1 Specific Requirements

1. Software Requirements
    - 1.1. The frontend portion of the software should be built using a combination of an open source Javascript web framework and a CSS framework.
        - 1.1.1. The Javascript framework should be utilized for building interactive user interfaces and components such as the home page UI, notification filters, and notification cards
        - 1.1.2. The CSS framework should be used to style our application with more ease by allowing the use of prebuilt CSS instead of building new CSS from scratch. This cuts production time and is easy to use
    - 1.2. The backend portion of our software should be built using a combination of an open source runtime environment and web application framework.
        - 1.2.1. The runtime environment should be utilized for server-side programming by handling client requests and events.
        - 1.2.2. The web application framework will be used for server routing by controlling what webpage endpoint to display.
    - 1.3. The software should be hosted on a cloud application platform.
    - 1.4. Version control software should be utilized.
    - 1.5. A repository will be utilized to host and assist in codebase sharing.
2. Functionality Requirements
    - 2.1. Notifications will be displayed on the main page.
        - 2.1.1. Each notification will have basic notification data consisting of the course name where the notification is coming from, the notification header, the type of notification, the notification message, and the date it was uploaded.
            - 2.1.1.1. Types of notification are general announcement, important announcement, assignment, and project.
            - 2.1.1.2. Notification pop-ups will also adjust to different priority levels and types
        - 2.1.2. By default, unread notifications will be presented in descending order by date.
    - 2.2. There will be a breakdown of the count of notifications by course at the top of the page.
    - 2.3. Unread notifications can be changed to ‘Read’ status when the user acknowledges the notification
    - 2.4. Notifications may be filtered by course, type, specific dates, priority, unread, and read.
    - 2.5. There will be distinguishable styling for different types of notifications.
    - 2.6. Notifications can only be viewed if the user is part of the course.

2.7. Notifications will be created by professors.

2.7.1. A fillable form will be provided to the professor so all the professor needs to do is fill in the notification information.

2.7.2. Professors will be able to attach a file.

3. User Experience Requirements

3.1. The notification system should be practical.

3.1.1. Reading notifications and using tools should be self-explanatory to the user

3.2. The notification system should be reliable with no glitches that hinder user experience and UI elements shouldn’t cause lag

3.2.1. Components of the system should make sense and placements of the components should be natural as if the user expects a certain component to be in a certain location.

3.3. The notification system should be flexible.

3.3.1. Notifications will be readable and usable at any resolution size.

3.3.1.1. The webpage will scale depending on the resolution size of the device being used to view the notifications.

3.3.1.2. Users should still be able to find the notifications and understand what is happening on the page.

3.3.1.3. Users should be able to alter and interact with the notifications.

3.3.1.4. The notification popups should be 1-2 seconds short and not take up the whole screen.