Software Requirements Specification (SRS)

Project Title:

School Management Information System of Lumban National High School

Capstone Group:

(Insert group names here)

Institution:

Laguna State Polytechnic University – Santa Cruz, Laguna

---

1. INTRODUCTION AND PURPOSE

1.1 Background

The **School Management Information System (SMIS)** is a web-based platform designed to streamline academic and administrative operations at Lumban National High School. Originally focused on registration, schedules, and grade encoding, the system has been upgraded to include **quizzes**, **assignments**, and **machine learning-based performance prediction**, enhancing both student evaluation and teacher insight.

1.2 Scope

This system provides functionalities for students, teachers, and administrators. It includes modules for student and teacher management, class and subject scheduling, quiz and assignment administration, and ML-based academic performance forecasting. The platform is designed for desktop browsers and is hosted using a PHP-MySQL environment.

1.3 Objectives

- Develop a comprehensive information system for managing school operations
- Implement quiz and assignment modules for digital learning

- Integrate ML features to predict student performance
- Provide dashboards for all user types (admin, teacher, student)
- Evaluate usability and effectiveness using the Technology Acceptance Model (TAM)

---

2. OVERALL DESCRIPTION

2.1 User Classes and Characteristics

- **Students**: Can take quizzes/assignments, view scores and predictions
- **Teachers**: Manage class content, upload assessments, monitor progress
- **Administrators**: Manage system settings, users, and reports

2.2 Operating Environment

- **Platform**: Web (PHP, HTML, CSS, JavaScript)
- **Database**: MySQL
- **ML Tools**: TensorFlow.js or Python integration (for score prediction)
- **Hosting**: Apache server with PHP support

2.3 Design and Implementation Constraints

- Internet connection required for real-time operations
- Minimum 3 historical student scores required for ML predictions
- Responsive only for desktop browsers (future upgrade for mobile)

2.4 User Needs

- Simplified and role-specific dashboards
- Accurate student performance insights

- Centralized data and academic management
- Secure login and data access

---

3. FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

3.1 Functional Requirements

- **Registration/Login**: Role-based access for admin, teachers, students
- **Quiz and Assignment**: Teachers can create, students can answer
- **Grading System**: Manual and automatic scoring
- **ML Prediction**: Forecast student performance trends
- **Admin Control Panel**: Full data management and reporting
- **Visualization**: Score trends and performance analytics via charts

3.2 Non-Functional Requirements

- **Performance**: <2s response time for major actions
- **Security**: Role-based access, encrypted login
- **Usability**: User-friendly UI with clear navigation
- **Scalability**: Can be extended for multiple grade levels or schools
- **Compatibility**: Runs on all major desktop browsers

---

4. SYSTEM FEATURES AND INTERFACES

4.1 User Interface

- Role-based layout (Admin, Teacher, Student)

- Quiz/assignment submission panels
- Dashboard with real-time statistics
- Score feedback and performance graphs

4.2 System Interfaces

- **MySQL Database**: Data storage and retrieval
- **PHP Backend**: Logic and controller functions
- **ML Model (TensorFlow.js/Python)**: Score predictions
- **Chart Libraries (e.g., Chart.js)**: Graph generation

---

5. ASSUMPTIONS AND CONSTRAINTS

- Target users have access to internet-connected computers
- Teachers are trained to manage quizzes and monitor predictions
- ML module requires a sufficient dataset to generate forecasts
- All student assessments are submitted within the system

---

6. USE CASE DIAGRAMS OR DESCRIPTIONS

Use Case 1: Student Takes Quiz

Actor: Student
Flow:

- Log in
- Select subject > take quiz/assignment
- Submit answers

- View results and predicted score trend

Use Case 2: Teacher Creates Assignment

Actor: Teacher
Flow:

- Log in
- Choose class and subject
- Upload quiz or assignment
- Monitor student submissions and performance

Use Case 3: ML Performance Prediction

Actor: System
Flow:

- Collect past scores
- Train prediction model
- Display predicted scores and analytics to teacher/student

---

7. TESTING TOOL DOCUMENTATION

7.1 Unit Testing

Tool: Jest / PHPUnit (depending on frontend/backend logic)

7.2 Performance Benchmarking

Tool: Lighthouse (Google DevTools for measuring performance)

7.3 Compatibility Testing

Tool: BrowserStack (test across Chrome, Firefox, Edge)

7.4 Rationale for Tool Selection

Chosen tools are open-source, compatible with PHP-based systems, and provide reliable testing for web applications.