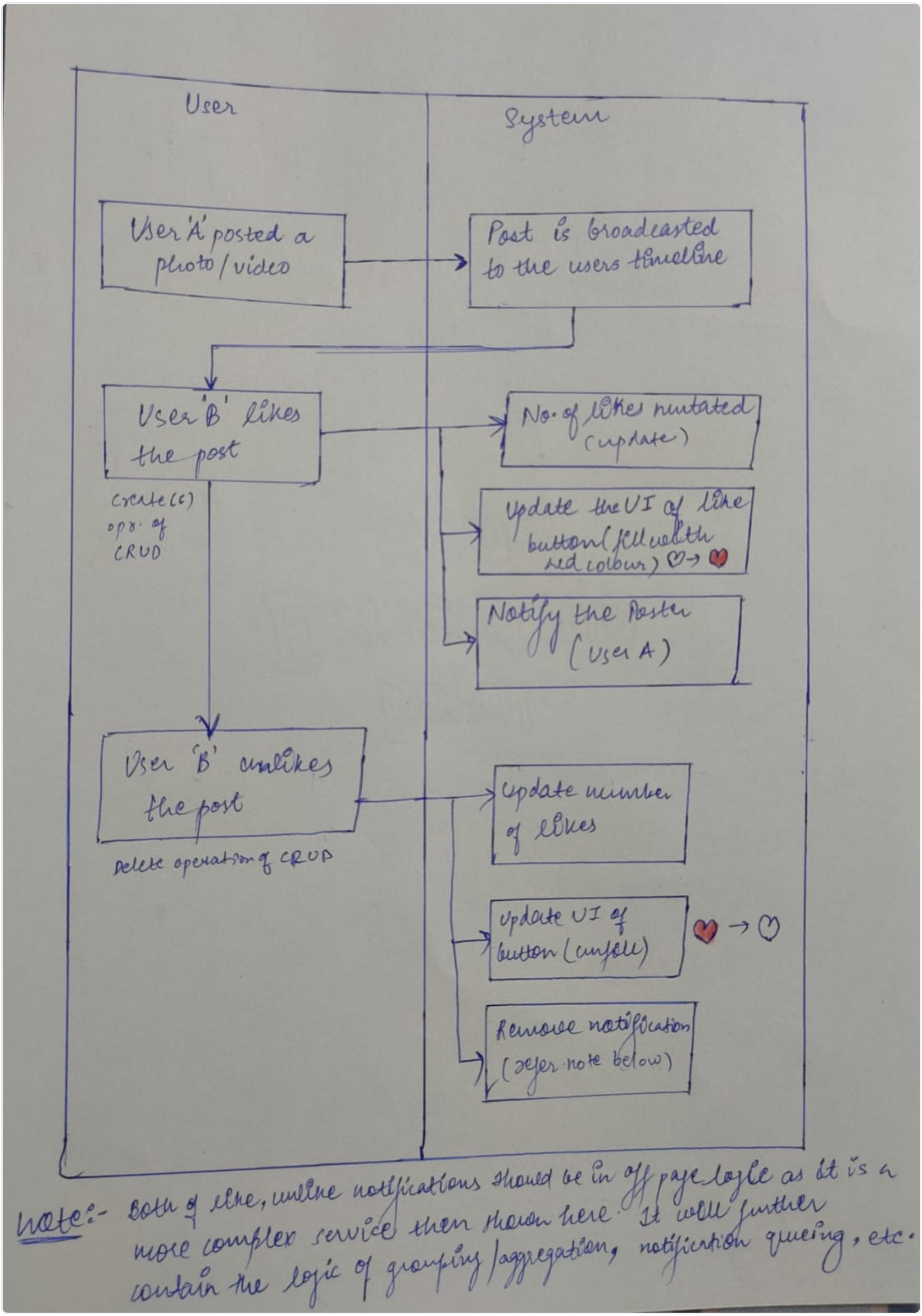
# 📝 Non-Functional Requirements

## 1. Performance

- **User Acceptance Criteria:**
  - The feature should not cause significant delays in loading or interacting with the app (basically it should not affect the existing system performance).
  - The system should handle a large number of like actions without performance degradation.

