CISC 327 A3 (Fall 2025)

Eron Chung (20393151)

Date: Nov 10, 2025

Repository: https://github.com/EronChung/cisc327-library-management-a2-3151

Stubbing vs Mocking

A stub is a fake replacement value used in place of a real one provided/returned by actual functionality, and is set with a value relevant to the test case it's created in; they are essentially hardcoded values/constants. A mock is a fake representation of an entire object that simulates the behavior of the real object's functionality without actually engaging with the real implementation, which may not be available (such as with the payment gateway here); it's similar to a stub in that it returns fake replacement data, but has the ability to assert that passed in input params are actually valid, as well as check if a function was or was not called.

pay_late_fees

- `calculate_late_fee_for_book` was stubbed in each case to provide the necessary `fee_info`s.
- `PaymentGateway` was mocked in each case as the gateway is part of what is being tested.
    - Mocking it over stubbing also allows asserting if a gateway function was (not) called.

refund_late_fee_payment

- `PaymentGateway` was mocked in each case as the gateway is part of what is being tested.
    - Mocking it over stubbing also allows asserting if a gateway function was (not) called.

Test Execution Instructions

- Clone repository
- Run `python -m venv venv` to create virtual environment
- Run `source venv/bin/activate` in Linux, or `venv\Scripts\activate` on Windows
- Run `pip install -r requirements.txt` to install dependencies

New Test Cases

Name	Purpose	Stubs	Mocks
test_successful_payment	Test a successful payment.	calculate_late_fee_for_book	PaymentGateway
test_payment_declined_by_gateway_limit_exceeded	Test getting declined by payment service (limit exceeded).	calculate_late_fee_for_book	PaymentGateway
test_payment_invalid_patron	Test invalid patron ID.	calculate_late_fee_for_book	PaymentGateway
test_payment_zero_late_fees	Test with no late fees.	calculate_late_fee_for_book	PaymentGateway
test_payment_network_error	Test gateway network error exception.	calculate_late_fee_for_book	PaymentGateway
---	---	---	---
test_successful_refund	Test a successful refund.		PaymentGateway
test_refund_invalid_id	Test an invalid transaction ID.		PaymentGateway
test_refund_amount_negative	Test a negative refund amount.		PaymentGateway
test_refund_amount_zero	Test a refund amount of 0.		PaymentGateway
test_refund_amount_exceeds_15	Test a refund amount over 15.		PaymentGateway
---	---	---	---
test_add_book_invalid_duplicate_isbn	Test adding a book with an ISBN already in use (input same as first valid test).		
test_add_book_database_error	Test adding a book with an ISBN already in use (input same as first valid test).	insert_book	
---	---	---	---
test_borrow_book_invalid_no_copies	Test borrowing a book with no more copies left.	get_book_by_id	
test_borrow_book_invalid_borrow_limit	Test database error when adding a book.	get_patron_borrow_count	

Coverage Analysis

These branches were covered to reach 81%:

add_book_to_catalog

```
49      # Check for duplicate ISBN
50      existing = get_book_by_isbn(isbn)
51      if existing:
52          return False, "A book with this ISBN already exists."
53
54      # Insert new book
55      success = insert_book(title.strip(), author.strip(), isbn, total_copies, total_copies)
56      if success:
57          return True, f'Book "{title.strip()}" has been successfully added to the catalog.'
58      else:
59          return False, "Database error occurred while adding the book."
```

borrow_book_by_patron

```
81
82      if book['available_copies'] <= 0:
83          return False, "This book is currently not available."
84
85      # Check patron's current borrowed books count
86      current_borrowed = get_patron_borrow_count(patron_id)
87
88      if current_borrowed > 5:
89          return False, "You have reached the maximum borrowing limit of 5 books."
```

Negative test cases were written to reach these branches.
In order to not have to manipulate the database for testing as was done before, some functions were stubbed:

- `insert_book`: Stubbed to simulate a database error
- `get_book_by_id`: Stubbed to simulate having no copies left
- `get_patron_borrow_count`: Stubbed to simulate the patron having reached the borrow limit

Screenshots

```
(venv) PS C:\Users\Eron\Documents\__Dev\Python\CISC327\CISC327-CMPE327-F25> pytest -s
============================= test session starts =============================
platform win32 -- Python 3.12.0, pytest-7.4.2, pluggy-1.6.0
rootdir: C:\Users\Eron\Documents\__Dev\Python\CISC327\CISC327-CMPE327-F25
plugins: cov-7.0.0, mock-3.15.1
collected 54 items

tests\test_add_book_to_catalog.py ..........
tests\test_borrow_book_by_patron.py ........
tests\test_calculate_late_fee_for_book.py .......
tests\test_get_patron_status_report.py .....
tests\test_pay_late_fees.py .....
tests\test_refund_late_fee_payment.py .....
tests\test_return_book_by_patron.py .......
tests\test_search_books_in_catalog.py ......

============================== 54 passed in 0.50s ==============================
```

```
(venv) PS C:\Users\Eron\Documents\__Dev\Python\CISC327\CISC327-CMPE327-F25> pytest --cov=services --cov-report=html --cov-report=term tests/
---------- coverage: platform win32, python 3.12.0-final-0 ----------

Name                          Stmts   Miss  Cover
-------------------------------------------------
services\library_service.py     160     14    91%
services\payment_service.py      30     22    27%
-------------------------------------------------
TOTAL                           190     36    81%
Coverage HTML written to dir htmlcov
```