Build a FakeBook News Feed

System Design

* The system has a list of **USERS**.
* Each user will have a group of **FRIENDS**.
* A user can **POST** a message which only his friends can see.
* A user needs to be able to read recent messages from his friends - this is what we call a **NEWS FEED**.

Task List

*Fast forward a week and your team has already completed these features:*

1. Build and Display a List of Users - DONE
2. Build and Display a List of Friends - DONE
3. Check if a User is a Friend - DONE
4. Randomly Generate some Posts for Testing - DONE

*Test the features your team already wrote:*

print\_users()

print\_friends()

print(is\_friend("Jackie"))

print(POSTS)

*And then work on the remaining tasks:*

1. Add Friends
2. Build a News Feed
3. Add a Post
4. Like a Post

*Work in pairs. Assume we are already logged in for a single user.*

TASK 1.

Test the features your team already wrote:

print\_users()

print\_friends()

print(is\_friend("Jackie"))

print(POSTS)

TASK 2.

You have no friends! How sad...

Let’s fix that by adding some friends:

a) Hard-code adding a friend

b) Ask the user to choose a friend’s name

c) Ask the user to enter 5 friends’ names

TASK 3.

Trying to display the POSTS just doesn’t seem to work - it’s a MESS! There are just far too many to read and it doesn’t look pretty…

a) Build a News Feed to display all Posts

b) Build a News Feed to display only Posts from a Friend

c) Build a News Feed to display only Posts from your Friends

d) Build a News Feed to display only the last 20 Posts from all your Friends

TASK 4.

Time to start interacting with the world!

a) Add a Post

b) Like the Post you Added

c) Like all Posts from a Friend

Extension Ideas

1. Change the names of the 20 Users and change the Random Messages
2. Generate 200 Random Posts instead of 100
3. Change the way the Post is displayed to look nicer.
4. Make sure that no Post is repeated by the same User
5. Have you noticed that there is a bug? When you add a new post, it is displayed last in your feed instead of first. Modify the **Post** class to save the full timestamp, and sort the posts before displaying them.