# 🔧 Business Process Flow

- This flow diagram contains basic flow of what will happen in backend when the like button gets clicked.
- Note that there will be a totally different service for notifications containing complex business logics like queuing of notifications (may be using SQS) , database transactions etc. Here we have covered the high level working just to explain the newly added functionality on already built notification service
-


User
System
User 'A' posted a photo/video
Post is broadcasted to the users timeline
User 'B' likes the post
create(c) opr. of CRUD
No. of likes mutated (update)
Update the UI of like button (fill with red colour) ♡→ ❤
Notify the Poster (User A)
User 'B' unlikes the post
Delete operation of CRUD
update number of likes
Update UI of button (unfill)
❤ → ♡
Remove notification (refer note below)

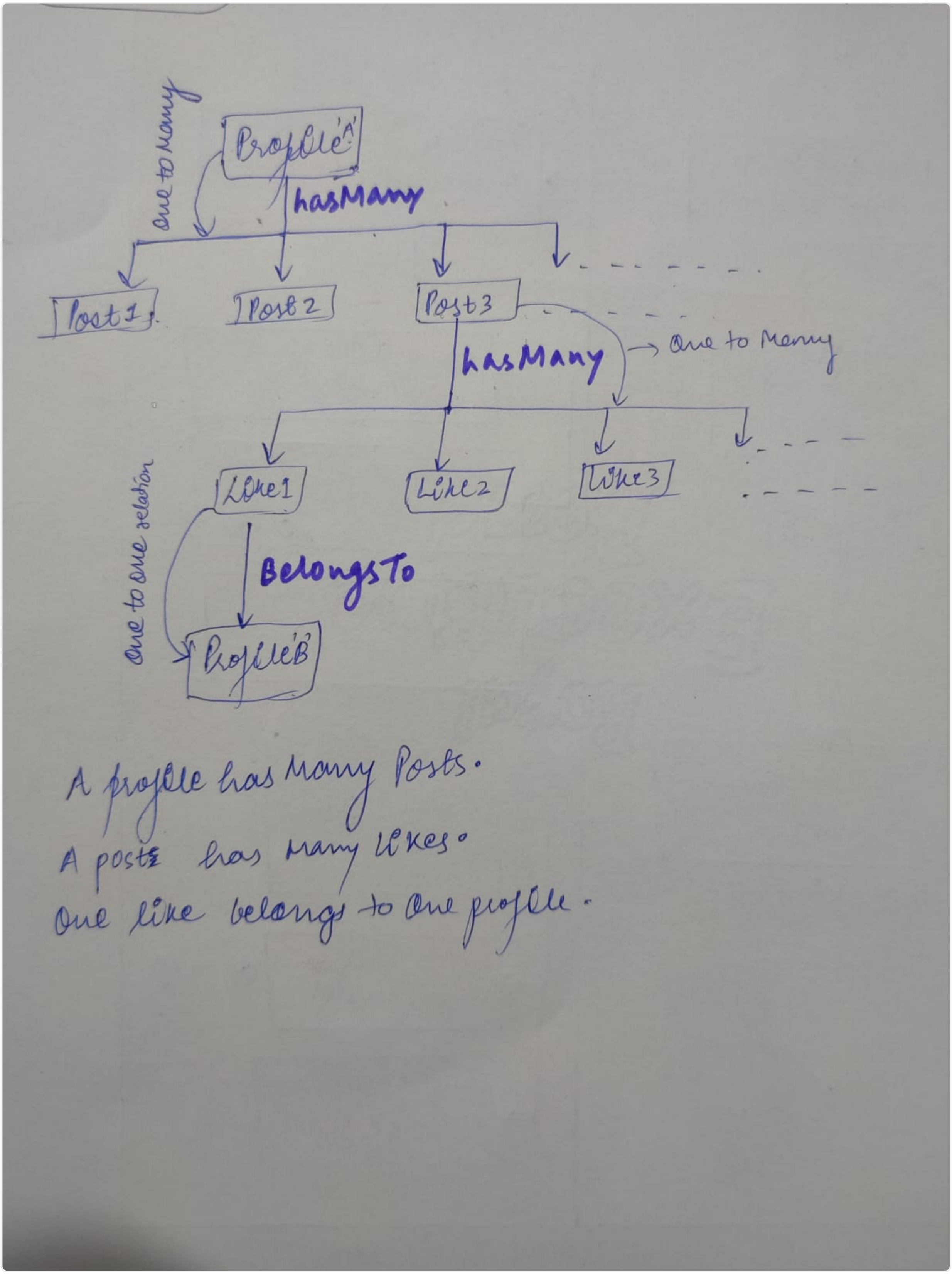
Note:- Both of like, unlike notifications should be in off page logic as it is a more complex service then shown here. It will further contain the logic of grouping/aggregation, notification queing, etc.

