Test Plan Document

Student Management System

Document Version: 1.0

Date: December 2025

---

1. Introduction

1.1 Purpose

This document defines the testing strategy, test cases, and procedures for validating the Student Management System.

1.2 Scope

The test plan covers functional testing, integration testing, and system testing for all components of the Student Management System.

---

2. Test Strategy

2.1 Testing Levels

Unit Testing

- Test individual methods and classes
- Validate encapsulation and data integrity
- Test edge cases and boundary conditions

Integration Testing

- Test interactions between classes
- Validate data flow between layers
- Test CRUD operation sequences

System Testing

- End-to-end user scenarios
- Menu navigation and workflow testing
- Error handling and recovery testing

User Acceptance Testing

- Validate against requirements
- Verify usability and user experience
- Confirm business rules implementation

2.2 Testing Types

- **Functional Testing**: Verify features work as specified
- **Negative Testing**: Test error handling and validation
- **Boundary Testing**: Test limits and edge cases
- **Regression Testing**: Ensure changes don't break existing functionality

---

3. Test Environment

3.1 Hardware Requirements

- Standard PC/Laptop
- Minimum 2GB RAM
- Console/Terminal support

3.2 Software Requirements

- Java Development Kit (JDK) 8 or higher
- Operating System: Windows/Linux/macOS
- Terminal/Command Prompt

3.3 Test Data

- Sample student records
- Valid and invalid input data
- Edge case values

---

4. Test Cases

4.1 Authentication Test Cases

TC-001: Valid Login

Objective: Verify successful login with correct credentials **Preconditions**: System is started **Test Steps**:

1. Enter username: "admin"
2. Enter password: "admin123" **Expected Result**: Login successful, main menu displayed **Priority**: High

TC-002: Invalid Login - Wrong Password

Objective: Verify login fails with incorrect password **Preconditions**: System is started **Test Steps**:

1. Enter username: "admin"
2. Enter password: "wrongpass" **Expected Result**: Error message, prompt to retry **Priority**: High

TC-003: Invalid Login - Maximum Attempts

Objective: Verify system locks after 3 failed attempts **Preconditions**: System is started **Test Steps**:

1. Enter wrong credentials 3 times **Expected Result**: System exits after 3 attempts **Priority**: High

TC-004: Invalid Login - Empty Credentials

Objective: Verify login handles empty inputs **Preconditions**: System is started **Test Steps**:

1. Press Enter without entering username
2. Press Enter without entering password **Expected Result**: Login fails, counted as attempt **Priority**: Medium

---

4.2 Student Creation Test Cases

TC-005: Create Valid Student

Objective: Verify student creation with valid data **Preconditions**: Logged in to system **Test Steps**:

1. Select option 1 (Create Student)
2. Enter first name: "John"
3. Enter last name: "Doe"
4. Enter email: "john.doe@example.com"
5. Enter age: 20 **Expected Result**:

- Student created successfully
- Unique ID generated (e.g., STU1001)
- Success message displayed **Priority**: High

TC-006: Create Student - Invalid Name

Objective: Verify name validation **Preconditions**: Logged in, creating student **Test Steps**:

1. Enter first name: "John123" (contains numbers) **Expected Result**: Error message, prompt to re-enter **Priority**: High

TC-007: Create Student - Invalid Email

Objective: Verify email validation **Preconditions**: Logged in, creating student **Test Steps**:

1. Enter valid names
2. Enter email: "invalidemail" (no @ or domain) **Expected Result**: Error message, prompt to re-enter **Priority**: High

TC-008: Create Student - Invalid Age

Objective: Verify age validation **Preconditions**: Logged in, creating student **Test Steps**:

1. Enter valid names and email
2. Enter age: -5 (negative) **Expected Result**: Error message, prompt to re-enter **Priority**: High

TC-009: Create Student - Age Boundary

Objective: Test age boundaries (0, 1, 149, 150) **Preconditions**: Logged in, creating student **Test Steps**:

1. Try age: 0 - Should fail
2. Try age: 1 - Should succeed
3. Try age: 149 - Should succeed
4. Try age: 150 - Should fail **Expected Result**: Validation as specified **Priority**: Medium

TC-010: Unique ID Generation

Objective: Verify each student gets unique ID **Preconditions**: Logged in **Test Steps**:

1. Create student 1
2. Note ID (e.g., STU1001)
3. Create student 2
4. Note ID (e.g., STU1002) **Expected Result**: IDs are sequential and unique **Priority**: High

---

4.3 Student Retrieval Test Cases

TC-011: View All Students - Empty

Objective: Verify display when no students exist **Preconditions**: Logged in, no students in system **Test Steps**:

