SRS - Software Requirements Specification (LMS)

Phiên bản: 1.0  
Ngày tạo: 2025-08-23  
Tác giả: Generated by assistant  
Nguồn: Nội dung gốc: SDD (SDD.docx)

# Giới thiệu

## Mục tiêu

Tài liệu SRS này mô tả các yêu cầu chức năng và phi chức năng cho hệ thống LMS (Learning Management System). Mục tiêu: cung cấp nền tảng rõ ràng cho việc phát triển, kiểm thử và xác nhận tính năng của hệ thống.

## Phạm vi

Hệ thống LMS hỗ trợ quản lý khóa học, module, bài học, bài kiểm tra (quizzes), ghi danh học viên, theo dõi tiến độ, thông báo và quản lý tài nguyên (file assets). API REST sẽ được cung cấp cho giao diện web/mobile. Authentication dùng JWT / ASP.NET Identity.

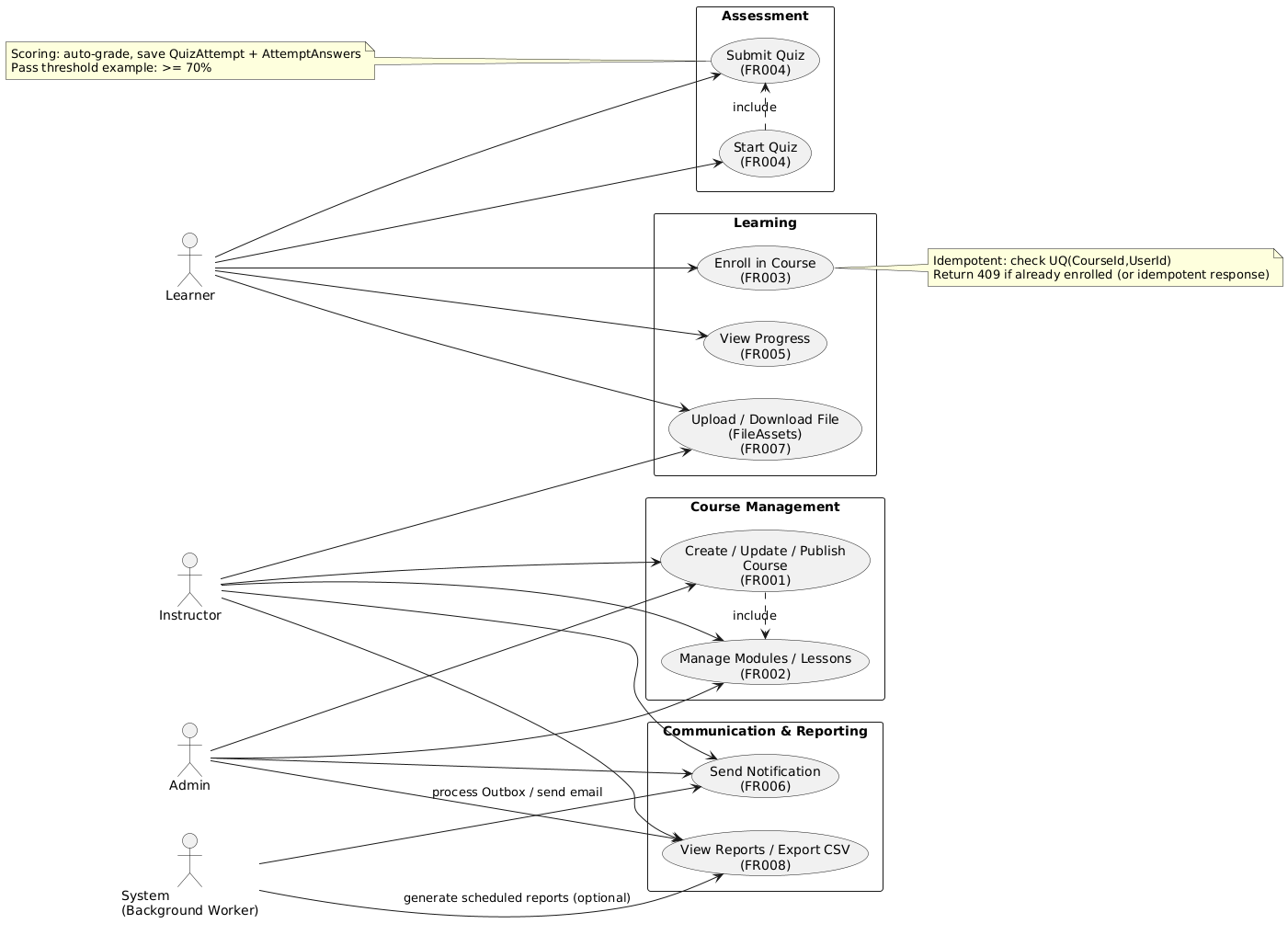
# Các actor (vai trò)

