CSCI 3060U - Phase 1

Brandon - 100912514
Jason - 100921022,
Moksh - 100916422
Richard - 100914490

Overview

Project: Banking Application - Bankify

Objective: Develop a two-part banking application using the console, composed of frontend features simulating an ATM for file transactions and backend features handling the master banking account files.

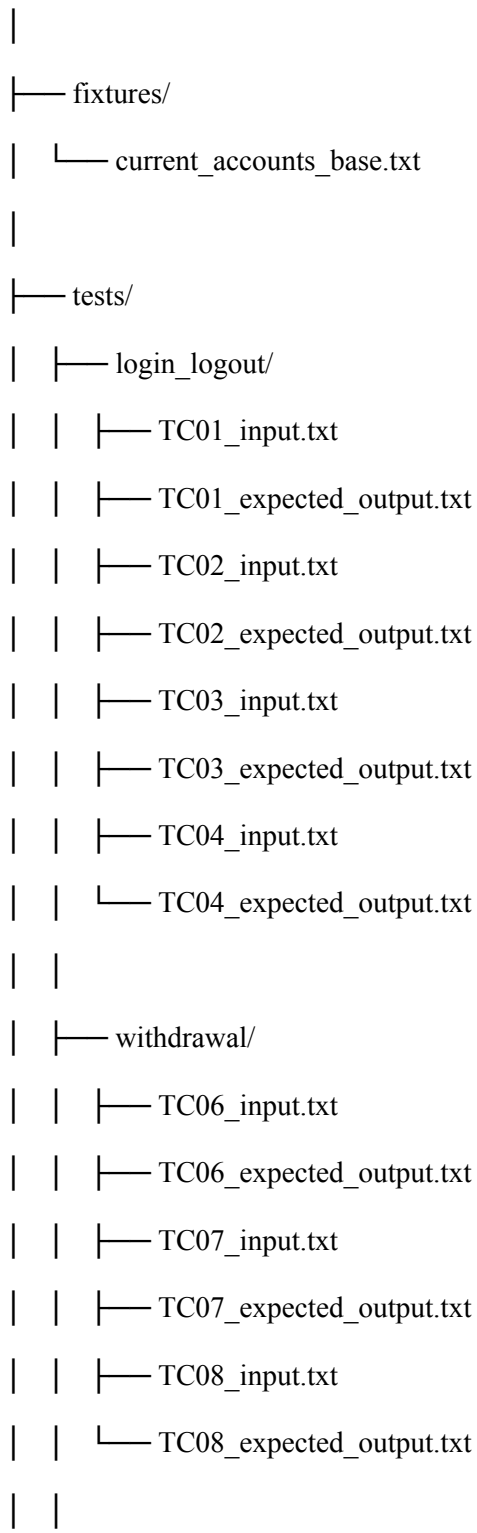
Scope: The project is a console-based Python application that operates via the terminal and text file input and output.

The application's **frontend** component mimics an ATM, using transaction input and output files. It also supports user login/logout, withdrawal, transfer, bill payment, deposit, account creation/deletion/disabling, and account plan changes.

Link to Github: https://github.com/brandonviaje/Bankify

Test Organization

banking_phase1/



```
│   ├── transfer/
│   │   ├── TC10_input.txt
│   │   ├── TC10_expected_output.txt
│   │   ├── TC11_input.txt
│   │   └── TC11_expected_output.txt
│   │
│   ├── paybill/
│   │   ├── TC14_input.txt
│   │   ├── TC14_expected_output.txt
│   │   ├── TC15_input.txt
│   │   └── TC15_expected_output.txt
│   │
│   ├── deposit/
│   │   ├── TC17_input.txt
│   │   └── TC17_expected_output.txt
│   │
│   ├── privileged/
│   │   ├── TC18_input.txt
│   │   ├── TC18_expected_output.txt
│   │   ├── TC19_input.txt
│   │   ├── TC19_expected_output.txt
│   │   ├── TC20_input.txt
│   │   └── TC20_expected_output.txt
│   │
│   └── validation/
│       ├── TC24_input.txt
```

```
│      └── TC24_expected_output.txt
│
├── README.txt
│
└── test_plan/
    └── Phase1_Test_Plan.pdf
```

Test Execution

All test cases are executed using pytest to automate the testing for the banking application. Tests can be run individually or as a complete suite from the terminal, which returns pass/fail status, and the generated output files are stored in directories used for comparison with expected outputs.

Test Case Comparison & Problems

- There is no requirement for creating admin accounts. All admin transactions assume the admin user exists.
- The requirements do not explicitly state that admins must provide the account number when performing transactions, which could lead to inconsistent system behaviour or test cases.
- The requirements state that the system must “gracefully handle bad input,” but there are no clear acceptance criteria. Does invalid input cancel a transaction or retry it?
- Disabled vs deleted accounts: Both reject further transactions, but differences (like recreation) are not defined (i.e Can a deleted account be re-created with the same account number and/or name? Can a disabled account be re-enabled and used in the same session?).
- The requirements do not say whether admin accounts exist beforehand or can be created; test setup is unclear.
- Admin transactions require account holder name, but it’s not explicitly stated that the admin must also provide the account number; could lead to inconsistent test behavior.
- The spec does not mention what happens if multiple sessions try to access the same accounts simultaneously.