Laboratory Project №2 – Unit Testing Report

Name: Emon Ashraful

Group: [972205]

Project: PersonalFinanceManagement - Lab 2 Unit Testing

# Summary

This report presents unit tests created for flaws identified in Laboratory Project №1. Each test covers a specific bad code practice such as Long Method, Primitive Obsession, or Data Clumps, and verifies correctness after improvement or refactoring. All tests are green (passing) and validated through xUnit.

# Test Descriptions

|  |  |  |  |
| --- | --- | --- | --- |
| Shortcoming (Flaw) | Test Description | Example | Test Method Name |
| Long Method in Log\_in() | Checks ReadInt returns correct integer for valid input | Input '3' -> returns 3 | ReadInt\_ValidInput\_ReturnsCorrectInteger |
| Long Method in Log\_in() | Handles invalid input followed by valid integer | Input 'abc' then '5' -> returns 5 | ReadInt\_InvalidThenValidInput\_ReturnsValidInteger |
| Long Method in Log\_in() | Verifies ReadDouble returns correct double | Input '12.5' -> returns 12.5 | ReadDouble\_ValidInput\_ReturnsCorrectDouble |
| Long Method in Log\_in() | Handles invalid double input then valid | Input 'wrong' then '7.7' -> returns 7.7 | ReadDouble\_InvalidThenValid\_ReturnsCorrect |
| Long Method in Log\_in() | Checks ReadString returns input string | Input 'wallet1' -> returns 'wallet1' | ReadString\_ReturnsInputString |
| Primitive Obsession in Password Handling | Verifies same input gives same hash | Input 'mypwd' twice -> same hash | HashPassword\_SameInput\_ReturnsSameHash |
| Primitive Obsession in Password Handling | Checks different input returns different hashes | 'pass1' vs 'pass2' -> different hashes | HashPassword\_DifferentInput\_ReturnsDifferentHash |
| Primitive Obsession in Password Handling | Ensures password is hashed not stored | Input 'mysecret' -> not 'mysecret' | Password\_IsHashed\_NotStoredAsPlainText |
| Long Method in Log\_in() | Valid credentials return user | User with correct password -> success | TryAuthenticate\_ValidCredentials\_ReturnsUser |
| Long Method in Log\_in() | Non-existing email returns null | Email 'abc@x.com' -> null | TryAuthenticate\_NonExistingEmail\_ReturnsNull |
| Long Method in Log\_in() | Wrong password returns null | Correct email but wrong password -> null | TryAuthenticate\_InvalidPassword\_ReturnsNull |
| Data Clumps in Wallet.RemoveWallet | Removes existing wallet | 'TestWallet1' exists -> removed | RemoveWallet\_ExistingWallet\_Removed |
| Data Clumps in Wallet.RemoveWallet | Handles non-existing wallet | 'MissingWallet' -> nothing removed | RemoveWallet\_NonExistingWallet\_NoEffect |
| Data Clumps in Wallet.RemoveWallet | Handles null wallet gracefully | Null wallet -> exception or handled | RemoveWallet\_NullInput\_HandledGracefully |
| Primitive Obsession in Wallet.AddIncome | Valid input increases balance | AddIncome 100 -> balance +100 | AddIncome\_ValidInput\_IncreasesBalance |
| Primitive Obsession in Wallet.AddIncome | Negative amount throws exception | -50 -> exception | AddIncome\_NegativeAmount\_ThrowsArgumentException |
| Primitive Obsession in Wallet.AddIncome | Empty description throws exception | '' -> exception | AddIncome\_EmptyDescription\_ThrowsArgumentException |
| Primitive Obsession in Wallet.AddExpense | Valid input adds expense | AddExpense 50 -> added | AddExpense\_ValidInput\_AddsExpense |
| Primitive Obsession in Wallet.AddExpense | Negative amount throws exception | -10 -> exception | AddExpense\_NegativeAmount\_ThrowsArgumentException |
| Primitive Obsession in Wallet.AddExpense | Empty description throws exception | '' -> exception | AddExpense\_EmptyDescription\_ThrowsArgumentException |