* Admin
* Instructor / Giảng viên
* Learner / Học viên
* System (background worker)

# Yêu cầu chức năng (Functional Requirements)

## Use Case Diagram

Use case diagram: Admin, Instructor, Learner và System (Background Worker) tương tác với các use case: Create/Manage Course, Enroll, Start/Submit Quiz, View Progress, FileAssets, Notify, Reports.



Hình 1. Use Case Diagram — LMS mini

## Functional Requirements

**FR001 - Quản lý khóa học**

Admin/Instructor có thể tạo, cập nhật, publish và xóa (soft-delete) khóa học. POST /api/v1/courses, PUT /api/v1/courses/{id}, DELETE semantics = soft-delete.

**FR002 - Quản lý module & lesson**

Instructor thêm/sửa/xóa module và lesson thuộc course; lesson có contentType, contentUrl, duration; DELETE = soft-delete.

**FR003 - Ghi danh khóa học (Enroll)**

Learner có thể ghi danh vào course; hệ thống phải ngăn ghi danh trùng (unique constraint CourseId+UserId). Endpoint POST /api/v1/courses/{id}/enroll.

**FR004 - Bắt đầu & nộp bài kiểm tra (Quiz)**

Learner có thể start quiz và submit answers. Hệ thống tạo QuizAttempt (snapshot), tính điểm, lưu AttemptAnswers và cập nhật tiến độ nếu đạt yêu cầu.

**FR005 - Theo dõi tiến độ (Progress)**

Hệ thống lưu Progress cho từng lesson (CompletedAt, Percent). Learner có thể mark lesson completed; đảm bảo idempotency.

**FR006 - Thông báo (Notification)**

Admin/Instructor gửi notification tới user hoặc course; lưu Notification vào DB; hỗ trợ gửi email qua Outbox pattern hoặc immediate send.

**FR007 - Quản lý file tài nguyên (FileAssets)**

Instructor upload file; metadata lưu vào FileAssets; file lưu trên storage (Ví dụ Azure Blob). Hạn mức kích thước và MIME type phải được kiểm tra.

**FR008 - Báo cáo & export**

Cung cấp endpoint báo cáo tiến độ, điểm số, hỗ trợ export CSV với safety (escape, BOM).

**FR009 - User & Auth**

Hệ thống dùng ASP.NET Identity, JWT Bearer cho API; role-based access control (Admin, Instructor, Learner).

# Yêu cầu phi chức năng (Non-Functional Requirements)

**NFR001 - Audit & Soft Delete**

Mọi entity chính kế thừa BaseAuditable (CreatedAt, CreatedBy, UpdatedAt, UpdatedBy). Xóa mềm bằng IsDeleted + DeletedAt + DeletedBy; áp dụng Global Query Filter trong EF Core.

**NFR002 - Timezone & Clock**

Lưu tất cả thời gian ở UTC; hiển thị theo timezone user ở UI; dùng IDateTimeProvider để testable.

**NFR003 - Concurrency**

Sử dụng optimistic concurrency (RowVersion) cho entity thường xuyên chỉnh sửa; trả 412 Precondition Failed khi mismatch; hỗ trợ ETag/If-Match.

**NFR004 - Idempotency & Consistency**

Các POST gây side-effect (Enroll, StartQuiz, CreateCourse, SendNotification) hỗ trợ Idempotency-Key; dùng Outbox để đảm bảo at-least-once delivery.

**NFR005 - Bảo mật**

Auth: JWT/ASP.NET Identity. Các endpoint nhạy cảm require RBAC. Không lưu mật khẩu ở plaintext. Mask/encrypt PII trong logs.

**NFR006 - Hiệu năng & Thông lượng**

Pagination (page,size), size max=100; endpoints nặng (reports/files) cần rate-limiting và caching; tối đa response time SLA cho các API thường là < 500ms under normal load (tùy cấu hình).

**NFR007 - Khả năng mở rộng & Reliability**

Sử dụng Outbox + background worker cho gửi email; retry policy, monitoring backlog; health checks.

**NFR008 - Accessibility (A11y)**

Tuân thủ ARIA attributes, keyboard navigation, visible focus ring, contrast >= 4.5:1, toast/alerts dùng aria-live.

