Syllabus Management and Digitalization System of the University (SMD)

Group Members		
Supervisor		
Capstone Project code		

- Ho Chi Minh, May 2025 -

Table of Contents

Trần Thị Hồng Anh	083306005553	honganh20963@gmail.com

2. Existing Systems

2.1 Traditional Syllabus Management (PDF/Word-based)

Description:

In traditional approaches, course syllabi are created and stored as Word or PDF documents. These files are usually shared via email, institutional websites, or cloud storage platforms such as Google Drive. Updates are handled manually by instructors or academic staff, and there is little system support for managing changes.

Drawbacks:

- Difficult to control versions, leading to outdated or inconsistent syllabi
- Lack of interactivity and integration with teaching and learning activities
- Limited search, comparison, and data analysis capabilities
- Time-consuming manual updates and approval processes
- No structured linkage between learning outcomes, content, and assessments

---

2.2 Learning Management Systems (LMS)

Description:

Learning Management Systems (LMS) are digital platforms designed to manage courses, learners, instructional materials, assignments, and assessments. Popular systems such as Moodle, Canvas, and Blackboard support online and blended learning and allow instructors to track student progress.

Drawbacks:

- Primarily focused on course delivery rather than syllabus design and management
- Complex interfaces that require training for effective use
- Limited support for syllabus standardization across departments or programs
- Difficult to customize for academic governance and accreditation requirements
- High implementation and maintenance costs for some institutions

---

3. Business Opportunity

The rapid digital transformation in education has increased the demand for standardized, transparent, and data-driven curriculum management. Educational institutions require a specialized solution to efficiently manage, update, and analyze syllabi across programs. This creates a strong business opportunity for a dedicated syllabus management platform that fills the gap left by traditional document-based methods and general-purpose LMS platforms.

---

4. Software Product Vision

The vision is to develop a modern, intelligent syllabus management platform that enables educational institutions to easily create, update, standardize, and analyze course syllabi. The system aims to align learning outcomes, instructional content, and assessment methods while supporting quality assurance, accreditation processes, and continuous curriculum improvement.

5. Project Scope & Limitations

5.1 Major Features

5.1.1 Major Features for Admin

- **FE-01: Login / Logout**
- **FE-02: User Management** – Create, update, assign roles, and manage user accounts.
- **FE-03: System Configuration** – Manage academic terms, CLO/PLO templates, grading schemes, and workflow rules.
- **FE-04: Publishing Management** – Control syllabus approval, publication, unpublishing, and archiving.
- **FE-05: System Audit Log** – Track system activities for auditing and compliance purposes.

5.1.2 Major Features for Head of Department (HoD)

- **FE-01: Login / Logout**
- **FE-02: Syllabus Review & Approval (Level 1)** – Review academic content and CLO compliance.
- **FE-03: Collaborative Review Management** – Manage department-level consultation and feedback.
- **FE-04: Lookup & Analysis** – Compare syllabuses across courses and academic years.
- **FE-05: Notification** – Receive alerts on submissions and review outcomes.

5.1.3 Major Features for Lecturer

- **FE-01: Login / Logout**
- **FE-02: Create Syllabus** – Enter syllabus content with structured metadata (CLO, PLO, assessments, prerequisites).
- **FE-03: Update Syllabus** – Edit existing syllabuses and submit new versions for review.
- **FE-04: Syllabus Management** – Search, filter, compare syllabus versions, and follow syllabus updates.
- **FE-05: Collaborative Review** – Participate in peer feedback and comment on syllabuses.
- **FE-06: Notification** – Receive real-time alerts on review status and feedback.

5.1.4 Major Features for Student

- **FE-01: Search Syllabus** – Search by course code, course name, or program.
- **FE-02: View Syllabus Detail** – View full syllabus content, AI summary, and CLO–PLO mapping.
- **FE-03: Subscribe / Follow** – Receive notifications when syllabus content is updated.
- **FE-04: Feedback** – Report errors or issues in syllabus content.

