DIGT2107: Practice of Software Development

Project Iteration 3: Testing and Initial Development

Course: DIGT2107 – Fall Term 2025
Instructor: Dr. May Haidar
Due Date: November 2, 2025

1. Introduction

Project Name: Your Project Name
Team Number: Your Team Number
Team Members: List of Team Members

Document Overview: This document outlines the deliverables for Iteration 3, focusing on the initial development phase. It highlights how the test cases align with the functional and non-functional requirements specified in previous iterations. The primary goals include implementing core functionality, developing comprehensive unit tests, and delivering a working prototype.

2. Iteration Goals and Objectives

For this iteration, the team should achieve the following:

- Implement core functionality linked directly to high-priority user stories.
- Develop and run unit tests that are directly mapped to specific requirements.
- Deliver an initial prototype that demonstrates the relationship between test cases and the project's requirements.

3. Requirement to Test Case Traceability

You are required to create a **Requirement-to-Test Case Traceability Matrix** that explicitly links each functional and non-functional requirement to the relevant test cases. This ensures that all requirements are validated through appropriate testing.

Sample Traceability Matrix:

Requirement ID	Requirement Description	Test Case ID	Test Case Description	Coverage
FR-001	The system shall allow users to add inventory items	TC-001	Test adding a new item with valid inputs	Fully Covered
FR-002	The system shall enable user authentication	TC-002	Test login functionality for admin and regular users	Partially Covered

NFR-001	The system shall handle 100 transactions per minute	TC-003	Performance test simulating 100 transactions per minute	Fully Covered

Students should explain the traceability matrix by discussing:

- How each requirement is validated by one or more test cases.
- Any gaps where additional test cases may be required.
- Coverage status for each requirement (e.g., fully covered, partially covered, or uncovered). **This result may be kept empty until you perform the test coverage in future iterations.**

Note: coverage status for non-functional requirements is not required in this project.

4. Detailed Test Case Descriptions

For each test case, students must include the following details:

1. **Test Case ID**: A unique identifier for the test case (e.g., TC-001).
2. **Related Requirement ID(s)**: Link to the corresponding requirement(s) (e.g., FR-001).
3. **Test Case Title**: A short descriptive title (e.g., Add Inventory Item - Valid Inputs).
4. **Preconditions**: Conditions that must be met before executing the test (e.g., User is logged in).
5. **Test Steps**: A step-by-step procedure for executing the test.
6. **Expected Results**: The expected outcome if the system behaves correctly.
7. **Actual Results**: The actual outcome when the test is run.
8. **Status**: Pass/Fail/Blocked.
9. **Priority**: Indicate the criticality (e.g., High, Medium, Low).

Results of 7 and 8 may be reported in later iterations.

Sample Test Case:

Test Case ID	TC-001
Title	Add Inventory Item - Valid Inputs
Requirement	FR-001
Preconditions	User is logged in as an Admin
Steps	1. Navigate to the Inventory page 2. Click "Add Item" 3. Enter valid item details (name, quantity, price) 4. Click "Save"
Expected Result	The new item is added, and the inventory list updates.
Actual Result	(To be filled after test execution)
Status	(Pass/Fail)
Priority	High

5. Code Repository and Branching Strategy

- **Repository Link**: [Insert GitHub Repository URL]
- **Branching Strategy**:
  - **Main Branch**: Contains stable code.
  - **Development Branch**: Ongoing development.
  - **Feature Branches**: Separate branches for key features (e.g., feature/inventory, feature/auth).

6. Task Allocation and Timeline

- **Task Breakdown**: Define tasks for each team member, including coding, testing, and documentation.
- **Sprint Plan**:
  - Week 1: Implement core features and set up unit tests.
  - Week 2: Refine features and run tests.
  - Week 3: Finalize the prototype and gather feedback.

7. Submission Guidelines

- **GitHub Repository**: Push all code and documentation to the repository. Tag this version as “ITR2.1”.
- **eClass Submission**: Submit a PDF containing:
  - Prototype Overview
  - Updated Requirements and Use Cases
  - Test Plan and Results
  - Updated Iteration Plan