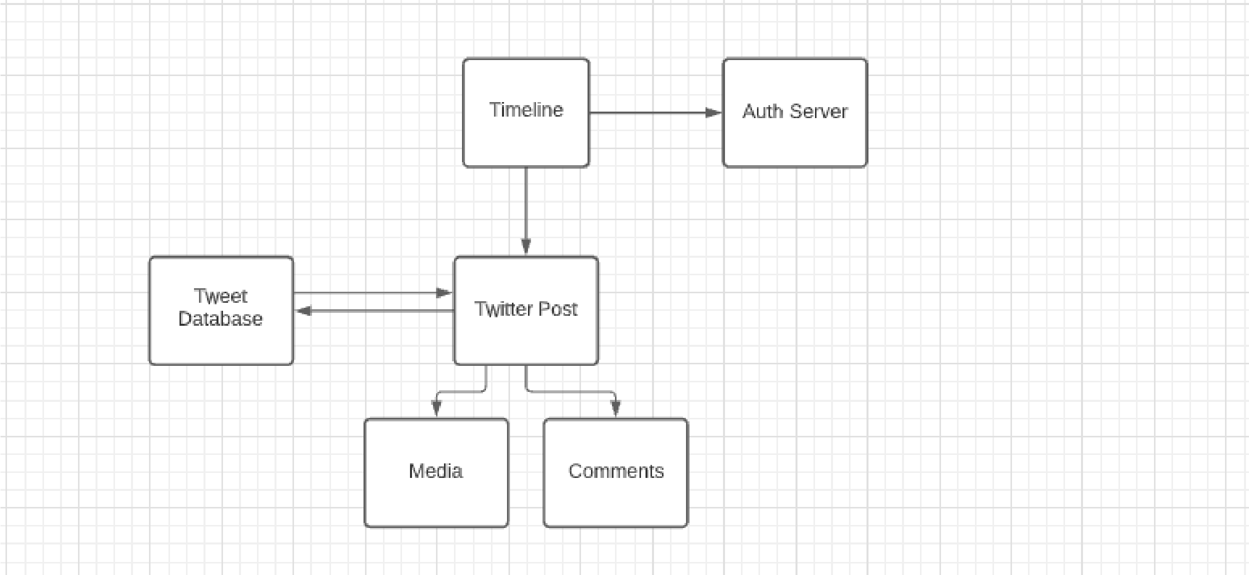
1a) I would apply unit testing to the software system A. This is because unit tests deal with the testing of individual components on a software system, and system A has an individual component isolated.

1b) I would apply integration testing to the software system C. This is because integration testing is meant to check how modules of components work together in a system and how they interact with one another.

1c) I would apply system testing to the software system B. This is because for system testing, you are testing the entire system as a whole, not just the individual components. This testing is meant to check to see if requirements are met.

2a)



Verification 1: Test that the tweet database is able to communicate with the twitter post correctly when creating a tweet. This would be an integration test between the two components.
Verification 2: Test for consistency in all documentation for each module. Make sure documentation conforms to guidelines and is present for each module.
Validation 1: Test that the GUI of the software meets usability requirements specified in the design documentation when executing the system.
Validation 2: Perform a system test to see if we have met system requirements of the application, that being letting an authed user tweet and have it appear on a timeline.

2b)
Functional 1: Will adding a new comment and then deleting a tweet preserve the comment?
Functional 2: Will logging in as an authed user bring up that user's profile and timeline?

Non-Functional 1: Does the timeline load tweets within 3 seconds?

Non-Functional 2: Does the system run on both Chrome, Android, Apple, and Firefox?

3) https://github.com/cameron-mccawley/leap_year

4a) Creating a basic shopping app similar to etsy.

4b)

Testing in Agile	**Testing in Waterfall**
Testing purchase functionality as we implement the component, making changes as needed along the way.	Have testing the purchase functionality be a separate phase after the development team implements the component.
We would perform usability tests to the product with each component as we implement it, and have the developers keep the usability requirements in mind as they work on the system.	Usability tests would come after we implement most of the system. We would separate developers from the testers and only makes changes to the UI after the testing phase.