University of
Applied Sciences

Programming 2

OOP Practical Exercises

Dr. Dhungana
Dr. Klikovits
SS2024 • Krems

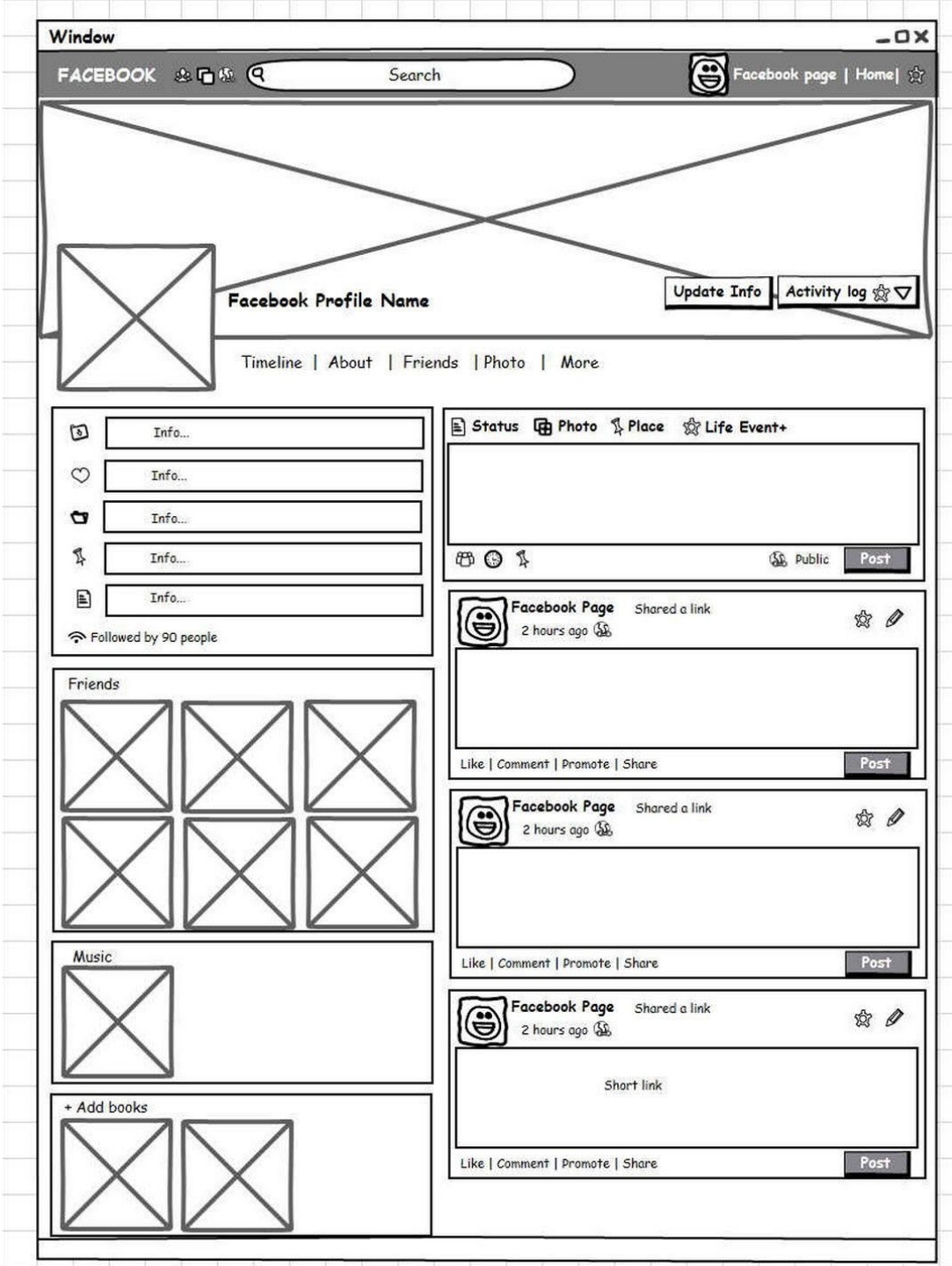
Exercise: Facebook like platform

Create a UML class diagram to replicate the functionality of a Facebook Profile Page.