5.1.5 Major Features for Academic Affairs (AA)

- **FE-01: Login / Logout**
- **FE-02: Academic Approval (Level 2)** – Verify syllabus alignment with program learning outcomes (PLO).
- **FE-03: Program & Curriculum Management** – Manage PLO standards and curriculum structure.

- **FE-04: Lookup & Analysis** – Analyze syllabus consistency and curriculum alignment.
- **FE-05: Notification** – Receive system alerts related to approvals.

5.1.6 Major Features for Principal

- **FE-01:** Final Strategic Approval Final approval of important academic documents/proposals.
- **FE-02:** System Oversight View system overview, reports and operational status.

6.2 Limitations & Exclusions

- **LI-1: Legacy Data Constraints**
  Historical syllabuses stored in unstructured formats may require manual verification after digitization.
- **LI-2: AI Accuracy Limitation**
  AI-based CLO–PLO mapping and semantic analysis provide decision support but do not replace human academic judgment.
- **LI-3: LMS Integration Scope**
  Integration with LMS platforms is limited to syllabus linking and metadata synchronization, not full course content management.
- **LI-4: Performance Constraints**
  Advanced AI processing tasks are executed asynchronously and may not provide immediate results.

II. Project Management Plan

1. Overview

1.1 Scope & Estimation

#	WBS Item	Complexity	Est. Effort (manday)
1	**Initiating**		**20**
1.1	Define project scope	Medium	12
1.2	Collect requirements	Medium	8
2	**Planning**		**14**
2.1	Create kick-off meeting	Medium	7
2.2	Create plan document	Medium	7
3	**Executing**		
3.1	**Analysis**		**18**

3.1.1	Analysis requirements	Medium	8
3.1.2	Feasibility Analysis	Complex	10
3.2	**Design**		**12**
3.2.1	Design conceptual ERD	Medium	4
3.2.2	Design code architecture	Medium	4
3.2.3	Design web application	Medium	4
3.3	**Implementation**		**52**
3.3.1	User authentication and role-based authorization	Medium	6
3.3.2	Syllabus creation and update module	Medium	8
3.3.3	Syllabus version control management	Complex	8
3.3.4	Approval and review workflow (Lecturer → HoD → Academic Affairs → Principal)	Complex	10
3.3.5	Syllabus search and comparison feature	Medium	6
3.3.6	CLO–PLO mapping and validation	Medium	6

3.3.7	AI integration (change detection, content summary)	Complex	8
3.3.8	**Project Progress Management**		**7**
3.3.8.1	Academic Affairs manage review schedules	Medium	2
3.3.8.2	Lecturer submit syllabus review checklist	Simple	1
3.3.8.3	Lecturer update syllabus after review round 1 & 2	Simple	1
3.3.8.4	HoD review and respond to syllabus updates	Simple	1
3.3.8.5	Academic Affairs review and respond after HoD approval	Simple	1
3.3.8.6	Final approver provide feedback	Simple	1
3.3.9	**Contribution & Activity Tracking**		**4**
3.3.9.1	Track lecturer contributions and syllabus changes	Simple	4

3.3.10	**Academic Management**		**4**
3.3.10.1	Manage syllabus approval timeline	Medium	2
3.3.10.2	Update syllabus approval results	Simple	1
3.3.10.3	Lecturer view syllabus approval status	Simple	1
3.3.11	**Notification System**		**9**
3.3.11.1	Send syllabus update notifications	Complex	5
3.3.11.2	View notification	Medium	2
3.3.11.3	Admin create system notifications	Medium	2
3.4	**Testing**		**24**
3.4.1	Unit Testing	Complex	8
3.4.2	Integration Testing	Complex	8
3.4.3	System Testing	Complex	8
3.5	**Monitoring and Controlling**		**24**
3.5.1	Control project process	Complex	12
3.5.2	Track performance and quality	Complex	12

1.2 Project Objectives

- **Timeline:** The project must be completed before **30 April, 2025**
- **Allocated Effort (Man-days): 232**

- **Defect Prediction and Quality Objectives:** The project aims to minimize defects through staged testing and early detection of system issues.

