**Test Summary Report Template**

Table of Contents

[1. Introduction 3](#_Toc186449260)

[1.1 Background 3](#_Toc186449261)

[1.2 Structure of the Report 3](#_Toc186449262)

[1.3 References 3](#_Toc186449263)

[2. Overview 3](#_Toc186449264)

[3. Adjustments 4](#_Toc186449265)

[4. Assessment 4](#_Toc186449266)

[5. Results 4](#_Toc186449267)

[6. Evaluation 4](#_Toc186449268)

[7. Summary of Activities 5](#_Toc186449269)

# Introduction

## 1.1 Background

This document summarizes the complete testing process for the logistics and delivery system project. The system models a city map, assigns truck routes, and finds the shortest paths for delivery while avoiding buildings. Testing includes unit, integration, whitebox, and acceptance tests to verify correctness, edge case handling, and system behavior.   
**Testing Tool:** CppUnitTestFramework in Visual Studio

## 1.2 Structure of the Report

This report is structured in the following manner:

* Section 2 provides a testing overview.
* Section 3 outlines adjustments made during development.
* Section 4 evaluates test coverage.
* Section 5 reports results and bug resolutions.
* Section 6 contains final evaluation.
* Section 7 summarizes testing activities and participation.

## 1.3 References

* main.c, mapping.c, mapping.h: Original implementation files
* UnitTest1.cpp: Complete test cases
* traceability-matrix-MS6.xlsx: Requirement test mapping
* Jira Board: Issue tracking
* GitHub Repository: Project collaboration and commits

# Overview

This section provides a high-level overview of the testing activities conducted across the Seneca Polytechnic Deliveries project, from initial unit testing in Milestone 3 to final acceptance testing in Milestone 6.

**Test Categories:**

* Unit Tests: Validate function behavior and output correctness
* Integration Tests: Assess map + route interaction and overlay updates
* Whitebox Tests: Probe internal control flow of shortest path algorithms
* Acceptance Tests: Mapped to 6 official system requirements

**Environment:**

* OS: Windows 10
* Compiler: Visual Studio 2022
* Language: C
* Framework: CppUnitTestFramework
* Data:
  + Map: 25x25 grid generated by populateMap with predefined buildings.
  + Routes: Blue, Green, Yellow routes from getBlueRoute, getGreenRoute, getYellowRoute.
  + Shipments: Weights (0–10000 kg), volumes (0.4–5.1 m³), destinations (e.g., “2B”, “26Z”).
  + Test cases: Defined in test-cases-template.xlsx, executed via UnitTest1.cpp.

# Adjustments

The testing process largely adhered to the initial test plan, with one notable adjustment. Adjustment: Additional debugging and code modification for calculateDiversionPath in Milestone 6.

* Details: Test CD-02 (destination off the route) initially failed because the function returned an empty route for adjacent points (e.g., destination {4,10} from route point {4,9}). The test plan assumed all unit tests would pass by Milestone 5, but this edge case required a fix.
* Reason: The original implementation of calculateDiversionPath did not account for single-step diversions, relying solely on shortestPath for multi-step paths.
* Fix Applied: Modified calculateDiversionPath to check for adjacency (Euclidean distance ≈ 1) and return a single-point route if applicable (see Section 5 for code).
* Supporting Documentation: Jira issue #XXX [Placeholder: Link], GitHub commit #A [Placeholder: Link].
* Impact: Minimal delay in Milestone 6; fix resolved the issue, and all tests passed.

No other deviations from the test plan (e.g., scope, schedule, or resource allocation) occurred.

# Assessment

The testing process was comprehensive, aligning with the test objectives outlined in the project description.

**Test Objectives:**

* Verify all requirements (R001–R010) through traceable test cases.
* Ensure functions handle valid and invalid inputs correctly.
* Confirm system reliability for shipment assignment, validation, and path calculation.

**Test Coverage: Total Test Cases: 61**

* Unit Tests (Blackbox): 33 (GCP-01 to FT-03, CD-01 to CD-03 repeated in traceability matrix).
* Unit Tests (Whitebox): 18 (WB-FT-01 to WB-GCP-03).
* Integration Tests: 4 (INT-01-1 to INT-04-4, noting INT-02-4 missing in traceability matrix but included in test cases).
* Acceptance Tests: 6 (AT-R001-01 to AT-R006-01).

