6CCS3SMT & 7CCSMASE Software Measurement and Testing

Coursework Overview

The coursework accounts for 15% of the final assessment and is an individual project divided into two equally important components: the test code and the report. The primary objective is to evaluate your reasoning when making decisions on the testing of a given piece of software and your ability to use the taught technologies such as JUnit and Mockito (or equivalent frameworks in Python) to write test cases. The source code for the software system under test is provided in both Java and Python, so you can use the selected technologies for any of these two programming languages.

The Task

You are provided with the source code of the software and the list of the requirements for it. The provided software contains faults, the number and locations in code of which will not be disclosed.

You are required to write a report that explains your approach to testing the provided software. In your report, describe the types of testing you implemented based on what you've learned in class, and specify which parts of the code you prioritised for testing along with your reasoning. Discuss any trade-offs you encountered, such as time constraints or resource limitations, and reflect on the challenges you faced during the testing process, including how you addressed these challenges. Document any faults you detect during testing and explain how they were identified through your test cases. The report should be approximately 2000 words.

In addition to the report, you need to write test cases that effectively test the functionality of the software. Focus on the quality and readability of your test code rather than solely aiming for full coverage. Generate a coverage report using a coverage tool of your choice and include it in your report. Use the provided system requirements to guide the design of your test inputs and expected outputs, and be sure to incorporate unit tests with mocks where necessary to isolate components of the software.

Marking

1. **Test Inputs/Outputs (5 Points)**

You need to provide comprehensive test inputs and expected outputs that effectively validate the specified requirements of the Shopping Cart system.

The performance indicators for this part are as follows:

- Inputs and outputs are clearly aligned with the system requirements.
- A variety of scenarios are tested, including edge cases and normal use cases.
- The rationale for selecting specific test cases is well articulated and logical.

2. **Unit Tests and Mocks (5 Points)**
   You should design unit tests that demonstrate a clear understanding of the software's functionality and incorporate mocks where necessary to isolate components.
   The performance indicators for this part are as follows:
   - Unit tests cover a significant portion of the functionality of the code and a test coverage report is provided.
   - Mocks are used effectively to isolate dependencies, enhancing the reliability of tests.
   - Tests are well-organized, readable, and follow best practices for testing.
3. **Fault Detection and Reporting (5 Points)**
   You should successfully identify and report any faults in the system, demonstrating the effectiveness of their test cases in detecting issues.
   The performance indicators for this part are as follows:
   - The percentage of detected faults out of all present in the system.
   - Detected faults are clearly documented, with explanations of how they were identified through testing.

Software System under Test

The Shopping Cart system is an online shopping platform designed to facilitate a seamless purchasing experience for customers. It enables users to browse and add products to their cart while providing robust functionality to calculate the total cost accurately. The system incorporates a variety of discount mechanisms, including tiered discounts based on cart value, customer-specific pricing based on classification (Regular, Premium, or VIP), and bundle discounts for specific product combinations. Additionally, it supports time-limited promotions and the use of coupon codes for further savings.

List of system requirements:

1. The system shall allow users to browse and select products to add to a shopping cart.

2. The system shall calculate the total price of items in the shopping cart before any discounts.
3. The system shall provide the ability to apply additional bundle discounts (5% off a mouse price if a laptop is also in the cart).
4. The system shall apply tiered discounts based on the total cart value, with predefined discount levels (10% for over £1,000, 15% for over £5,000, and 20% for over £10,000).
5. The system shall categorize customers into three types: Regular, Premium, and VIP, with Premium customers receiving an additional 5% discount and VIP customers receiving an additional 10% discount on their total.
6. The system shall allow users to enter coupon codes, which can provide additional percentage-based or fixed-amount discounts (10% off with code "DISCOUNT10" or £50 off with code "SAVE50"). Only one coupon code can be applied and the £50 off will be applied before any percentage discounts.
7. The system shall support time-limited promotions that can be activated or deactivated to apply a flat discount (25% off during a promotional event).
8. The discounts listed in points 3 to 7 are applied on top of each other in the order they have been specified.
9. The system shall print a detailed receipt summarizing the items in the cart, the total price before discounts, and the final price after all applicable discounts.
10. The system shall display an error message if the credit card number does not meet the 16-digit requirement or if the transaction amount is zero or negative, preventing the transaction from proceeding.

Example Expected Behaviour:

To explain further how the discounts are stacked let’s consider an example scenario. We have a VIP user with a shopping cart of one laptop (costs £1000) and one mouse (costs £100). The customer also has a coupon code of “SAVE50” and it is a time-limited promotion period. First, the 5% bundle discount will be applied to the customer’s shopping cart resulting in the overall amount of £1095. Then, the customer will have £50 off (due to coupon code of SAVE50) before any discounts and the amount will become £1045. Lastly, an overall discount of 45% (10% for a shopping cart over 1000 + 10% for being VIP + 25% for promotion time) will be applied altogether and the final amount the customer pays will be £602.25.