1. Select option 2 (View All Students) **Expected Result**: Message "No students found in the system" **Priority**: Medium

TC-012: View All Students - Multiple Records

Objective: Verify list display with multiple students **Preconditions**: Logged in, 3+ students exist **Test Steps**:

1. Select option 2 (View All Students) **Expected Result**:

- Tabular format displayed
- All students listed with ID, name, email, age, GPA
- Proper formatting and alignment **Priority**: High

TC-013: View Student Details - Valid ID

Objective: Verify individual student detail view **Preconditions**: Logged in, student STU1001 exists **Test Steps**:

1. Select option 3 (View Student Details)
2. Enter ID: "STU1001" **Expected Result**:

- Full student information displayed
- Includes all fields and enrolled courses
- Proper formatting **Priority**: High

TC-014: View Student Details - Invalid ID

Objective: Verify error handling for non-existent student **Preconditions**: Logged in **Test Steps**:

1. Select option 3
2. Enter ID: "STU9999" (doesn't exist) **Expected Result**: Error message "Student not found with ID: STU9999" **Priority**: High

TC-015: View Student Details - Case Insensitive

Objective: Verify ID lookup is case-insensitive **Preconditions**: Logged in, student STU1001 exists **Test Steps**:

1. Select option 3
2. Enter ID: "stu1001" (lowercase) **Expected Result**: Student found and displayed **Priority**: Medium

---

4.4 Student Update Test Cases

TC-016: Update First Name

Objective: Verify selective update of first name **Preconditions**: Logged in, student STU1001 exists **Test Steps**:

1. Select option 4 (Update Student)
2. Enter ID: "STU1001"
3. Select field: 1 (First Name)
4. Enter new name: "Jane" **Expected Result**: First name updated, other fields unchanged **Priority**: High

TC-017: Update Last Name

Objective: Verify selective update of last name **Preconditions**: Logged in, student exists **Test Steps**:

1. Select option 4
2. Select field: 2 (Last Name)
3. Enter new last name **Expected Result**: Last name updated, other fields unchanged **Priority**: High

TC-018: Update Email

Objective: Verify email update with validation **Preconditions**: Logged in, student exists **Test Steps**:

1. Select option 4
2. Select field: 3 (Email)
3. Enter new email: "newemail@example.com" **Expected Result**: Email updated successfully **Priority**: High

TC-019: Update Email - Invalid Format

Objective: Verify email validation during update **Preconditions**: Logged in, student exists **Test Steps**:

1. Select option 4
2. Select field: 3 (Email)

3. Enter invalid email: "notanemail" **Expected Result**: Validation error, prompt to re-enter **Priority**: High

TC-020: Update Age

Objective: Verify age update **Preconditions**: Logged in, student exists **Test Steps**:

1. Select option 4
2. Select field: 4 (Age)
3. Enter new age: 25 **Expected Result**: Age updated successfully **Priority**: High

TC-021: Update Non-Existent Student

Objective: Verify error handling for invalid ID during update **Preconditions**: Logged in **Test Steps**:

1. Select option 4
2. Enter ID: "STU9999" **Expected Result**: Error "Student not found" **Priority**: High

---

4.5 Student Deletion Test Cases

TC-022: Delete Student - With Confirmation

Objective: Verify student deletion with confirmation **Preconditions**: Logged in, student STU1001 exists **Test Steps**:

1. Select option 5 (Delete Student)
2. Enter ID: "STU1001"
3. Confirm: "yes" **Expected Result**:

- Student deleted successfully
- Confirmation message displayed
- Student no longer in system **Priority**: High

TC-023: Delete Student - Cancel Operation

Objective: Verify deletion can be cancelled **Preconditions**: Logged in, student exists **Test Steps**:

1. Select option 5
2. Enter student ID
3. Confirm: "no" **Expected Result**:

- Deletion cancelled
- Student still in system **Priority**: High

TC-024: Delete Non-Existent Student

Objective: Verify error handling for invalid ID **Preconditions**: Logged in **Test Steps**:

1. Select option 5
2. Enter ID: "STU9999" **Expected Result**: Error "Student not found" **Priority**: High

---

4.6 Search Test Cases

TC-025: Search by Student ID

Objective: Verify search by exact ID **Preconditions**: Logged in, student STU1001 exists **Test Steps**:

1. Select option 6 (Search Students)
2. Enter search term: "STU1001" **Expected Result**: Student STU1001 found and displayed **Priority**: High

TC-026: Search by First Name

Objective: Verify search by partial name **Preconditions**: Logged in, student "John Doe" exists **Test Steps**:

1. Select option 6
2. Enter search term: "John" **Expected Result**: All students with "John" in first name displayed **Priority**: High