Swimlane diagram for like button

Note:

## 💡 Additional Details for Developers

**BASIC RELATIONSHIPS BETWEEN VARIOUS ENTITIES:**



A profile has Many Posts.

A posts has many likes.

One like belongs to One profile.

## APIs based on Functionalities:

- **Like Button API :**
    - *HTTP Method:* POST
    - *Request Payload:* User ID, Post ID
    - *Response:* Confirmation message and updated like count
    - Sample Response JSON :
-
  ```
  {
  "status": "success",
  "message": "Post successfully liked",
  "data": {
    "userId": "12", // User ID of the user who liked the post
    "likeCount": 123, // Updated like count on the post
    "likeStatus": true // Like status indicating the user liked the post
  }
  ```
- **Likes Count API :**
    - *HTTP Method:* GET
    - *Request Payload:*  Post ID
    - *Response:* Confirmation message and total like count
    - Sample Response JSON :
-
  ```
  {
  "status": "success",
  "message": "Liked Fetched Successfully",
  "data": {
    "postId": "12",
    "likeCount": 123
  }
  ```
- **Unlike Button API :**
    - *HTTP Method:* POST
    - *Request Payload:* User ID, Post ID
    - *Response:* Confirmation message and updated like count
    - Sample Response JSON :
-
  ```
  {
   "status": "success",
   "message": "Post successfully unliked",
   "data": {
     "userId": "12",
  ```

```
"likeCount": 122,
"likeStatus": false }
```

- **Like Notifications API :**
  - Sample JSON :

  -
    ```
    {
     "status": "success",
     "message": "You've received a new kind of notification",
     "data": {
       "notificationId": "1234", // Unique ID for the notification
       "type": "like",
       "senderUserId": "10", // User ID of the user who liked your post
       "postUserId": "12", // User ID of the post owner
       "postId": "456789123", // ID of the liked post
       "timestamp": "2023-10-19T10:30:00Z" // Timestamp of the notification
     }}
    ```
- **Like List API**
  - *HTTP Method:* GET
  - *Request Payload:* Post ID
  - Sample JSON:
    ```
    {
     "status": "success",
     "message": "Likes retrieved successfully",
     "data": {
       "postId": "456789123", // ID of the post
       "likes": [
         {
           "userId": "13", // User ID of the user who liked the post
           "username": "TESTuser1",
           "timestamp": "2023-10-15T14:45:00Z"
         },
         {
           "userId": "14",
           "username": "testuser2",
           "timestamp": "2023-10-15T15:20:00Z"
         },
         // Additional like entries for the same post
       ]
     }}
    ```

```
{   "status": "success",   "message": "Likes retrieved successfully",   "data": {     "postId": "456789123", // ID of the post     "likes": [       {         "userId": "123456789", // User ID of the user who liked the post         "username": "sampleuser1", // Username of the user         "timestamp": "2023-10-15T14:45:00Z" // Timestamp of the like       },       {         "userId": "987654321",         "username": "sampleuser2",         "timestamp": "2023-10-15T15:20:00Z"       },       // Additional like entries for the same post     ]   }}
```

## 🏆 Success Metrics For Like Feature:

- Increase in user engagement.
- Increase in user retention.
- Increase in visibility.
- Increase in the number of downloads.
- Increase in the amount of content posted.
- Increase in the frequency of the content posted.

After this PRD, we can discuss and prepare a GTM (go to market ) strategy which will contain the information about marketing, rollout plans, analytics, feedback etc.