- **Classes, Attributes, Methods**
- **Associations**
- **Inheritance**

Some potential candidates for classes

- **User**
- **UserProfilePage**
- **Timeline**
- **ActivityLog**
- **Post**
- **Comment**



Step 1: Add a Friend

User A can add user B as friend

- User B gets notified (receives a request)
- User B can selectively choose to approve some requests
- If approved, User B is added to the friend list of A and vice versa
- If not approved, request is simply deleted.

```
u1 = User ("Joe")
u2 = User ("Jill")
print (u1.name + " Friend List = " + str(u1.friends))
u1.addFriend(u2)
print (u2.name + " Friend Requests = " + str(u2.requests))
u2.approve (u2.requests[0])
print (u1.name + " Friend List = " + str(u1.friends))
```

output

```
Joe Friend List = []

Jill Friend Requests = [Add friend request from Joe]

Approving add friend request
Joe Friend List = ['Jill']
```

Step 2: Post on Users' Wall

User A can post on user B's wall

- User B gets notified (receives a request)
- User B has a list of post requests and can selectively choose to approve some requests
- If approved, User A's message is added to the wall of B

```
u1 = User ("Joe")
u2 = User ("Jill")
u1.addFriend(u2)
u2.approve (u2.requests[0])

p1 = u1.post("What a lovely weekend")
print (u2.wall)
p2 = u2.post("Sunny and warm!")
```

output

```
Joe just posted 'What a lovely weekend'
Wall: Jill
=========
Wall: Jill
=========
[Joe]:What a lovely weekend
Jill just posted 'Sunny and warm!'
```

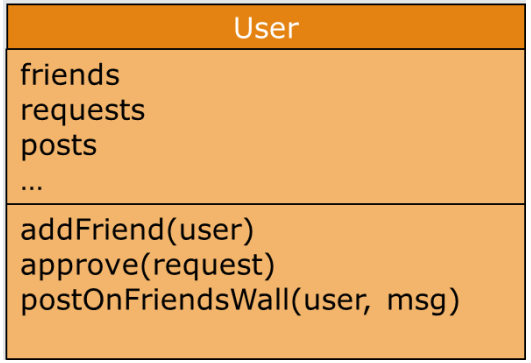
Step 3: Wall Posts

User A can post on user B's wall

User B gets notified (receives a request)

User B has a list of post requests and can selectively choose to approve some requests

If approved, User A's message is added to the wall of B




```
u1 = User ("Joe")
u2 = User ("Jill")
print (u1.posts)
req = u1.postOnFriendsWall(u2, "Hey, happy birthday")
u2.approve (req)
print (u1.posts)
```

