

## **ECE 448/528 – Application Software Design**

### **HOMEWORK #3**

***Due date: Thursday, April 17th, 2025, 23:59 PM***

1. As we are utilizing JUnit to implement test cases for our course project, you already know that `@Test` annotation is used to declare a method as a test method. In many test cases that you are currently making or made for Projects, you may have found that a pre-defined or post-defined environment/condition would be necessary to implement a successful test cases. Investigate and explain which annotations can be used to pre-define and post-define test case environments. *(10 points)*
2. Consider our members application discussed in Lecture 20-24 *(15 points)*
  - i) After the user clicks the checkbox to add/remove a member to/from a group, we wait until the server backend replies to update the state and update the view. What would happen if it took a while for the server backend to reply and the user clicks the same checkbox multiple times impatiently?
  - ii) To address the potential issues in i), someone proposes to disable the checkbox somehow after the user clicks it and enable it again once the RESTful response to get all groups arrives. List all events involved in implementing this solution.
  - iii) The solution in ii) should still be implemented within the MVC pattern. Discuss the changes you need to make for the model, view, and controller. (Note that you don't need to implement it but need to discuss what needs to be added/modified)
3. In our course project, we are utilizing Gradle version 6.9.3 and Spring Boot version 1.5 whereas the up-to-date version of Gradle is version 8.7, and Spring Boot is 3.2.4. Identify the key necessary steps required to perform this upgrade by identifying the key difference between the versions, and what changes you need to make in the 'build.gradle'. Your answer does not need to include any implementation. *(15 points)*