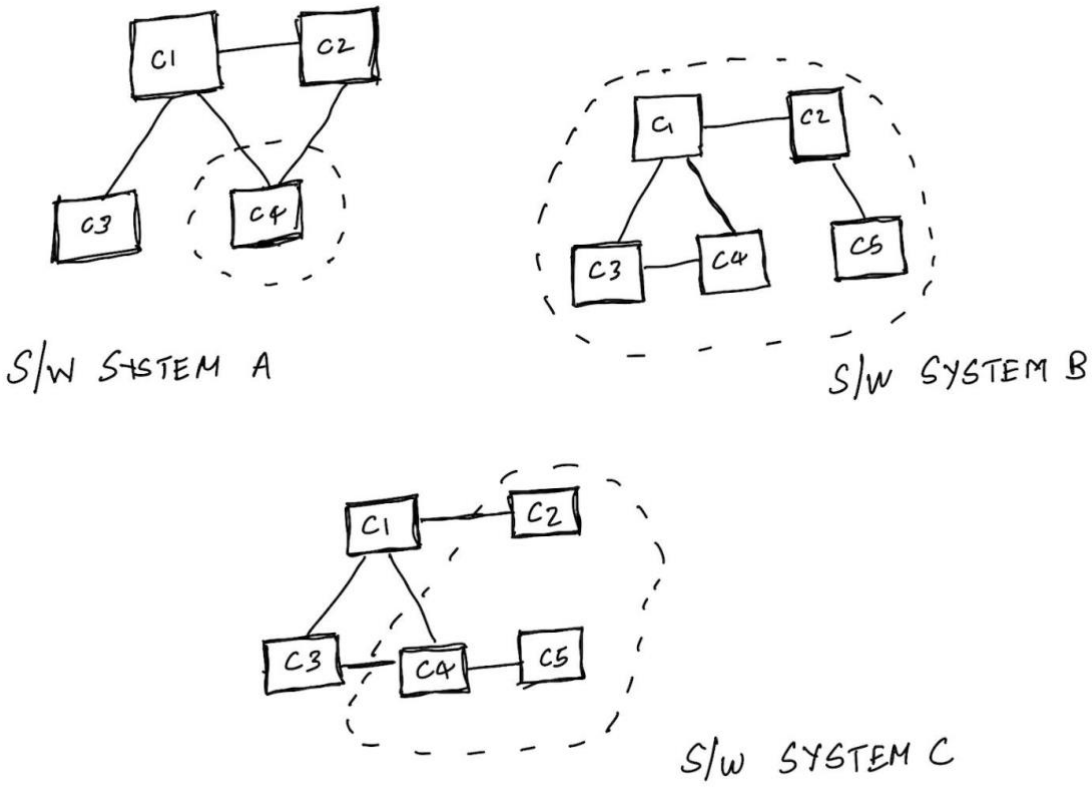
[[[CS 362 - Software Engineering II
Homework 3

Question 1 **10 Points**

Take a look at the software systems below. Assume that the following software systems contain the following components. There are 3 Software systems A, B, and C that consist of individual components within them.



Deliverable:

i) For System A, look at the dotted lines and describe whether you'd apply unit/integration/system testing and explain why.

I would choose unit testing for system A since it has a low number of components. This would make it easier to test each individual because there aren't too many connections and components to test for.

ii) For System B, look at the dotted lines and describe whether you'd apply unit/integration/system testing and explain why.

I would choose system testing since the whole system is also every component. To ignore the interactions of such a comprehensive system like that could cause the testing phase to be too narrow and lead to false confidence when integrating/viewing the tested components as a whole.

iii) For System C, look at the dotted lines and describe whether you'd apply unit/integration/system testing and explain why.

I would choose an integration testing method. There three macro-components that interact with each other. The subcomponents are essentially two that only interact with the other macro-components separately. These would make great sections of testing and those section only deal a few determined actions. The testing would provide good indications of the system's performance as a separate whole. This could also be done more methodically and micro-scaled with a unit testing method. I think time could be saved if done by integration.