**NFR009 - i18n & Localization**

Hỗ trợ đa ngôn ngữ; đọc Accept-Language; message.error có mã cố định và thông điệp dịch theo locale.

**SRS.NFR & SDD.Infra - Security / Token Policy**

* **Access Token (JWT):**
  + Algorithm: RS256 (recommended) or HS256 for simple setup.
  + Lifetime: **15 minutes** (recommended) — hoặc 60 minutes nếu muốn ít refresh.
  + Claims: sub (user id GUID), roles (array), iat, exp, jti (for idempotency correlation optional).
* **Refresh Token:**
  + Lifetime: **14 days** (or 30 days), stored hashed in DB with userId and expiry.
  + Flow: /auth/refresh accepts refresh token; issues new access token + optionally new refresh token (rotate and revoke old).
  + Revoke on password change / logout.
* **Password policy:**
  + Min length: **8**; require 1 uppercase, 1 lowercase, 1 digit, 1 special char.
  + Store using salted hashing (e.g. ASP.NET Identity default uses PBKDF2).
* **Token storage (client):**
  + Access token: in memory (never localStorage) for SPA; refresh token: HttpOnly Secure cookie (if web) or secure storage in mobile.
* **Brute-force / Rate-limits:**
  + Max failed login attempts before lockout: **5 attempts**, lockout duration **15 minutes**.
* **Logging & PII:** mask PII in logs, store audit logs encrypted at rest if needed.

**NFR Định lượng (Quantitative Targets)**

* API latency: median < **200 ms**, 95th percentile < **1s** under normal load.
* Availability (target): **99.5%** monthly uptime.
* Concurrent active learners (expected): **100–300** (điều chỉnh theo quy mô).
* Outbox retention: keep unprocessed messages **30 days** then archive.
* Audit logs retention: **1 year** (or as per policy).
* Max file upload size: **100 MB**; recommend chunked upload for > 20 MB.

# Ràng buộc & Giả định

* Database: SQL Server với EF Core.
* File storage: Azure Blob (hoặc S3) cho file assets.
* SMTP/3rd-party để gửi email.
* Người dùng đọc tài liệu này giả định rằng SDD kèm theo (SDD.docx) mô tả thiết kế chi tiết.
* Múi giờ hệ thống là UTC; các client chịu trách nhiệm hiển thị theo timezone.

# Traceability & Mapping to SDD

Các yêu cầu trong tài liệu này được rút ra từ SDD gốc (SDD.docx). Vui lòng mở SDD.docx để tham chiếu chi tiết kiến trúc, database schema, API contract và flow diagrams.

# Acceptance Criteria (Chi tiết theo FR)

* Các endpoint chức năng trả đúng mã HTTP và envelope response như mô tả (success, data, error, traceId).
* Soft-delete và audit hoạt động (CreatedAt/CreatedBy/IsDeleted) cho entity chính.
* Quiz: scoring phải đúng theo logic (score >= 70% => passed) và lưu QuizAttempt + AttemptAnswers.
* Idempotency: duplicate requests với cùng Idempotency-Key không tạo ra side-effect lặp.

## FR001 - Quản lý khóa học

Endpoints / Area: /api/v1/courses

Mô tả ngắn: Create/Update/Publish/Delete (soft-delete) courses

**Acceptance Criteria:**

1. Success scenario: Valid request + proper authorization => success HTTP code (200/201/204) + data persisted and traceId in response.
2. Validation errors: Invalid input => 400 with ERR\_VALIDATION and details.
3. Authorization: Insufficient permissions => 403 ERR\_FORBIDDEN.
4. Idempotency/Conflict: Duplicate side-effect requests with same Idempotency-Key should be handled idempotently or return 409 ERR\_CONFLICT. Updates require If-Match/ETag.

## FR002 - Quản lý module & lesson

Endpoints / Area: /api/v1/courses/{courseId}/modules  
/api/v1/modules/{moduleId}/lessons

Mô tả ngắn: Manage modules and lessons

**Acceptance Criteria:**

1. Success scenario: Valid request + proper authorization => success HTTP code (200/201/204) + data persisted and traceId in response.
2. Validation errors: Invalid input => 400 with ERR\_VALIDATION and details.
3. Authorization: Insufficient permissions => 403 ERR\_FORBIDDEN.
4. Edge cases: Operations on soft-deleted resources return 404 ERR\_NOT\_FOUND.

