**Software Requirement Specification (Version 1.2)**

**Project Overview**

The goal of this project is to develop a scalable Learning Management System (LMS) that supports role-based dashboards, SCORM integration, progress tracking, gamification, reporting, and user-friendly features. The LMS serves multiple roles: SuperAdmin, Admin, Instructor, and Learner, and includes functionalities tailored to branches and groups.

**Functional Requirements**

**1. User Management**

**Roles:**

* **SuperAdmin:** Full system-wide access to all branches and settings.
* **Admin:** Branch-specific access for user and course management.
* **Instructor:** Manage assigned courses and learners.
* **Learner:** Access enrolled courses and track progress.

**Features:**

* User creation and management (name, email, role, timezone, language).
* Group and branch assignments.
* Password management and profile updates.
* Role-based access controls (RBAC).
* Messaging capabilities between roles (e.g., Admin-Instructor, Instructor-Learner).

**2. Course Management**

**Course Features:**

* Create, update, and archive courses.
* Add content: text, videos, SCORM packages, assessments (quizzes, assignments).
* Support for prerequisites and dynamic learning paths.
* Manage units and modules with sequencing.

**Instructor Features:**

* Upload and manage SCORM content.
* Add quizzes, assignments (with file upload support), and discussions.
* Monitor learner progress and engagement.
* Grade and provide feedback on assignments.
* Message learners through the system.

**Learner Features:**

* View and interact with course materials.
* Submit assignments with file uploads.
* Take quizzes and participate in discussions.
* Track progress and view grades/feedback.

**3. Progress Tracking**

* Track course/module completion for learners.
* Display progress and scores on dashboards.
* Provide real-time tracking for SCORM runtime data (e.g., scores, completion status).

**4. SCORM Support**

* Upload, render, and interact with SCORM-compliant content.
* Runtime tracking for learner progress, scores, and completions.

**5. Notifications**

* In-app and email notifications for:
  + Course updates.
  + Deadlines and assignments.
  + Feedback from instructors.

**6. Reporting**

**Admin Reports:**

* User activity: Logins, progress, completions.
* Course metrics: Completion rates, performance.
* Export data (Excel, CSV).

**Instructor Reports:**

* Learner engagement and progress.
* Assessment submissions and grading analytics.

**7. Gamification**

* Points, badges, and levels for engagement.
* Leaderboards at group and branch levels.
* Customizable rewards for milestones.

**8. Accessibility and Responsiveness**

* Ensure compatibility across devices (desktop, tablet, and mobile).
* Adhere to accessibility standards (e.g., WCAG compliance).

**Non-Functional Requirements**

**1. Security**

* Secure authentication using Django’s built-in mechanisms.
* Role-based access control for sensitive data.
* Data encryption for sensitive fields.

**2. Scalability**

* Support up to 300 users with the ability to scale for future growth.
* Optimize database queries and SCORM storage.

**3. Performance**

* Fast page load times and efficient data handling.
* Real-time updates for dashboards and SCORM runtime data.

**4. Maintainability**

* Modular architecture for easy updates.
* Comprehensive developer documentation.

**Development Plan**

**Phase 1: Core Features (Weeks 1-4)**

* User and role management.
* Course creation and unit management.
* Basic reporting features.

**Phase 2: Advanced Features (Weeks 5-8)**

* SCORM integration: Upload, render, and track SCORM content.
* Enhanced dashboards for Admins, Instructors, and Learners.
* Reporting and analytics.
* Gamification elements.

**Phase 3: Post Launch**

* Scalability improvements.
* Integration with external tools (e.g., Zoom, CRM systems).
* Automation for data sync (e.g., FTP, API integrations).

**Immediate Next Steps**

1. Finalize SCORM integration for Instructor and Learner dashboards.
2. Add content management tools for Instructors.
3. Begin implementing gamification widgets for Learners.
4. Expand Admin dashboard functionality for user and course management.

**Version History:**

* **Version 1.0**: Initial draft (19/12/2024).
* **Version 1.2**: Added functionalities for messaging, assignment submissions, discussions, and gamification (22/12/2024).