Leigh Ann Stack

08/20/2025

CS320

Project Two

In project one we created 6 classes to meet the clients software requirements. In each class there were multiple variables created. We needed to make sure each variable had the proper checks in place to make sure it met the clients requirements. For example, in the contact class the variable called firstName, the client didn’t want to see more than 10 characters at a time. We used a else/if statement for the 10 character limit. (ex: else if (firstName.length() > 10).

JUnit testing was tough for me,but having it broken out over a few weeks certainly helped me understand the big picture. For the JUnit tests, for the contact class, we tested the inputs to make sure the user has inputted a correct userID, first and last name, phone number and address. Also tested were incorrect length of inputs, such as a phone number had exactly 10 characters. And null values of each variable were also tested.

I tried to add comments to my code to make sure the JUnit tests were technically sound. To make the code more efficient I tried to combine as much as I could and add comments where necessary. See paste below:

public void setLastName(String lastName) {

if (lastName == null || lastName.isEmpty()) {

this.lastName = "NULL";

} else if (lastName.length() > 10) {

this.lastName = lastName.substring(0, 10);

} else {

this.lastName = lastName;

}

There are a few techniques that could be used for software testing. I like running the code and using the debug button instead of static testing. I am not an expert in coding so visually looking through all of my code wouldn’t help me at this point. I tried to go through the requirements one by one and make sure the input and output was what the client was looking for. Security testing wasn’t used in this project. A dependency report like in Maven, can show you if the code shows any vulnerabilities.

This class has opened my eyes to software testing. Before this class I was naïve and thought the debug button was a catch all for testing. I have learned that short cuts usually doesn’t work in code writing.

I don’t think I was bias in the review of my code. I am simply not experienced enough to have bias. I can see that in the real world, that could really be an issue. I think having a peer review your code would help remove the bias. But also shows humility in the programmer. Caution is another trait that software programmers need. Sometimes you think you are making a fix in your code, and the whole thing blows up. I have learned to save your original file separately before making big changes. When big changes are made, its hard to pinpoint exactly where you went wrong. Caution with pushing big changes out in code can ensure you don’t have to start all over. One last thing I learned is that you need to know what the code should do and make sure I understood what the program was doing. These are two things I need to work on if I want to continue a career in coding after this degree.