

### **Current Testing Approaches**

We have been using White-Box Testing approaches:

1. Coverage Testing/Unit Testing: We have been using the JUnit framework to ensure that we are testing an acceptable percentage range of code. We will use unit testing and isolate units for separate testing. We will mock dependencies to minimise external interference.
2. Path Testing: We have tested different paths through the code to ensure that all paths/branches are tested.
3. Standard Code Review: We have had our team members explain to each other our respective codes to detect any abnormalities and/or gaps in logic.

We have also been using Unit Testing:

1. We are using the same framework for coverage testing as we have for unit testing (JUnit) to automate the testing process
2. Regression Testing: Making sure any new code pushed to the branches is not destroying the existing functionalities of the code

### **Future Testing Approaches:**

1. Integration/System Testing: We will ensure that interaction between all of the components in our system run smoothly and without errors, and we are employing system testing to test our system as whole after all the components of our system are integrated together to find any defects.
2. Acceptance Testing: After the completion of our system, we are using acceptance testing to make sure our project matches all specifications and goals. We will compare the performance and documentation with the instructions on Milestone 5.
3. We could also use Black-box testing on users to find any bugs in the final product. Users will not have any knowledge of the code and their evaluation will solely depend on their usage of our application.