Software Requirements Specification (SRS)

SPLWD: Student Profiling for Learners with Disabilities System Upgrade

Version: 2.0

Date: 2025

Project Lead and Document Owner: Mhar Andrei C. Macapallag

---

1. Introduction and Purpose

1.1 Purpose

This Software Requirements Specification (SRS) document describes the requirements for upgrading the existing SPLWD (Student Profiling for Learners with Disabilities) system. The upgrade focuses on modernizing the legacy codebase, improving system performance, enhancing code readability, and implementing best practices for maintainability.

1.2 Scope

The upgraded system will continue to serve the Sta. Cruz District of Laguna, specifically:

- Sta. Cruz Central Elementary School (111 LWD students)
- Bagumbayan Elementary School (32 LWD students)
- Gatid Elementary School (20 LWD students)

1.3 Definitions and Acronyms

- **LWD**: Learners with Disabilities
- **IEP**: Individual Educational Plan
- **ILP**: Individual Learner's Profile
- **ILMP**: Individual Learning Monitoring Plan
- **BIR**: Behavior Intervention Report
- **LRN**: Learner Registration Number
- **SPED**: Special Education

---

2. Overall Description

2.1 Product Perspective

The SPLWD system upgrade builds upon the existing web-based student profiling system. The original system was developed to digitize and centralize the management of LWD student records, replacing traditional paper-based filing systems vulnerable to natural disasters.

2.2 Product Functions

The upgraded system maintains all original functionality while improving:

- **Student Profile Management**: Digital storage and retrieval of LWD student information
- **Document Management**: Upload, storage, and organization of IEP, ILP, ILMP, and BIR documents
- **Progress Tracking**: Quarterly assessment and monitoring capabilities
- **Parent Engagement**: Limited access for parents to view progress and add observations
- **Multi-user Support**: Role-based access for Administrators, Principals, Secretaries, Teachers, and Parents

2.3 User Classes

1. **System Administrator**: Full system access and user management
2. **School Principals**: School-level oversight and reporting
3. **Principal's Secretary**: User account management and administrative support
4. **Teachers**: Student data entry, document upload, and progress tracking
5. **Parents/Guardians**: Limited access to view child's progress and add observations

---

3. Functional and Non-functional Requirements

3.1 Functional Requirements

3.1.1 User Authentication and Authorization

- **FR-001**: System shall provide secure login functionality for all user types
- **FR-002**: System shall implement role-based access control
- **FR-003**: System shall maintain user session management
- **FR-004**: System shall provide password reset functionality

3.1.2 Student Management

- **FR-005**: System shall allow creation of new student profiles
- **FR-006**: System shall support student information updates
- **FR-007**: System shall manage student enrollment status
- **FR-008**: System shall support student transfers between schools

3.1.3 Document Management

- **FR-009**: System shall support upload and storage of IEP documents
- **FR-010**: System shall support upload and storage of ILP documents
- **FR-011**: System shall support upload and storage of ILMP documents
- **FR-012**: System shall support upload and storage of BIR documents
- **FR-013**: System shall provide document versioning and history

3.1.4 Progress Tracking

- **FR-014**: System shall support quarterly progress report generation
- **FR-015**: System shall provide progress visualization through charts
- **FR-016**: System shall allow teacher remarks and observations

- **FR-017**: System shall support parent observation input

3.1.5 Reporting and Analytics

- **FR-018**: System shall generate enrollment reports
- **FR-019**: System shall provide student progress summaries
- **FR-020**: System shall maintain audit logs of all system activities

3.2 Non-functional Requirements

3.2.1 Performance Requirements

- **NFR-001**: System response time shall not exceed 3 seconds for standard operations
- **NFR-002**: System shall support concurrent access by up to 100 users
- **NFR-003**: File upload operations shall complete within 30 seconds for files up to 10MB

3.2.2 Security Requirements

- **NFR-004**: System shall implement data encryption for sensitive information
- **NFR-005**: System shall comply with Data Privacy Act requirements
- **NFR-006**: System shall implement secure file storage mechanisms
- **NFR-007**: System shall provide audit trails for all data modifications

3.2.3 Usability Requirements

- **NFR-008**: System interface shall be intuitive and require minimal training
- **NFR-009**: System shall be responsive and mobile-friendly
- **NFR-010**: System shall provide clear error messages and user feedback