#	Testing Stage	Test Coverage	No. of Defects	% of Defect	Notes
1	Reviewing	**Backend:** Code review (C#, Authentication, Authorization) **Frontend:** Code review (Next.js, React components, State management)	< 30	5%	Focus on security vulnerabilities, performance bottlenecks, and code maintainability
2	Unit Testing	**Backend:** Business logic, validation rules **Frontend:** Component testing, form validation, UI state handling	< 20	3%	Ensure correctness of authentication, approval logic, and UI interactions
3	Integration Testing	**Backend:** API integration, workflow processing **Frontend:**	< 5	1%	Verify consistency between modules, workflow

		API calls, data binding, UI responsiveness			accuracy, and system stability

1.3 Project Risks

#	Risk Category	Potential Errors / Risks	Impact Level	Probability	Mitigation Strategy
1	Requirements Risk	Unclear or changing requirements for syllabus approval workflow	High	Medium	Conduct frequent requirement reviews and stakeholder validation
2	Authorization & Security	Incorrect role-based access control leading to unauthorized actions	High	Medium	Implement strict RBAC testing and security reviews
3	Version Control	Data conflicts or loss when multiple users edit syllabi simultaneously	High	Medium	Apply locking mechanisms and version comparison features
4	Workflow Processing	Incorrect status transitions in multi-level	High	Low	Perform scenario-based integration testing

		approval process			
5	AI Integration	Inaccurate change detection or unreliable content summaries	Medium	Medium	Validate AI outputs and provide manual override options
6	Performance	Slow system response during peak usage periods	Medium	Low	Optimize database queries and conduct performance testing
7	Notification System	Missed or delayed notifications causing approval delays	Medium	Medium	Implement retry mechanisms and notification logging
8	Testing Coverage	Undetected defects due to insufficient test scenarios	High	Low	Expand test cases, especially for edge cases
9	Schedule & Resource	Underestimation of effort for complex features (workflow, AI)	Medium	Medium	Continuous monitoring and adjustment during execution

2. Management Approach

2.1 Project Process

The **Syllabus Management and Digitalization System (SMD)** project is implemented using a controlled software development process, combining an enhanced **Waterfall model** with iterative activities during the execution phase to ensure quality and schedule adherence.

The process consists of the following main phases:

Initiation:
Define project objectives, scope, stakeholders, and align on the overall direction.

Planning:
Develop a detailed plan covering scope of work (WBS), resource allocation, time estimation, risk assessment, and required resources.

Analysis & Design:
Analyze business requirements, develop the SRS documentation, design the system architecture, ERD, syllabus approval workflows, and AI-supported functionalities.

Implementation:
Develop core modules including syllabus management, approval workflows, search and comparison, user authorization, and AI integration.

Testing:
Conduct Unit Testing, Integration Testing, and System Testing to ensure the system meets requirements.

Monitoring & Controlling:
Track progress, control quality, manage risks, and adjust plans as necessary.

2.2 Quality Management

To minimize potential risks during system implementation, the project should standardize the process of requirement elicitation and management, and regularly review and update requirements with stakeholders to avoid business misunderstandings. From a technical perspective, the system must be designed with robust authentication, role-based authorization, and version control mechanisms to ensure data security and prevent conflicts during concurrent edits. Comprehensive and continuous testing, including functional, workflow, and performance testing, should be conducted to detect defects early before official deployment. For AI-related features, human validation should be incorporated, along with contingency plans to address inaccuracies or dependency on external services. In addition, close progress monitoring, automated notifications within workflows, and effective coordination among stakeholders will help reduce delays and ensure the overall quality and success of the project.