TC-027: Search by Email

Objective: Verify search by email **Preconditions**: Logged in, students exist **Test Steps**:

1. Select option 6
2. Enter search term: "example.com" **Expected Result**: All students with emails containing "example.com" **Priority**: High

TC-028: Search - Case Insensitive

Objective: Verify search is case-insensitive **Preconditions**: Logged in, student "John" exists **Test Steps**:

1. Select option 6
2. Enter search term: "john" (lowercase) **Expected Result**: Student found regardless of case **Priority**: Medium

TC-029: Search - No Results

Objective: Verify handling when no matches found **Preconditions**: Logged in **Test Steps**:

1. Select option 6
2. Enter search term: "NonExistent" **Expected Result**: Message "No students found matching: NonExistent" **Priority**: Medium

TC-030: Search - Empty Term

Objective: Verify validation of empty search term **Preconditions**: Logged in **Test Steps**:

1. Select option 6
2. Press Enter without typing **Expected Result**: Error "Search term cannot be empty" **Priority**: Medium

---

4.7 Course Assignment Test Cases

TC-031: Assign Valid Course

Objective: Verify course assignment with valid data **Preconditions**: Logged in, student STU1001 exists **Test Steps**:

1. Select option 7 (Assign Course)
2. Enter student ID: "STU1001"
3. Enter course code: "CS101"
4. Enter course name: "Introduction to Programming"
5. Enter credits: 3
6. Enter grade: 85 **Expected Result**:

- Course assigned successfully
- GPA calculated automatically
- Updated GPA displayed **Priority**: High

TC-032: Assign Course - Invalid Code Format

Objective: Verify course code validation **Preconditions**: Logged in, student exists **Test Steps**:

1. Select option 7
2. Enter student ID
3. Enter course code: "123" (invalid format) **Expected Result**: Validation error, prompt to re-enter **Priority**: High

TC-033: Assign Course - Invalid Credits

Objective: Verify credits validation **Preconditions**: Logged in, student exists **Test Steps**:

1. Select option 7
2. Enter valid student and course info
3. Enter credits: 0 (invalid) **Expected Result**: Validation error, must be 1-10 **Priority**: High

TC-034: Assign Course - Invalid Grade

Objective: Verify grade validation **Preconditions**: Logged in, student exists **Test Steps**:

1. Select option 7
2. Enter valid info
3. Enter grade: 105 (out of range) **Expected Result**: Validation error, must be 0-100 **Priority**: High

TC-035: Assign Multiple Courses

Objective: Verify multiple course assignments and GPA calculation **Preconditions**: Logged in, student exists **Test Steps**:

1. Assign course 1: CS101, 3 credits, 90% grade
2. Assign course 2: MATH201, 4 credits, 85% grade
3. Check student details **Expected Result**:

- Both courses listed
- GPA calculated correctly: (4.0×3 + 3.0×4) / 7 = 3.43 **Priority**: High

---

4.8 Course Removal Test Cases

TC-036: Remove Existing Course

Objective: Verify course removal **Preconditions**: Logged in, student has CS101 course **Test Steps**:

1. Select option 8 (Remove Course)
2. Enter student ID
3. Enter course code: "CS101" **Expected Result**:

- Course removed successfully
- GPA recalculated
- Updated GPA displayed **Priority**: High

TC-037: Remove Course - Student Has No Courses

Objective: Verify handling when student has no courses **Preconditions**: Logged in, student exists with no courses **Test Steps**:

1. Select option 8
2. Enter student ID **Expected Result**: Message "No courses enrolled" **Priority**: Medium

TC-038: Remove Non-Existent Course

Objective: Verify handling of invalid course code **Preconditions**: Logged in, student exists **Test Steps**:

1. Select option 8
2. Enter student ID
3. Enter course code: "INVALID999" **Expected Result**: Course not found, no error (silent handling) **Priority**: Low

---

4.9 GPA Calculation Test Cases

TC-039: GPA Calculation - Single Course

Objective: Verify GPA with one course **Test Steps**:

1. Create student
2. Assign course: 3 credits, 92% grade (A = 4.0) **Expected Result**: GPA = 4.0 **Priority**: High

TC-040: GPA Calculation - Multiple Courses

Objective: Verify weighted GPA calculation **Test Steps**:

1. Create student
2. Assign CS101: 3 credits, 90% (4.0)

3. Assign MATH201: 4 credits, 75% (2.0) **Expected Result**: GPA = (4.0×3 + 2.0×4) / 7 = 2.86 **Priority**: High

TC-041: GPA Calculation - Grade Boundaries

Objective: Verify grade point conversion **Test Cases**:

