Assignment 2
Task 6

Author Details
Zoe Yow Cui Yi, 33214476

Bug Report 1
Description:
In “business_logic.py”, the method “can_borrow_carpentry_tools()”: Patrons aged 18 is not allowed to borrow Carpentry tools if they have completed the training and do not have any outstanding fees. However, AAL’s series of custom rules in BAT states that any patron under the age of 18 (<=17) cannot borrow carpentry tools. This means patrons aged 18 and over but under 90 can borrow carpentry tools if they have completed the necessary training and do not have any outstanding fees.

Steps to reproduce:
Using the redacted version of BAT:

1. Create a new test file in the ‘tests’ folder
2. Import “unittest”
3. Import the method “can_borrow_carpentry_tools” from “business_logic.py” file in the “src” folder.
4. Create a class to include all test cases for this method using “unittest.TestCase”
5. Create a test method using appropriate naming conventions.
6. Use “self.assertTrue()” to test that “can_borrow_carpentry_tools” will return True if a patron aged 18 with necessary training completed and have no outstanding fees wants to borrow Carpentry Tools.
7. In the function “can_borrow_carpentry_tools”, add the following requirements, “patron_age” = 18, “length_of_loan” = (<14), “outstanding_fees” = 0, and “carpentry_tool_training” = True.
8. Run the test file by running “python -m unittest -v” in the terminal which will run all test files.
9. Check the terminal for the output to see if this test method has passed or failed.

Expected behaviour:

1. Patrons aged 18 is allowed to borrow Carpentry tools if they have completed the training and do not have any outstanding fees.
2. The “can_borrow_carpentry_tools” method returns True if a patron aged 18 with necessary training completed and have no outstanding fees wants to borrow Carpentry Tools.

3. The terminal output shows that this test case has passed with the word “ok” at the end.

Actual behaviour:

1. The “can_borrow_carpentry_tools” method returned False if a patron aged 18 with necessary training completed and have no outstanding fees wants to borrow Carpentry Tools.
2. The terminal output shows that this test case has failed with an error to indicate it.

Bug Report 2

Description:

In “business_logic.py”, the method “calculate_discount()”: Patrons aged 50 and over (>=50) but under 65 receive a 10% discount on their outstanding fee. However, AAL’s series of custom rules in BAT states that any patron over the age of 50 (>=51) but under the age of 65 gets a 10% discount on their fees, rounded to the nearest cent.

Steps to reproduce:

Using the redacted version of BAT:

1. Create a new test file in the ‘tests’ folder
2. Import “unittest”
3. Import the method “calculate_discount” from “business_logic.py” file in the “src” folder.
4. Create a class to include all test cases for this method using “unittest.TestCase”
5. Create a test method using appropriate naming conventions.
6. Use “self.assertEqual()” to test that “calculate_discount” will return 0 when calculating discount for a patron aged 50.
7. In the function “calculate_discount”, add the requirement, “patron_age” = 50.
8. Run the test file by running “python -m unittest -v” in the terminal which will run all test files.
9. Check the terminal for the output to see if this test method has passed or failed.

Expected behaviour:

1. Patrons aged 50 do not receive any discounts on their fees.
2. The “calculate_discount” method returns 0 when calculating discount for a patron aged 50.
3. The terminal output shows that this test case has passed with the word “ok” at the end.

Actual behaviour:

1. Patrons aged 50 receives a 10% discount on their fees.

2. The “calculate_discount” method returns 10 when calculating discount for a patron aged 50.
3. The terminal output shows that this test case has failed with an error to indicate it.

Bug Report 3

Description:
When searching for a patron by their name (“Search by name”) in BAT, it is case sensitive.

Steps to reproduce:
Using the redacted version of BAT and assuming you are in the correct working directory:

1. Open the terminal
2. Run BAT using “python run.py”
3. Type “3” as your input to get into the “Search for Patron” screen.
4. Type “1” as your input to search for a patron by their name.
5. Type “john doe” as your input, as you want to find an existing patron with the name “john doe”
6. Check if there is a patron with the name “john doe” displayed in the terminal.

Expected behaviour:

1. Shows details of existing patrons with the name “john doe” in the terminal.

Actual behaviour:

1. Shows “NO PATRON FOUND MATCHING SEARCH DATA” in the terminal.