3.2.4 Reliability Requirements

- **NFR-011**: System shall maintain 99.5% uptime during school hours
- **NFR-012**: System shall implement automated backup mechanisms
- **NFR-013**: System shall provide data recovery capabilities

---

4. System Features and Interfaces

4.1 User Interface Requirements

- Web-based responsive design compatible with modern browsers
- Bootstrap framework for consistent styling
- Intuitive navigation with role-based menu systems
- Print-friendly document formats

4.2 Hardware Interfaces

- Standard web server hardware requirements
- Database server with adequate storage capacity

- Network infrastructure supporting concurrent users

4.3 Software Interfaces

- **Database**: MySQL for data persistence
- **Web Server**: Apache/Nginx compatible
- **PHP Framework**: Modern PHP 8.x compatibility
- **Frontend**: HTML5, CSS3, JavaScript, Bootstrap

4.4 Communication Interfaces

- HTTPS protocol for secure data transmission
- RESTful API design for potential future integrations
- Email notifications for system alerts

---

5. Assumptions and Constraints

5.1 Assumptions

- Users have basic computer literacy and internet access
- Schools maintain reliable internet connectivity
- Administrative support is available for user training
- Data migration from legacy system is feasible

5.2 Constraints

- **Budget**: Limited budget for infrastructure upgrades
- **Timeline**: Upgrade must be completed within academic year
- **Compatibility**: Must maintain compatibility with existing data formats
- **Compliance**: Must adhere to DepEd regulations and Data Privacy Act

---

6. Use Case Diagrams or Descriptions

6.1 Primary Use Cases

Use Case 1: Teacher Manages Student Profile

- **Actor**: Teacher
- **Description**: Teacher creates, updates, and maintains student profiles
- **Preconditions**: Teacher is authenticated and has appropriate permissions
- **Main Flow**:
  1. Teacher accesses student management interface
  2. Teacher selects student or creates new profile
  3. Teacher enters/updates student information
  4. System validates and saves data
  5. System logs the activity

Use Case 2: Parent Views Student Progress

- **Actor**: Parent/Guardian
- **Description**: Parent accesses child's progress information
- **Preconditions**: Parent has valid LRN and system access
- **Main Flow**:
  1. Parent logs in using child's LRN
  2. System displays student progress dashboard
  3. Parent views grades, reports, and teacher comments
  4. Parent optionally adds observations
  5. System saves parent input

Use Case 3: Administrator Manages System Users

- **Actor**: System Administrator
- **Description**: Administrator manages user accounts and permissions
- **Preconditions**: Administrator is authenticated
- **Main Flow**:
  1. Administrator accesses user management interface
  2. Administrator creates/modifies user accounts
  3. Administrator assigns roles and permissions
  4. System validates and applies changes
  5. System notifies affected users

---

7. Testing Tool Documentation

7.1 Testing Framework: PHPUnit

Tool Selection Rationale: PHPUnit was selected as the primary testing framework for the SPLWD system upgrade based on the following criteria:

1. **Industry Standard**: PHPUnit is the de facto standard for PHP testing, ensuring long-term support and community resources
2. **Comprehensive Testing**: Supports unit testing, integration testing, and functional testing capabilities
3. **CI/CD Integration**: Seamlessly integrates with continuous integration pipelines
4. **Mature Ecosystem**: Extensive documentation, plugins, and third-party tools available
5. **Code Coverage**: Built-in code coverage analysis for quality assurance

7.2 Testing Implementation

Test Suite Configuration:

- **Total Tests**: 220 comprehensive test cases
- **Total Assertions**: 962 individual assertions
- **Success Rate**: 100% (all tests passing)
- **Coverage**: Comprehensive coverage of core system functionality
- **Deprecation Warnings**: 1 PHPUnit deprecation (addressed in upgrade)
- **Risky Tests**: 1 risky test (documented and monitored)

Test Categories:

1. **Unit Tests**: Individual component and function testing
2. **Integration Tests**: Database and API integration verification
3. **Functional Tests**: End-to-end user workflow testing
4. **Security Tests**: Authentication and authorization validation

7.3 Additional Testing Tools

Supporting Tools:

- **Composer**: Dependency management and autoloading

Quality Assurance Metrics:

- Code coverage: >98% for critical components
- Cyclomatic complexity: Maintained below threshold levels
- Technical debt: Significantly reduced through refactoring
- Performance benchmarks: All response times within acceptable limits

---