2.3 Training Plan

Training Content	Participants	Duration
System overview, business workflow, and user roles	All stakeholders	

System administration training (user management, system configuration, role-based access control)	System Administrator (Admin)	
Lecturer training (create, update, and submit syllabus for approval)	Lecturers	
Academic review and approval process training	Heads of Department, Academic Affairs staff	

3. Project Deliverables

The project deliverables represent all tangible outputs created during the development of the Syllabus Management and Digitalization System (SMD). These deliverables ensure that the project fulfils academic requirements, supports effective implementation, and provides a complete record of outputs for evaluation.

The primary deliverables include:

- **Project documentation:** including the Project Introduction, Software Requirement Specification, Software Design Document, Test Documents, User Guide, and Final Report.
- **Software system deliverables:** including the functional syllabus management system, approval workflow, version control, CLO–PLO mapping, search and analytics features, and notification functions.
- **Testing deliverables:** including test scripts, unit test results, integration test results, system test results, and defect tracking.
- **Deployment deliverables:** including deployable system build, database structure, configuration documentation, backup procedures, and installation instructions.

All deliverables are stored on shared project platforms and maintained under version control to ensure accuracy, traceability, and accessibility.

4. Responsibility Assignments (D~Do; R~Review; S~Support;)

Responsibility	Ngọc Loan	Hồng Anh	Thanh Dung	Quốc Triệu	Kim Phương	Minh Tân	Tấn Phát
Project Planning & Tracking	D	R	R	R	S	S	S

Prepare Project Introduction Document	D	D	D	D	D	D	D
Prepare Software Requirement Document	R	D	D	D	S	S	S
Prepare Software Design Document	D	D	D	D	D	D	D
Prepare Test Document	R	R	D	D	R	S	S
Prepare Software User Guides	D	R	S	D	R	S	S
Prepare Final Report	D	D	D	D	D	D	D

5. Project Communications

Communication Item	Who/ Target	Purpose	When, Frequency	Type, Tool, Method(s)
Daily Task Sync	Team members	Share daily progress, identify blockers and	Daily (short meeting)	Online – Google

		coordinate implementation tasks		Meet / Chat
Issue Resolution Meeting	Related members	Discuss and resolve urgent issues affecting system functionality or schedule	When critical issues occur	Online – Google Meet
Progress Update Report	Team members	Report project progress, completed features, issues, risks and next-week plan	Weekly	Online - chat

6. Configuration Management

Configuration Management ensures that system assets—including documents, source code, and deployment environments—are controlled and maintained consistently throughout the project lifecycle. It enables traceability, prevents unauthorized changes, and supports reliable system development.

6.1 Document Management

All project documents are managed using Confluence and Google Drive to ensure version control, accessibility, and secure storage.

Documents must follow:

- Standard naming conventions
- Defined version numbering (v1.0, v1.1, v2.0, etc.)
- Access control rules
- Backup and archival procedures

Document changes are reviewed and approved by the assigned owner before updates are published. This ensures consistency, prevents information loss, and maintains transparency across the team.

6.2 Source Code Management

All code for the SMD system is managed using GitHub to maintain traceability, prevent conflicts, and ensure safe collaboration.

Source code practices include:

- Branch-based development
- Code review before merge
- Commit message standards
- Git issue tracking system
- Change history documentation

The Git repository structure supports organized development, stable releases, and easy rollback if required.

6.3 Tools & Infrastructures

The project uses a combination of cloud-based tools and software services to support collaboration, development, testing, and deployment.

Key tools include:

Purpose	Platform	Description
Documentation	Confluence	Stores shared project files and reports
Source control	GitHub	Manages coding, branching, and versioning
Project management	Jira	Tracks tasks, issues, and progress
Storage/backup	Google Drive	Archives documents and deliverables
CI/CD	GitHub Actions	Supports automated build and testing

These infrastructures ensure that the project environment remains consistent, secure, and scalable throughout development.

