

Question 1;

- i) **System A** would be used for Unit testing. This is so that we would be able to test the individual components in the system as shown. Component  $C_4$  is the one in question that is to be tested to validate that it performs as expected before moving onto the next unit. I chose Unit Testing here because the diagram shows that the component is being tested in isolation from the rest of the system.
- ii) Integration testing would be best for **system C**. As the diagram shows, Integration testing would allow us to test the individual components as they are added to the system. Components  $C_2$ ,  $C_4$ , and  $C_5$  are being tested here and then added to the system as a whole to make sure they are functioning as intended when they are added. This is mainly used to highlight an issue between components of the system.
- iii) System testing would be best suited for **system B**. The diagram shows that we are to be testing every component in the system simultaneously in order to expose issues with the entire running system. This is mainly to make sure that the system itself falls within the specified system requirements.

Question 2; My chosen social media platform for In-class activity 1 was Facebook.

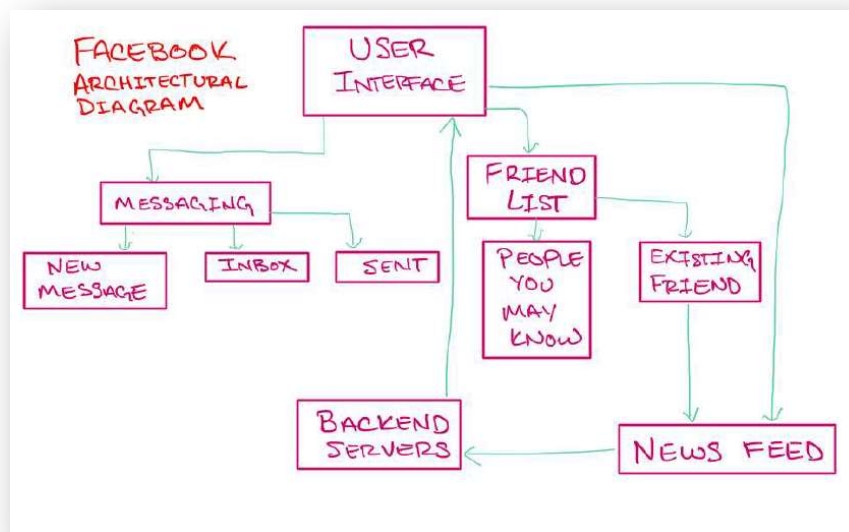


Figure 1 - Screen Clip taken from In-Class activity 1 depicting a social media platform (Facebook)

- i) Verification:
  - a) Making sure that there are no spelling issues in any of the pages or buttons. This is to verify that we are building the product correctly so that the user has the most professional experience possible while using our social media system.
  - b) Making sure that the design and colors of the system are coherent and pleasing. Once again, this goes back to the previous example where we would want to build the most professional, clean, and user-friendly system as possible. We want to make sure that we are building the system right.
- Validation;
  - a) Making sure that messages are being delivered to their intended recipients and not someone else. Because a lot of the user experience of this social media platform takes place on the messaging side of the system, we would want to validate that the messaging system is being constructed correctly and that the messages are being delivered correctly.
  - b) Making sure that user posts are visible to the community on the specific user profiles. Besides messaging, the Facebook platform is designed to share life events of the user to other people. Because of this, we would want to validate that the mass-sharing outlet is working correctly and is displaying the posts that the user would like to display.
- ii) Functional System Test;
  - a) An automated test to determine if the messaging system is able to handle all of the known input characters of different languages and different combinations of these characters.
  - b) A manual test of each of the buttons that are present on a user's profile to determine that the button is linked to the correct end-page.
- Non-Functional Test;
  - a) An automated test to determine how many users can be simultaneously logged onto the system
  - b) An automated test to determine how fast a user's feed/wall updates after a post is pushed to it.

### Question 3;

- i) Full REPO for this question can be found at:  
[CS362\\_Homework/Week\\_4 at master · pattons-OSU/CS362\\_Homework \(github.com\)](https://github.com/pattons-OSU/CS362_Homework)
- a) Leap-year code without error handling may be found at:  
[CS362\\_Homework/leap\\_year\\_no\\_error.py at master · pattons-OSU/CS362\\_Homework \(github.com\)](https://github.com/pattons-OSU/CS362_Homework/blob/master/leap_year_no_error.py)
- b) Leap-year code with error handling may be found at:  
[CS362\\_Homework/leap\\_year\\_with\\_error.py at master · pattons-OSU/CS362\\_Homework \(github.com\)](https://github.com/pattons-OSU/CS362_Homework/blob/master/leap_year_with_error.py)

I added an error-handling for “ValueError”, if the user did not enter a valid year as an integer. I literally could not think of a single other error that could occur with this program

- c) README.md file may be found at:

[CS362 Homework/README.md at master · pattons-OSU/CS362\\_Homework \(github.com\)](https://github.com/pattons-OSU/CS362_Homework/blob/master/README.md)

#### Question 4;

- i) I would like to focus on an augmented reality program that displays the torque specs for bolts and fasteners upon re-assembly of a vehicles mechanical systems on a user’s display (tablet or phone). This program would use information from the original equipment manufacture service manual so that the correct torque specs are displayed to the mechanic as they are moving through the repair process.

ii) **Testing in Agile:**

Because agile testing can occur almost immediately, there would be a lot of unit testing happening here as we would be looking at testing each component as they are being developed. Once we have the individual components tested, we would look at moving onto some integration testing because we would be integrating these completed components into the system as a whole. A good unit test here would be to make sure that the software was identifying the correct bolts to torque. If the wrong bolts were going to be identified then the user would most likely break something in the real world from overtightening. Once we have that specific unit battened down and completed, then we would want to make sure that the correct torque speck is being displayed on the correct bolt through the use of the augmented reality camera. This would be an example of the integration testing phase.

**Testing in Waterfall:**

Testing here is a little different because it has to take place during the predetermined testing phase, after the software has been built. This would definitely lend itself to a system testing model where the system in its entirety is put to the test only after it has been completed. There is a little room for some additional tests when the software beginning to evolve and change and then we would be looking at more of an integration type testing again, because we would be adding features and functionality to the system after the fact and making sure that it is working as it is integrated and upgraded. With this type of testing in a waterfall method we would ensure that the program met all of the customers needs at the end of the project, this would be much like a beta-test style. We would release the program to some users and have them test it as a whole and in the software’s intended environment.