Project Two

By: Onasis McCuien

This report is to elaborate on the testing techniques and approaches used to accomplish the customer requirements for this project. The costumer requested that a mobile application, and what was delivered via that request was contact, task, and appointment services. The code was essentially the ability to add and take away contacts, tasks, and appointments. To be able to accomplish such tasks, the code had to be written as such. The code allowed the option to add contacts, tasks, and appointments as needed. You also would have the option to remove those same tasks, contacts, and appointments as needed.

To apply Junit testing, the first thing that we need to do is import the needed imports to apply the Junit testing. Those would include imports like “import static org.junit.Assert.\*;”. This import will allow us to apply actual junit testing to the code. For example, if I wanted to test the line of code that is before a module, I would simply apply and “@Before” command on the line right before the module. When the junit tests are executed, your percentage tests will be displayed on the left side of your screen. This will you a percentage of how good and accurate your junit test was.

Graphical user interface, text, application

Description automatically generated

My joint testing was not accurate at all. But lets just imagine that my application of my junit testing was somewhat accurate. Let’s just say, that I set up my code accordingly, and it yielded me a zero defect, green line as shown above? This will indicate a technically sound set up. I will also demonstrate good, effective code writing within itself.

The type of testing that was administered in this code, was junit testing. Junit testing in the simplest form of testing as far as I am concerned. There is also maven testing, which is like the junit testing. There are different levels of testing. They can include Acceptance testing, System testing, Integration testing, and of course unit testing. Acceptance testing is a quality assurance process that determines to what degree an application meets end users’ approval. System testing, is the testing of a com;ete and fully integrated software product. This is like black box testing wherein knowledge of the inner design of the code is not a pre-requisite and is done by the testing team. Integration testing is integrating small portions of code and testing that code as you move through your code.

You must adopt the mindset of wanting to do something through your computer. Basically, you want the computer to output what you are thinking. For example, in my code, if I want to add a new contact, I want to add a new contact with that contacts information. If that new contact requires service, than I add what services that contact requires. I must bring this thought process to fruition through the code, in a neatly structured manner.

Let’s talk about bias. As a code developer, I must take an oath to never use my skill for wrongdoing. In my testing technique, I try to dedicate all my understanding and knowledge to develop what the customer as requested. Now, if it involves immoral or unethical practices, then I will not move forward with business from that customer. As a developer, I want to demonstrate results that produce a simpler life for the customer. I want the customer to be so pleased with our services and abilities that they come back for more, and also let the world know what type of services that we can offer.

In the world of today, everything is digital. There are no more rotary phones. There are no more pagers. There are cell phones, that are more powerful than laptop computers now a days. My manager has his whole life in his phone, lol. I could only imagine in the grid went completely out, how people would survive.

It is important to generate fully functional code, that is checked periodically during the development stage. It is not a good idea to try to complete code, and then try to check it. You will just make that job more time consuming and more expensive, because of all the bugs that you will have to fix at the end. Also, in this digital age, there are millions of people who are waiting to scam. It’s sad that a person can just sit behind a desk and literally rob you blind, rather than having to confront you on the street to do it. That is what we have come to today. Is advancement a curse? Or is advancement of the digital age a blessing? Ask yourself that question,….now let it dwell in your mind for a bit.

The Hiromi Satellite was a $268 million dollar wreck. It’s important to always check your code. Double, triple, and even quadruple check your code. Bugs do happen, even when you have fully functional code, it’s still possible for natural disasters to happen. But even with that, if you have good robust code, you can have a 95 percent confidence level with fully checked, and functional code.