- 90-100% = 4.0 (A)
- 80-89% = 3.0 (B)
- 70-79% = 2.0 (C)
- 60-69% = 1.0 (D)
- 0-59% = 0.0 (F) **Expected Result**: Correct grade points for each range **Priority**: High

TC-042: GPA Calculation - No Courses

Objective: Verify GPA when no courses enrolled **Test Steps**:

1. Create student
2. Check GPA (no courses assigned) **Expected Result**: GPA = 0.0 **Priority**: Medium

TC-043: GPA Recalculation - After Course Removal

Objective: Verify GPA updates when course removed **Test Steps**:

1. Student has 2 courses, GPA = 3.5
2. Remove one course **Expected Result**: GPA recalculated based on remaining course **Priority**: High

---

4.10 Statistics Test Cases

TC-044: View Statistics - With Data

Objective: Verify statistics display **Preconditions**: Logged in, multiple students with courses **Test Steps**:

1. Select option 9 (View Statistics) **Expected Result**:

- Total students count displayed
- Students with courses count
- Average GPA displayed
- Top students (GPA >= 3.0) listed **Priority**: Medium

TC-045: View Statistics - No Students

Objective: Verify statistics when system is empty **Preconditions**: Logged in, no students **Test Steps**:

1. Select option 9 **Expected Result**: Message "No students in the system" **Priority**: Low

TC-046: View Statistics - Top Students Filter

Objective: Verify top students filtering **Preconditions**: Students with various GPAs exist **Test Steps**:

1. Select option 9
2. Check top students section **Expected Result**: Only students with GPA >= 3.0 listed **Priority**: Medium

---

4.11 Input Validation Test Cases

TC-047: Non-Numeric Input for Menu

Objective: Verify menu handles non-numeric input **Preconditions**: Logged in, at main menu **Test Steps**:

1. Enter "abc" for menu choice **Expected Result**: Error message, prompt to re-enter **Priority**: High

TC-048: Out of Range Menu Selection

Objective: Verify menu validates choice range **Preconditions**: At main menu **Test Steps**:

1. Enter "99" (out of valid range 0-9) **Expected Result**: Error "Invalid choice. Please try again." **Priority**: Medium

TC-049: Empty Input Handling

Objective: Verify system handles empty inputs **Test Steps**:

1. Press Enter at various prompts without typing **Expected Result**: Appropriate validation messages **Priority**: Medium

---

4.12 Exit and Navigation Test Cases

TC-050: Exit Application

Objective: Verify clean exit from application **Preconditions**: Logged in, at main menu **Test Steps**:

1. Select option 0 (Exit) **Expected Result**:

- Goodbye message displayed
- Application terminates cleanly **Priority**: High

TC-051: Menu Navigation Flow

Objective: Verify return to main menu after operations **Preconditions**: Logged in **Test Steps**:

1. Perform any operation (e.g., create student)
2. Press Enter at "Press Enter to continue" **Expected Result**: Main menu displayed again **Priority**: Medium

---

5. Test Execution

5.1 Test Schedule

- Unit Testing: Days 1-2

- Integration Testing: Days 3-4
- System Testing: Days 5-6
- UAT: Days 7-8

5.2 Test Execution Process

1. Set up test environment
2. Execute test cases in order
3. Record results (Pass/Fail)
4. Document defects found
5. Retest after fixes
6. Generate test report

5.3 Test Result Recording

For each test case, record:

- Test Case ID
- Test Date
- Tester Name
- Status (Pass/Fail/Blocked)
- Comments
- Defect ID (if failed)

---

6. Defect Management

6.1 Defect Severity Levels

- **Critical**: System crash, data loss
- **High**: Major functionality broken
- **Medium**: Feature works but has issues
- **Low**: Minor UI issues, typos

6.2 Defect Lifecycle

1. New - Defect reported
2. Assigned - Assigned to developer
3. In Progress - Being fixed
4. Fixed - Fix completed
5. Retest - Awaiting verification
6. Closed - Verified and closed

---

7. Exit Criteria

Testing is complete when:

- All critical and high priority test cases pass
- All critical and high severity defects are fixed

- 95% of medium priority test cases pass
- Test coverage >= 90%
- Documentation is complete

---

8. Test Deliverables

- Test Plan Document (this document)
- Test Cases Spreadsheet
- Test Execution Report
- Defect Report
- Test Summary Report

---

9. Sample Test Execution Report Template

```
Test Execution Report
Date: __________
Tester: __________

Total Test Cases: 51
Executed: ____
Passed: ____
Failed: ____
Blocked: ____
Pass Rate: ____%

Critical Issues: ____
High Issues: ____
Medium Issues: ____
Low Issues: ____

Comments:
_______________________
```

Document Prepared By: QA Team
Approved By: Project Manager
Review Date: December 2025