## FR003 - Ghi danh khóa học (Enroll)

Endpoints / Area: /api/v1/courses/{id}/enroll

Mô tả ngắn: Enroll a learner in a course

**Acceptance Criteria:**

1. Success scenario: Valid request + proper authorization => success HTTP code (200/201/204) + data persisted and traceId in response.
2. Validation errors: Invalid input => 400 with ERR\_VALIDATION and details.
3. Authorization: Insufficient permissions => 403 ERR\_FORBIDDEN.
4. Idempotency/Conflict: Duplicate side-effect requests with same Idempotency-Key should be handled idempotently or return 409 ERR\_CONFLICT. Updates require If-Match/ETag.

## FR004 - Quiz (Start & Submit)

Endpoints / Area: /api/v1/quizzes/{id}/start  
/api/v1/quizzes/{id}/submit

Mô tả ngắn: Start and submit quizzes, grade automatically

**Acceptance Criteria:**

1. Success scenario: Valid request + proper authorization => success HTTP code (200/201/204) + data persisted and traceId in response.
2. Validation errors: Invalid input => 400 with ERR\_VALIDATION and details.
3. Authorization: Insufficient permissions => 403 ERR\_FORBIDDEN.
4. Idempotency/Conflict: Duplicate side-effect requests with same Idempotency-Key should be handled idempotently or return 409 ERR\_CONFLICT. Updates require If-Match/ETag.

## FR005 - Theo dõi tiến độ (Progress)

Endpoints / Area: /api/v1/lessons/{id}/complete  
/api/v1/courses/{id}/progress

Mô tả ngắn: Mark lesson complete and view progress

**Acceptance Criteria:**

1. Success scenario: Valid request + proper authorization => success HTTP code (200/201/204) + data persisted and traceId in response.
2. Validation errors: Invalid input => 400 with ERR\_VALIDATION and details.
3. Authorization: Insufficient permissions => 403 ERR\_FORBIDDEN.
4. Edge cases: Operations on soft-deleted resources return 404 ERR\_NOT\_FOUND.

## FR006 - Thông báo (Notification)

Endpoints / Area: /api/v1/notifications

Mô tả ngắn: Create/send notifications to users or course

**Acceptance Criteria:**

1. Success scenario: Valid request + proper authorization => success HTTP code (200/201/204) + data persisted and traceId in response.
2. Validation errors: Invalid input => 400 with ERR\_VALIDATION and details.
3. Authorization: Insufficient permissions => 403 ERR\_FORBIDDEN.
4. Idempotency/Conflict: Duplicate side-effect requests with same Idempotency-Key should be handled idempotently or return 409 ERR\_CONFLICT. Updates require If-Match/ETag.

## FR007 - Quản lý FileAssets

Endpoints / Area: /api/v1/files

Mô tả ngắn: Upload/download file assets

**Acceptance Criteria:**

1. Success scenario: Valid request + proper authorization => success HTTP code (200/201/204) + data persisted and traceId in response.
2. Validation errors: Invalid input => 400 with ERR\_VALIDATION and details.
3. Authorization: Insufficient permissions => 403 ERR\_FORBIDDEN.
4. Edge cases: Operations on soft-deleted resources return 404 ERR\_NOT\_FOUND.

## FR008 - Báo cáo & Export CSV

Endpoints / Area: /api/v1/reports/\*

Mô tả ngắn: Course reports, export CSV

**Acceptance Criteria:**

1. Success scenario: Valid request + proper authorization => success HTTP code (200/201/204) + data persisted and traceId in response.
2. Validation errors: Invalid input => 400 with ERR\_VALIDATION and details.
3. Authorization: Insufficient permissions => 403 ERR\_FORBIDDEN.
4. Edge cases: Operations on soft-deleted resources return 404 ERR\_NOT\_FOUND.

## FR009 - User & Auth

Endpoints / Area: /api/v1/auth/\*

Mô tả ngắn: Authentication and basic user management

**Acceptance Criteria:**

1. Success scenario: Valid request + proper authorization => success HTTP code (200/201/204) + data persisted and traceId in response.
2. Validation errors: Invalid input => 400 with ERR\_VALIDATION and details.
3. Authorization: Insufficient permissions => 403 ERR\_FORBIDDEN.
4. Edge cases: Operations on soft-deleted resources return 404 ERR\_NOT\_FOUND.

# API Endpoints & DTOs (Mẫu)

Headers (áp dụng chung):  
- Authorization: Bearer <JWT>  
- Idempotency-Key: <string> (for POST with side-effects)  
- If-Match: "<rowversion-base64>" (for PUT/PATCH concurrency)  
- Accept-Language: <locale>

## POST /api/v1/courses

Create a new course. Admin/Instructor only.

