**Grand Strand Systems**



Lauren Jaime

CS320 Software Test QA

04/20/2025

**\*\*Appointment/Contact Application\*\***

**\*\*Summary Report\*\***

**Summary**

1. Moderate application testing approach into summary process reflection:
2. The Enterprise Data Management of this application required three storage feature testing requirements: #Contact Information #Appointment Status and #Task Services. The inventory testing requisites concluded that data was formatted per objective with JUnit4 and JUnit5 test parameters. Each method is isolated then stabilized with JUnit Testing confinement to measure accuracy and continuity during the lifetime of the program.
3. The quality of test units substantially improved the application during execution, including superior performance adaptability with user input. Background JUnit staging provided the actual percentage of accurate or failed application test runs per function as insight into the reliability, validation and achievement of the development process.
4. Define development process and code correspondence of JUnit test programming:
5. How did you ensure that your code was **technically sound**? JUnit4 and JUnit5 are built-in test components for java in eclipse IDEs to provide professional coding engines. Utilizing testing approaches of the code is an improved technical development in contrast to applications that don’t employ software testing. A breakdown of JUnit testing components is as follows:

**Grand Strand Systems**

@Test

public void testEditContact() {

ContactService editContactFirstName = new ContactService();

String contact\_output = editContactFirstName.editContactFirstName();

assertEquals("", contact\_output);

}

* @Tests is an identifier for JUnit to execute methods with the annotation.
* Classes define the functions with variables and qualifiers to test each method’s scope reliability.
* assertEquals method compares actual and expected values ("", contact\_output) and marks that specific test as pass or fail.

By applying each method and distributing test cases across the application, the final product’s value is improved by Whitebox testing utilities.

1. A
2. How did you ensure that your code was **efficient**? **Grand Strand Systems** designs and implements high quality development techniques and application performance for customers across the globe. Our guarantee provides each product with values and agreements before development to ensure customer satisfaction. The Appointment System Application’s efficiency is defined by reliable accuracy during runtime and product proficiency measured by JUnit testing. The final method in Test Files are executable result outputs for the developer to verify results. For example,

org.junit.runner.Result result = JUnitCore.runClasses(AppointmentTest.class);

for (Failure failure : result.getFailures()) {

System.out.println(failure.toString());

}

System.out.println(result.wasSuccessful());

}

The code above exemplifies how the AppointmentTest.class will gather test results and print the defined result method to the console.

**Grand Strand Systems**

**Reflection**

1. **Testing Techniques**
2. The software testing techniques employed for this application included Unit Testing and Acceptance Testing. Unit testing is the core concept for this project. This method of tests improves performance, identifies code malfunction early in development phases and verifies which methods in the application require additional code maintenance throughout the process. Acceptance testing is performed after the application is complete and verified by development testers or end users. This is the final measure of performance, reliability, functionality and that the application meets business requirements.
3. Additional software testing techniques available for applications include:

* Regression Testing – Double standard of unit testing to ensure reliability after code newer changes or updates.
* Performance Testing - “Performance testing aims to identify bottlenecks, measure system performance under various loads and conditions, and ensure that the system can handle the expected number of users or transactions.” (Geeks4Geeks, 2024)
* Security Testing - This method scans for malware, risk assessment and vulnerabilities among numerous other defenses at attempts to breach the software’s security reliance.

1. B
2. For each of the techniques you discussed, explain the **practical uses and implications** for different software development projects and situations. Unit testing and performance testing are great methods to provide user agreements to the customer that the application is stable and functions properly. Regression, performance and security testing are additional measures to provide comprehensive test procedures that improve performance and security.
3. a
4. **Mindset**
5. Assess the mindset that you adopted working on this project. In acting as a software tester, to what extent did you employ **caution**? Why was it important to appreciate the complexity and interrelationships of the code you were testing? Provide specific examples to illustrate your claims.

This application had specified requirements that made method testing much easier to deploy because of the provided requirements and caution when verifying methods in

**Grand Strand Systems**

the application. Verification of the Date function utilized the .before method after parsing user string input into a java date utility. That created a professional aspect of the application to verify business requirements and accountability on an as-is basis.

1. Assess the ways you tried to limit **bias** in your review of the code. On the software developer side, can you imagine that bias would be a concern if you were responsible for testing your own code? Provide specific examples to illustrate your claims.

Aspects of bias are not present because of the product requirements for this software. Fields had limited specifications and were equal, yet unique to the end user. There are no indications of a bias throughout numbers and letters unless vision accessibility requirements are utilized for some users. Text to speech applications is often helpful in that situation as well. Developing test methods to verify input length is a great example to provide users with a chance to update or verify information. Creating tests based on a null or specific length helps the code run uniformly and stay within specified boundaries for the application.

1. Finally, evaluate the importance of being **disciplined** in your commitment to quality as a software engineering professional. Why is it important not to cut corners when it comes to writing or testing code? How do you plan to avoid technical debt as a practitioner in the field? Provide specific examples to illustrate your claims.

Discipline in coding consists of uniform code, comments for other developers and also maintaining structure during formatting for readable applications. Methods are clear and concise, manufacturing coverage results more effectively and measure the performance of using testing techniques for exhaustive testing procurement. Thorough testing provides the best advancement in software development for writers/ customers since technology advances often implicate the application’s performance. The task name field is an example of the importance of testing and discipline because it has parameters and induces best practices per method.

**References**

GeeksforGeeks. (2024, September 26). *Performance testing - software testing*. <https://www.geeksforgeeks.org/performance-testing-software-testing/>