

# Steam 2 - Test Plan Document

## Test Organization

The tests are organized within the `Project/Test Cases` directory, structured to reflect different functionalities of the system. Each directory contains `.in` and `.out` files representing input and expected output for each test case, respectively.

### Directory Structure

```
Project/
|
├── Scripts/
|   ├── run_tests.ps1
|   └── generate_reports.ps1
|
└── Test Cases/
    ├── 1.Login/
    |   ├── TestCase1.in
    |   ├── TestCase1.out
    |   └── ...
    |
    ├── 2.Logout/
    ├── 3.Create/
    ├── 4.Delete/
    ├── 5.Sell/
    ├── 6.Buy/
    ├── 7.Refund/
    └── 8.AddCredit/
```

## Test Execution

Tests will be run using PowerShell scripts on Windows, leveraging the existing project structure. The primary script, `run_tests.ps1`, will iterate through each directory within `Project/Test Cases`, executing the application with inputs from `.in` files and comparing the application's output with the corresponding `.out` files.

## Execution Methodology

1. **Running Tests:** The `run_tests.ps1` script will be executed from the `Project/Scripts` directory. This script automates the process of running each test case, capturing the output, and comparing it with the expected output.
2. **Capturing Outputs:** For each test case, the actual output generated by the application will be saved in a temporary file within the same directory as the test case. This file will be compared against the expected output file.
3. **Reporting:** Upon completion of all test cases, the script will generate a summary report indicating the success or failure of each test case. Detailed results, including any discrepancies between actual and expected outputs, will be logged for further analysis.

## Output Storage and Organization

The output from each test run, including summaries and detailed logs, will be stored in a designated directory within `Project/Scripts/Reports`. This allows for easy access to historical test results for comparison and trend analysis over time.

### Directory Structure for Reports

```
Project/
|
|— Scripts/
|   |— Reports/
|       |— Report_YYYYMMDD_HHMM.txt
|       |— ...
|   ...
```

### Reporting Methodology

- **Report Generation:** After running tests, the `generate_reports.ps1` script will compile the outcomes into a report, including the date and time of the test run, the number of tests passed/failed, and any failures highlighted for review.
- **Report Storage:** Each report will be saved with a unique name reflecting the run's date and time, ensuring that historical data is preserved and easily accessible.
- **Comparison and Analysis:** For ongoing development and quality assurance, teams can compare current reports with previous ones to track improvements and identify recurring issues.