Notes: Returns 201 Created. Idempotency-Key optional; 409 if Code already exists.

C# DTO (example):

public record CreateCourseRequest(string Code, string Title, string? Description, string Status);

Request (JSON):

{  
 "code": "CS101",  
 "title": "Intro to Programming",  
 "description": "Short description",  
 "status": "Draft"  
}

Response (JSON):

{  
 "success": true,  
 "data": { "id": "GUID" },  
 "error": null,  
 "traceId": "..."   
}

## PUT /api/v1/courses/{id}

Update course. Requires If-Match for concurrency.

Notes: Returns 204 No Content on success; 412 if ETag mismatch; 403 if not authorized.

C# DTO (example):

public record UpdateCourseRequest(string Title, string? Description, string Status);

Request (JSON):

{  
 "title": "Updated Title",  
 "description": "...",  
 "status": "Published"  
}

Response (JSON):

{  
 "success": true,  
 "data": null,  
 "error": null,  
 "traceId": "..."  
}

## DELETE /api/v1/courses/{id}

Soft-delete course (IsDeleted = true). Admin/Instructor.

Notes: Returns 204 No Content. To hard-delete use ?force=true and Admin only.

## POST /api/v1/courses/{id}/enroll

Enroll current authenticated learner into the course.

Notes: Idempotency-Key required; returns 200 on success, 409 if already enrolled.

C# DTO (example):

public record EnrollRequest(Guid? UserId);

Request (JSON):

{  
 "userId": "GUID" // optional if using current token  
}

Response (JSON):

{  
 "success": true,  
 "data": null,  
 "error": null,  
 "traceId": "..."  
}

## POST /api/v1/quizzes/{id}/start

Start a quiz; creates QuizAttempt if allowed and returns attemptId + questions (snapshot).

Notes: Idempotency-Key required; returns 409 if attempts exhausted.

Response (JSON):

{  
 "attemptId": "GUID",  
 "expireAt": "2025-08-30T10:00:00Z",  
 "questions": [ { "questionId": "GUID", "text": "...", "options": [ {"id":"GUID","text":"..."} ] } ]  
}

## POST /api/v1/quizzes/{id}/submit

Submit quiz answers; service validates, calculates score and returns result.

Notes: Validates attempt ownership and state; idempotent by attemptId.

Request (JSON):

{  
 "attemptId": "GUID",  
 "answers": [ { "questionId": "GUID", "optionId": "GUID" } ]  
}

Response (JSON):

{  
 "attemptId": "GUID",  
 "score": 85.5,  
 "correctCount": 17,  
 "total": 20,  
 "passed": true  
}

## POST /api/v1/lessons/{id}/complete

Mark a lesson completed by current user.

Notes: Idempotent: calling multiple times does not create duplicates.

Request (JSON):

{}

Response (JSON):

{  
 "success": true,  
 "data": { "percent": 40 },  
 "error": null,  
 "traceId": "..."  
}

## POST /api/v1/notifications

Create a notification targeted to user(s) or course. Can be queued via Outbox.

Notes: Returns 200/202; requires Idempotency-Key for bulk notifications.

Request (JSON):

{  
 "toUserId": "GUID?",  
 "courseId": "GUID?",  
 "title": "Exam scheduled",  
 "body": "Exam on Monday"  
}

## POST /api/v1/files

Upload file (multipart/form-data). Returns metadata and signed URL.

Notes: Validate MIME, max size (e.g., 100MB).

Response (JSON):

{  
 "id": "GUID",  
 "fileName": "notes.pdf",  
 "storagePath": "blob://...",  
 "url": "https://.../signed-url"  
}

## GET /api/v1/reports/course/{id}/progress?from=&to=&format=csv

Get course progress report; supports Accept: text/csv or ?format=csv

Notes: Apply rate-limiting for heavy reports; set caching where possible.

## POST /api/v1/auth/login

Authenticate user and return JWT + refresh token.

C# DTO (example):

public record LoginRequest(string Username, string Password);

Request (JSON):

{  
 "username": "user@example.com",  
 "password": "P@ssw0rd"  
}

Response (JSON):

{  
 "accessToken": "<jwt>",  
 "refreshToken": "<token>",  
 "expiresIn": 3600  
}

# Phụ lục

Tham khảo: SDD.docx Các chi tiết kỹ thuật (UQ, FK, Index, DTOs, API contract) nằm trong SDD.  
  
Nếu cần, có thể mở rộng SRS thành định dạng formal hơn (use case diagrams, detailed functional specs per endpoint, sequence test cases).