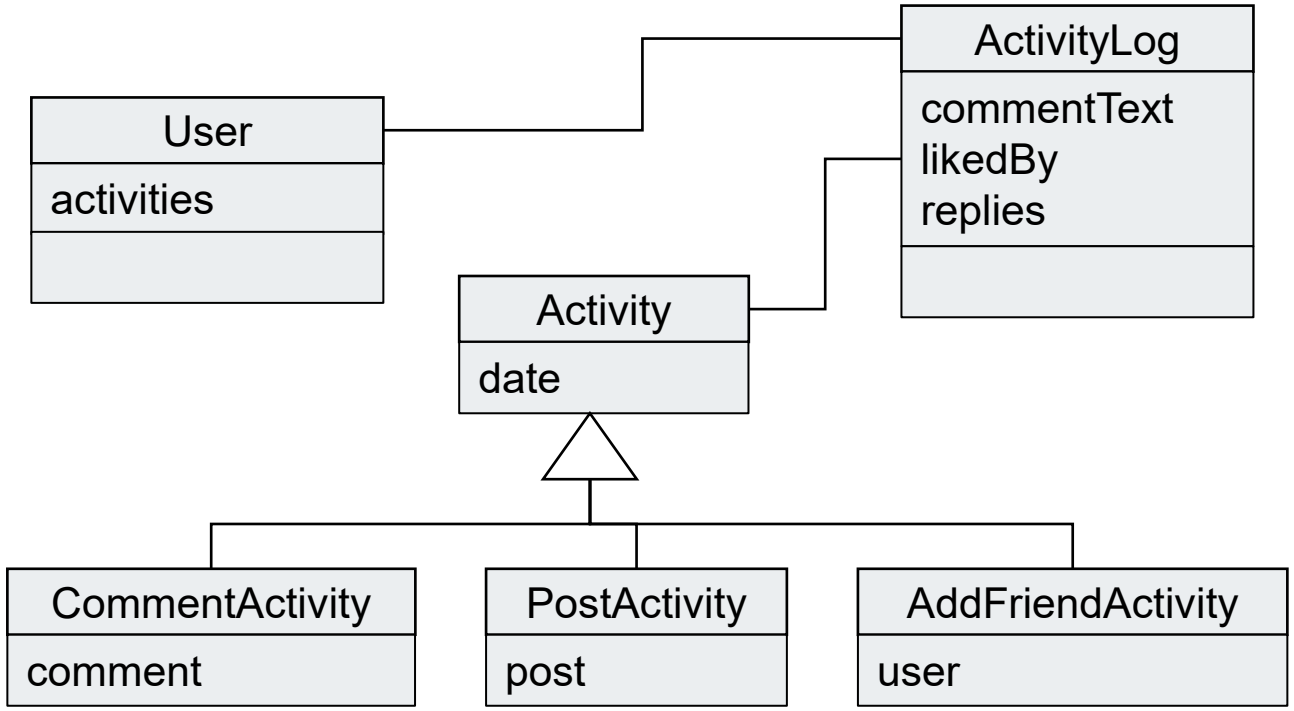
output

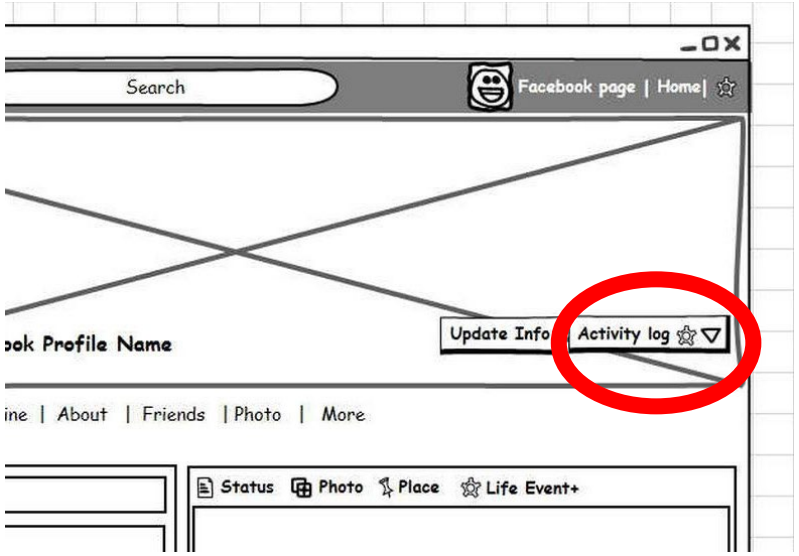
```
[]
Approving add post request
['Hey, happy birthday‘]
```

