Michael Smith

Question #1 Deliverables:

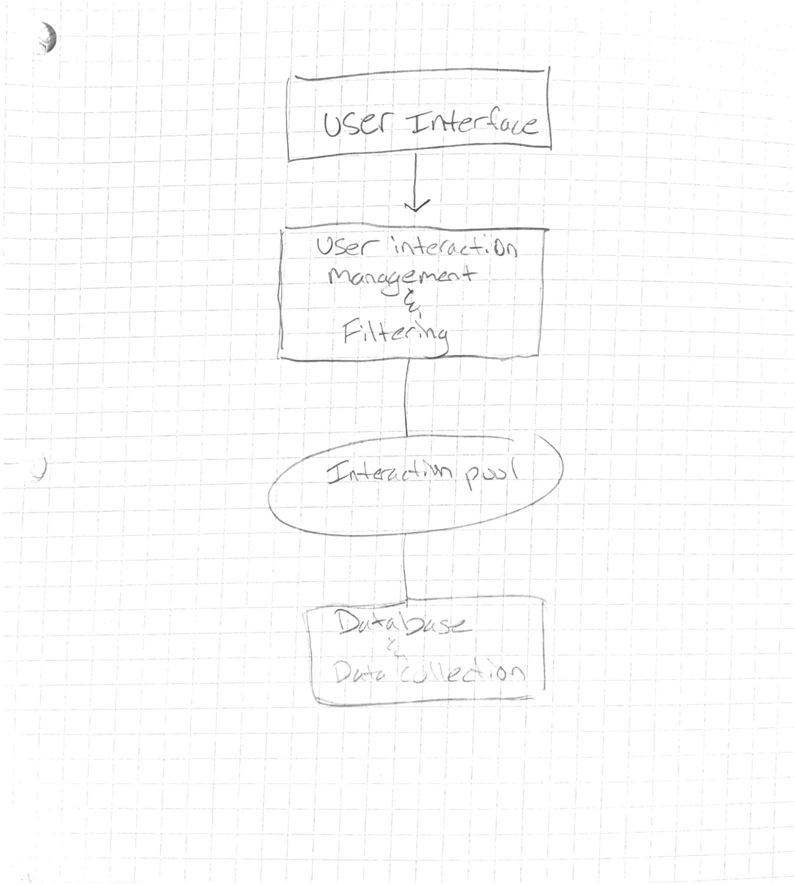
1. Unit testing would be the best form of testing for system A. This is because Unit testing involves testing an individual component of the system to make sure it runs as expected. Judging from the dotted lines in system A, only c4 is circled which means that only a single component would be tested.

2. System testing would be the best form of testing for system B because the dotted lines circle the entire system and all its components. System testing involves testing the entirety of the system and is used to determine if the system meets the requirements.

3. Integration testing would be the best form of testing for system C because the dotted line ecompasses three components and not the entire system. Integration testing is done to make sure individual components still operate correctly when interacting with other components. Components c2, c4, and c5 would be tested on how well they integrate with the system.

Question #2 Deliverables:

1. In-class Activity #1(Instagram):


User Interface
User interaction Management & Filtering
Interaction pool
Database & Data Collection

Note: I missed the first in-class activity so I apologize if this system sketch isn't exactly what was expected.

2.

Verification Examples	Validation Examples
For a social media site like Instagram, I would first verify the users profile page. This would include checking if the profile page looks correct and reflects the user in the form of # of followers, # of people following, profile picture, and a history of the users posts. All of these things should appear on the users profile page.	When operating the social media site, you could validate the following feature. The user could follow a new person and then see if that user was added to the people they are following on their profile page. This would validate the fact that the software keeps track of how many people a user is following.
Verifying each individual post for all users. This would involve any characteristics that apply to all posts like the number of likes and comments as well as a title showing which user made the post. This example would take place on the user interface in the diagram above.	Testing and checking the database would be the best way to validate various parts of the software. You could validate by making a new post and checking the database to see if the data made on the user interface is found within the database.

3.

Functional Test Examples	Non-functional Test Examples
As a social media site, user communication should be a requirement. A functional test for this would be testing the comment characteristic for each post and possibly other ways for user-to-user interaction. This could be tested by evaluating how messages are sent to and from users as well as the notifications they get.	There would be a lot of users operating the social media site at the same time. A non-functional test would be evaluating how much data the software can handle at the same time so that the software doesn't crash or function improperly. This would test the page reload speed and login for example.
Making posts to your own profile page is a requirement for all social media sites. A functional test would be testing how individual posts are made and how they are present to users. This could be tested with the "create" button while making the post.	Testing usability would be important for a social media software. Say a user isn't following anyone, there should still be content that is suggested to them so that they are able to have a similar experience. Testing how content is suggested to users would be a good non-functional test. This would test how much content could be found and ready to display to a user.

Question #3 Deliverables:

1. Github

Question #4 Deliverables:

1. Project example is creating an accounting software for a business.

2.

Testing in Agile	Testing in Waterfall
Testing in Agile would mean that there isn’t one specific stage/time period for testing, so the testing would have to be ongoing. This would mean both the development team and testing team would need to work together throughout the sprints. For this project, you would need both the development and testing teams to work together while the accounting algorithm functions are being developed.	The Waterfall methodology has its own specific time period where testing takes place. This means that testing would happen after the full development of the accounting software. For this project, the development team would need to finish the software before testing, allowing for the development to happen faster. This would mean that the testing and dev team don’t work as closely together as they would in Agile.
Regression testing would also need to happen after every sprint. This would need to happen in order to see if the new changes meet the requirements and if they negatively impact code from past sprints. For this project, say the second sprint adds a new input system for the user. Regression testing after the sprint would make sure the new input system doesn’t interfere with the algorithm functions that generate an output based on the input.	Since testing happens at a specific stage, acceptance and regression testing would happen at the very end. For this project, the fully developed accounting software would go through all forms of testing. This is unlike Agile because there is less testing going on while the software is being developed, potentially making this stage longer.