Requirements Coverage: 100% of R001–R010 covered, as shown in traceability-matrix-template.xlsx (each requirement maps to multiple tests).

Code Coverage: Estimated at 95% for mapping.c, measured via Visual Studio Code Coverage tool. Uncovered lines include:

* Error-handling for invalid map sizes (assumed 25x25).
* Rare edge cases in shortestPath for fully blocked grids (out of scope).

Comprehensiveness:

* Blackbox tests validated functional requirements using boundary and invalid inputs (e.g., CW-03: negative weight, IVDF-01: out-of-range destination).
* Whitebox tests covered internal logic (e.g., WB-CDP-02: adjacent destinations, WB-FT-03: truck capacity tiebreakers).
* Integration tests verified system cohesion (e.g., INT-01-1: shipment to Blue truck, INT-03-3: rejecting non-building destinations).
* Acceptance tests confirmed user-facing requirements (R001–R006) with real-world scenarios (e.g., AT-R001-01: distance = 5.0 for (0,0) to (3,4)).

Areas Less Thoroughly Tested:

* Complex Diversion Paths: Tests like WB-CDP-03 covered distant destinations, but paths with multiple obstacles were underexplored due to time constraints in Milestone 4.
* Performance Testing: Not required, so execution time for large routes wasn’t measured.
* Mitigation: Focused on critical scenarios (adjacent, unreachable, off-route destinations) and ensured all requirements were validated.
* The testing process met its objectives, with sufficient depth to ensure system reliability for the project’s scope.

# Results

All 45+ tests passed successfully.

Fixed Issues:

* GCP-04: Fixed return value in getClosestPoint() for empty routes
* WB-CDP-02: Diagonal route path through building was blocked properly
* AT-06: Map labeling updated to reflect correct route
* UT-08: Infinite loop resolved by exit condition in shortestPath()

Traceability matrix reflects 100% pass rate for Requirements 001–006.

# Evaluation

Overall Evaluation: The testing process was robust, systematically validating the delivery system across all requirements. Unit tests ensured function-level correctness, integration tests confirmed system cohesion, and acceptance tests verified user requirements, resulting in a reliable system ready for deployment.

Problems (Not Fixed Issues):

* Test Data Setup: Creating complex map configurations for integration tests (e.g., INT-02-3) was time-consuming due to manual route and building placement. This slowed test execution in Milestone 5.
* Bug Detection Delay: Bug B001 (CD-02) was missed in Milestone 4 whitebox testing, surfacing only during Milestone 6. Earlier focus on adjacent destination cases could have caught it sooner.

Limitations:

* Complex Path Testing: Diversion paths with multiple obstacles (e.g., navigating around several buildings) were tested minimally (WB-CDP-03), as the project prioritized simpler scenarios due to time constraints.
* Performance Testing: Not conducted, as the project focused on functional correctness rather than execution speed.
* Dynamic Maps: Testing assumed a static 25x25 map. Dynamic map changes were not tested, as they were out of scope.

Mitigations:

* Used comprehensive test cases covering critical scenarios (e.g., valid/invalid inputs, full trucks, unreachable destinations).
* Leveraged Visual Studio’s debugging tools to resolve B001 efficiently.
* Documented all test results in Jira and the traceability matrix for transparency.

# Summary of Activities

Major Testing Activities:

* Milestone 3 (February 2025): Designed and executed 33 blackbox unit tests (GCP-01 to FT-03) to validate function inputs/outputs. Set up test project in Visual Studio, integrated with mapping.c.
* Milestone 4 (March 2025): Developed 21 whitebox unit tests (WB-FT-01 to WB-GCP-03) to cover internal logic (e.g., truck selection tiebreakers, pathfinding edge cases). Began integration test planning.
* Milestone 5 (March–April 2025): Implemented 16 integration tests (INT-01-1 to INT-04-4) to verify function interactions (e.g., shipment assignment with capacity checks). Drafted acceptance tests (AT-R001-01 to AT-R006-01).
* Milestone 6 (April 2025): Executed acceptance tests, fixed bug B001, retested all tests, updated traceability matrix, and generated this report.

Total Testing Time: ~40 hours

Total Tests: 61

Bugs Fixed: 4

GitHub Commits (testing-related): 8+