Question 2

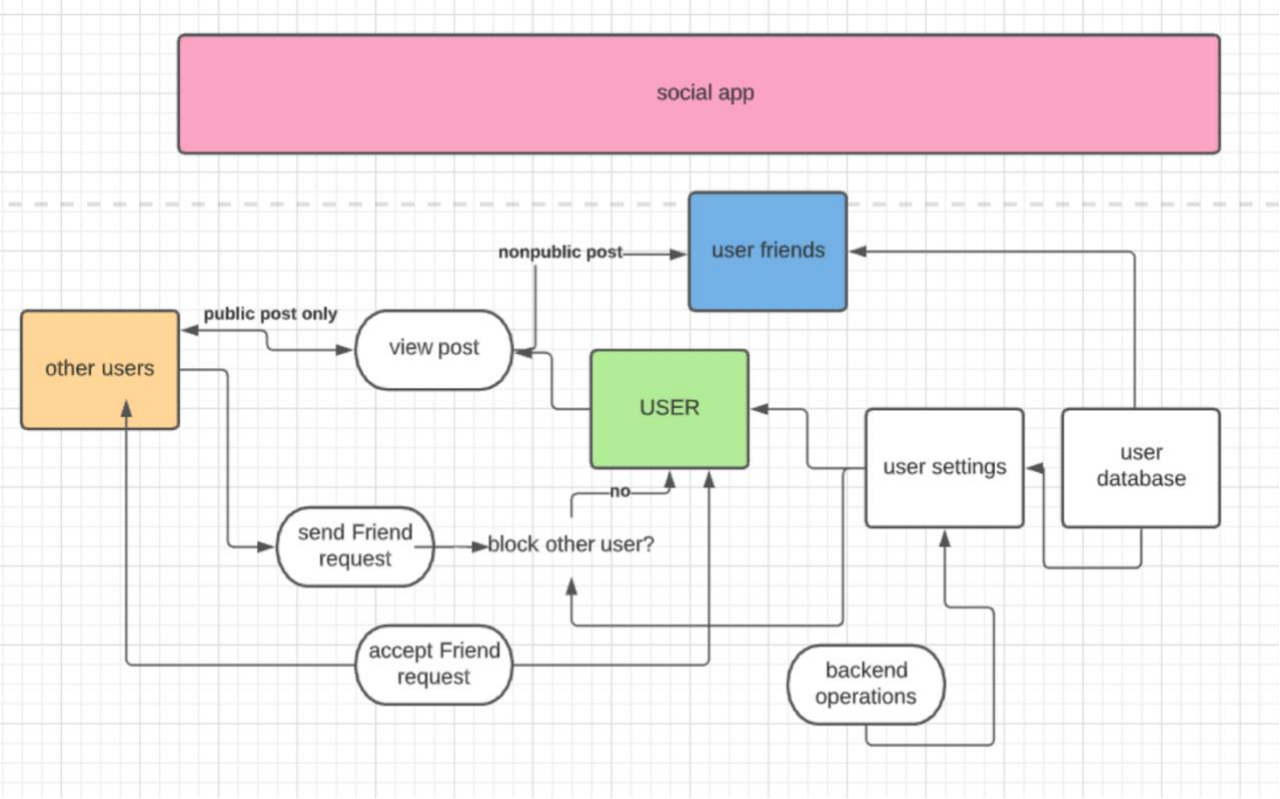
From **In-class activity 1**: **10 Points**

Remember the question where you were asked to draw the structure of a social media platform of your choice?

Deliverable:

Revisit your system from *In-class activity 1*. Describe or draw your system.

Assuming that it has a few components within it, answer the following:



i)Verification with 2 examples

Checking if the auto-generate will search the user's database and make auto suggestions from the user's friend list.

Making sure that the friend/social groups update its member/memberships database upon accepting a request.

Validation with 2 examples

Does the user find the homepage easy to navigate?

Is it easy for the user to find and become a member of a social group using the application?

ii) Functional tests 2 examples Non-functional tests 2 examples

Functional

i) Does saving a “friend” automatically cause my account to follow that friend’s feed.

ii) Does updating “my status” automatically inform the followers of my feed?

Non-functional

i) Does the homepage refresh every time the users log back in the application?

ii) Does the user’s inbox keep a copy of every message received/sent.

Question 3 **10 Points**

Version Control

Leap year program

i) Assuming that you have not written the leap year program with error handling the input in mind, write the program with error handled input.

ii) You will check in the following to Github: a. Leap year code without error handling

b. Leap year code with error handling

c. The flowchart in the readme.md file.

For part, c refer to this.

Provide the URL to your git repository in the pdf that you submit.

https://github.com/jerred-shifflett/leapyear.git

Question 4 **10 Points**

You are now aware of Agile and Waterfall methodologies. How would testing differ between both the methodologies?

Deliverable:

You have two methodologies - Agile and Waterfall. How would the software testing process be carried out in both of these methodologies? Write down 2 differences with a project as an example.

i) Describe a project in a few words.
Project- creating software interaction between user and IOT capable coffee/tea machine.

ii)In the tabular form below explain how you'd apply testing in waterfall(2), how you'd apply testing in agile(2).

Testing in Agile	Testing in Waterfall
I would imagine it depends on general flow of the project, but testing would most likely be done with iterations of feature changes. The testing results of the last phase would most likely be the task of the next Spring Cycle or two. Testing could also be implemented as errors or worse arise during the software development.	With the extensive planning intrinsic to the waterfall design testing would happen after the implementation process. There may be small amounts of unit testing, but the idea of waterfall is to produce a first try product. Most of the testing will likely occur just before the product is ready to be launched and the prototype is its most functional.