

Phase #1 - Front End Requirements (CSCI 3060U)

Group Members:

- Aliseena Ahmar (100870078)
- Haider Saleem (100870637)
- Samir Ahmadi (100916280)
- Smit Kalathia (100871659)

Test Plan

Purpose

The purpose of this test plan is to define how the Front End requirements of the Banking System are validated in Phase #1.

This phase focuses on analyzing the requirements and representing them as a complete set of requirements-based tests, without implementing the system.

Each test case consists of:

- A test input file that simulates user interaction
- A corresponding expected output file that describes the correct observable behaviour

These test cases will be reused in later phases for automated and regression testing.

Scope of Testing

The test cases cover the following Front End requirements:

- Session handling (login and logout)
- Standard and administrative session modes
- Enforcement of privileged transaction restrictions
- Standard transactions:
 - Withdrawal
 - Transfer
 - Deposit
 - Paybill
- Administrative transactions:
 - Create

- Delete
 - Disable
 - Changeplan
- Constraint enforcement:
 - Account ownership and existence
 - Account status (active, disabled, deleted)
 - Session transaction limits
 - Invalid input handling

Test Case Structure

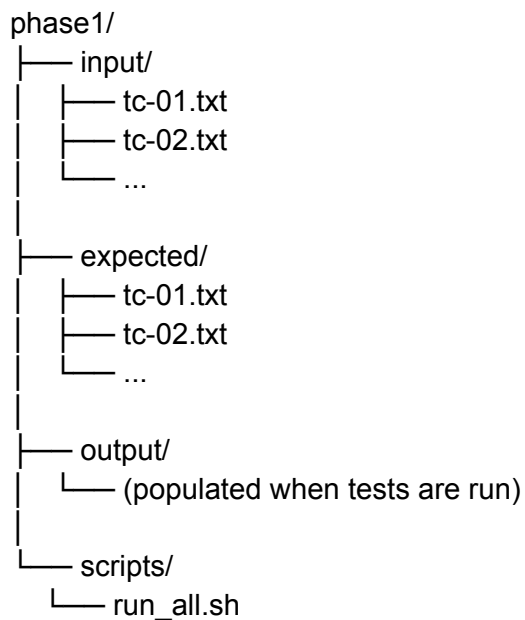
Each test case is identified by a unique test ID (TC-XX) and is represented by two files:

- Input file: tc-xx.txt (located in the input/ directory)
- Expected output file: tc-xx.txt (located in the expected/ directory)

The input file contains the exact sequence of user inputs, while the expected output file describes the correct behaviour of the system for that input.

Test Input and Output Organization

Test inputs, expected outputs, and actual outputs are organized into separate directories to maintain clarity and allow automated testing.



- The input/ directory contains all test input files.
- The expected/ directory contains the expected outputs corresponding to each test.
- The output/ directory stores actual outputs generated when the tests are executed.
- File names are identical across directories to allow direct comparison.

Expected Output Definition

Expected output files describe the observable behaviour of the Front End, including:

- Whether a command is accepted or rejected
- Session state changes
- Error conditions and constraint violations

Exact wording of console messages is not specified in the requirements, so expected outputs focus on behavioural correctness rather than formatting.

Validation of transaction file contents is intentionally deferred to later phases.

All test cases are designed to be executable automatically once the Front End program is implemented.

Each test is run by redirecting its input file into the Front End program and capturing the program's console output. The captured output is then compared against the corresponding expected output file.

Automation Scripts

Two shell scripts are provided to support test execution:

run_one.sh

Executes a single test case specified by test ID. This script is used to run and debug individual tests without executing the entire test suite.

- Reads the test input from input/tc-xx.txt
- Captures the program output in output/tc-xx.txt

run_all.sh

Executes all test cases sequentially by iterating through the input/ directory.

- Runs every test input file
- Stores each result in the output/ directory using the same file name

Output Comparison and Reporting

Test results are validated by comparing the actual output files in the output/ directory against the corresponding expected output files in the expected/ directory.

Comparison is performed using standard text comparison tools. Matching files indicate a passing test case, while differences indicate a failure. This allows straightforward identification of incorrect behaviour and supports regression testing in later phases.

Requirements Issues Identified

During test design, the following ambiguities were identified in the project requirements:

- It is unclear whether per-session transaction limits apply cumulatively or per transaction.
- The requirements do not specify whether account numbers for newly created accounts are user-entered or system-generated.
- The expected behaviour of the changeplan transaction for accounts already on a non-student plan is not defined.
- The exact wording and format of console output messages are not specified.

These issues have been documented but do not prevent requirements-level testing.