Annamarie Cortes

CS 320 Software Test, Automation, QA

Module Seven

June 23, 2024

Project Two

Summary:

The mobile application requirements included services for contact, task, and appointment. Each service would have two classes and two Junit tests for each of the class. For instance, there are two classes for contact service which are named Contact and ContactService and it has the corresponding unit tests which are named ContactTest and ContactServiceTest. The following services follow the same format for appointment and task service. Each service has its own requirements such as it components cannot be null and have a certain length.



The requirements were implemented and each class are verified by the Junit tests.



The Junit test was written in a way that would pinpoint the error if the test were to fail. For example, there is a test written when the contact ID is null and another when the contact ID is too long. It also checks that when inputting the correct values, the test will verify that it passes. There is a Junit test for each class and it checks for that the correct values passes and each of the incorrect values such as being null, too long, etc. Making each test specific to an incorrect value allows us to pinpoint the error in the code and ensures that it is efficient and technically sound.

The overall Junit coverage of the mobile application is 83.9% which indicates that the code is well-tested and that potential bugs are caught early on. The Junit coverage was also ran for each services with contactService at 87%, taskService at 100%, and appointmentService at 100%. This demonstrates that the code is efficient and are sufficiently tested. With these numbers, it can be confidently said that the mobile application should behave as intended and handle errors when it encounters one.

A screenshot of a computer

Description automatically generated

Reflection:

The software testing technique used in this project is unit test and more specifically Junit testing. Unit testing is where individual components are tested in isolation. This can be seen with the Junit test created for this project. For instance the tests, testFirstNameNull and testLastNameTooLong is testing for specific use cases where the first name is null and when the last name is too long. When the test fails, we can pinpoint exactly what the error is and fix it. Unit tests helps detect errors early on and prevent this from creating a bigger damage to the overall project. Not to mention, avoiding errors such as this will save developers time, effort, and costs.

Other testing techniques that were not explored are integration testing, acceptance testing, system testing, regression testing, and many more. Each test have different characteristics and serves a different purpose. Integration testing focuses on the interactions between the different components and ensures that they behave as intended. It is typically done after unit testing and during the system integration phases when the developer combines the different components to build the overall system. For example, if system testing were to be done in our mobile applicate, the system test will verify that appointmentID, contactID, and taskID will interact with the database and other components correctly. Acceptance testing is typically done just before the software is released to the market. It is the final test verifying that the program meets the requirements defined by the client. In this mobile application, acceptance testing will go through all of the requirements and sure that they are implemented correctly. It will also go through end-to-end user scenarios and verify that the program is working as intended and look at the features in the user perspective. System testing is completed after integration testing and before acceptance testing It will involve testing the overall system with its integrated components. It will include functional testing which involves validating the requirements, performance testing which evaluates the performance, and more. Regression testing are typically performed when there is an update in the software and this is to ensure that the new update does not affect the current features and behavior of the software. If there was an update to our mobile application in which we add another service, regression testing will need to be performed to make sure that it does not disrupt the other services.

While developing this mobile application, it is important to proceed with caution and ensure that the code is both efficient and technically sound. In this program, we are dealing with sensitive user information such as their contact info, name, phone number, and address so it is important that the code does not have errors that would make it vulnerable. Having a technical sound code is another layer of protection from others that try to obtain sensitive information. It is also important to note the behaviors of the different components. It is important that components such as contactID, taskID, and appointmentID cannot be duplicated otherwise this will lead to errors when trying implement features such as updating contact information, task details, and appointment details. Thinking about the overall behavior of the system and what this program is trying to achieve is very important in order to make sure that relationships between components are applied correctly.

In any aspect work, there will always be a certain degree of bias so it is important to have another person to look over the code and ensure that it is tested thoroughly. Each individual think differently and there may be another scenario that I might not have thought of. In this project the requirements were laid out and we can effectively ensure that the requirements are being met but in real-world scenarios the requirements may not be as clear and it is up to the development team to identify the requirements to satisfy the client’s needs and wants. For instance, in this project another developer may look and think of another scenario that can make the system more robust such as adding time for appointmentService. Currently it only takes in the id, date, and description but for different tasks it may indicate different times and how long this would take. This would be a missing component that I and my team have overlooked.

Discipline is very important to ensure the quality of the program. No corners should be cut and all the necessary testing should be completed. Not doing the tests may implicate the customers and damage the company’s reputation. Also, correcting an error after the software has been pushed into the market will lead to more costs and customers potentially suing the company. There is also government regulations where it is the company’s responsibility to protect their customers’ information.