Step 4: Activity Log

Each interaction of the use with the system is saved in Activity Log

User browse the entries in the log, but not change the history



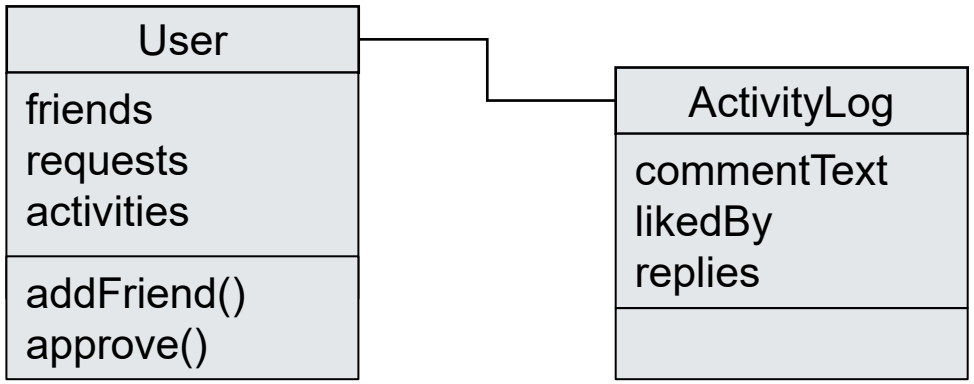


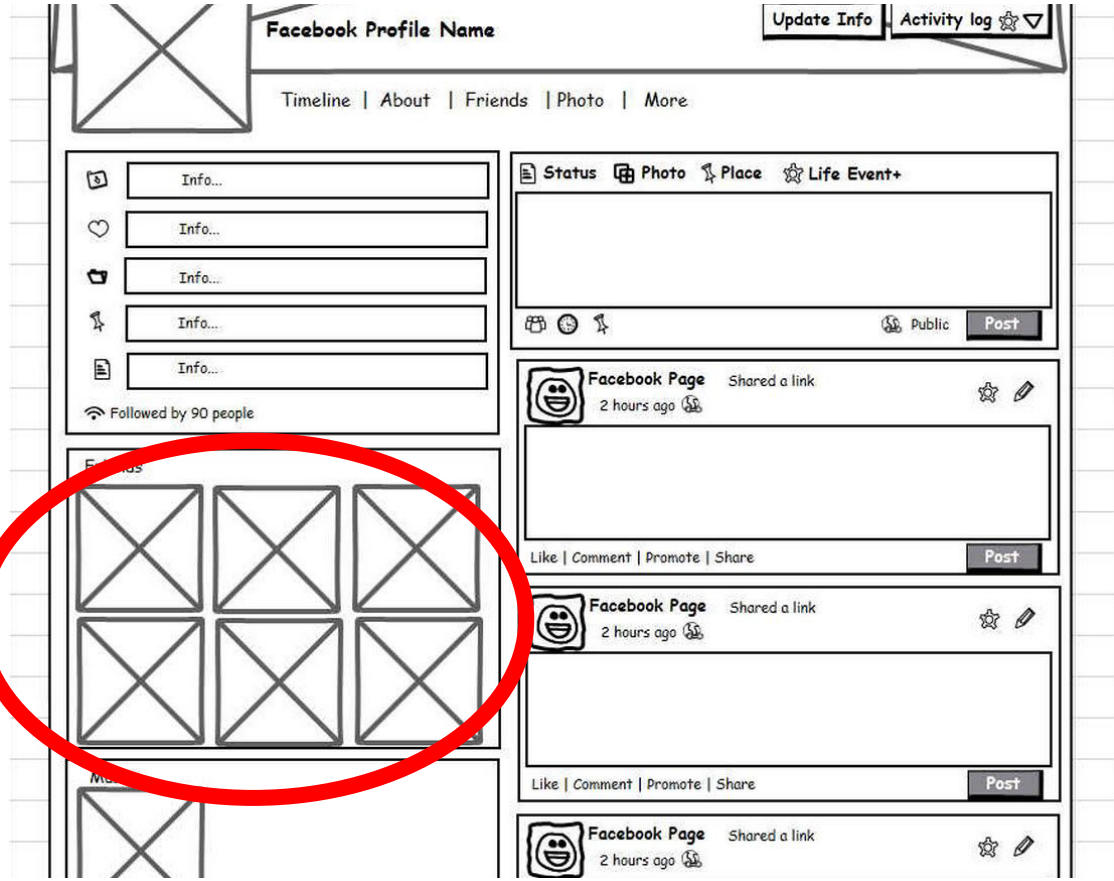
Step 5: Sorting Friends

Based on how often a user interacts with other users (comments, likes, shares), the order of the appearance of the users in the friends list changes

More interaction -> closer friends

List of friends is sorted accordingly





Try a Youtube-like platform

Task 3.1 Classes

Create a UML class diagram to replicate the functionality of Youtube Page.

- Classes, Attributes, Methods
- Associations
- Inheritance

Some potential candidates for classes

- Video
- User
- Channel
- Comment
- Subscription
- History
- Advertisment

.

Task 3.2 Methods

Add behavior to your classes, think of typical activities like uploading a video, subscribing to a channel, getting notifications, maintaining a history of viewed videos, displaying Ads, etc

Create some example objects of the classes and test the behaviour based on some example interactions of the user with the platform.