III. Software Requirement Specification

Section	**Key PointsObjective**
Objective	Automate core library operations: loan/return, member, and catalog management.
Scope	**In:** Member management, online catalog search, automated circulation (Web). **Out:** Online payments, native mobile apps, accounting system integration.
Users	Admins, Library Staff, Readers, Management, IT Support, External Systems.
Core Features	1. Catalog Management 2. Member Management

	3. Automated Circulation (Scanning) 4. Advanced Search 5. Reporting & Export
Tech Requirements	Supports Chrome, Firefox, Safari; API response <2s; 1000 concurrent users; privacy law compliance.
Prerequisites	**Assumptions:** Stable internet, trained staff. **Dependencies:** Auth Service (LDAP/AD), Email Server.
Documentation	User Manual, API Docs, Deployment & Maintenance Guides, Test Suite.

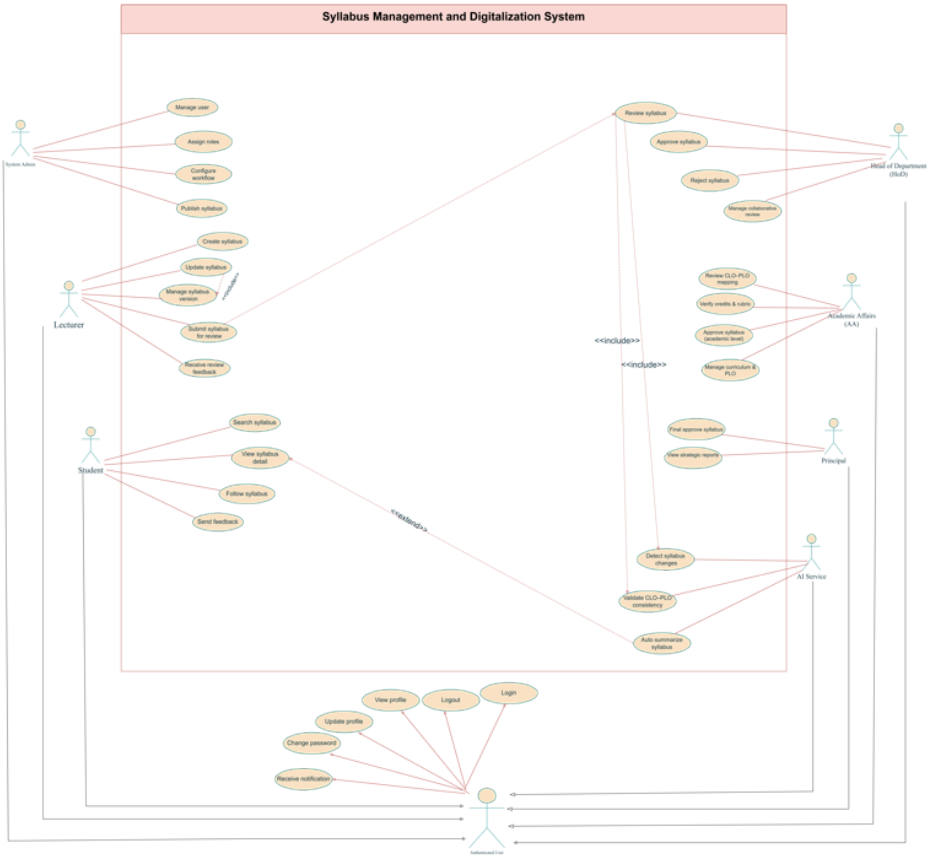
2. User Requirements

2.1 Actors

#	Actor	Description
1	Admin	Responsible for managing the entire platform, including user account approvals, maintaining system security, and ensuring smooth operation.
2	Manager	Oversees the thesis process and manages topic approval periods, defense schedules, and capstone results.
3	Lecturer	Acts as a mentor, reviewer, and council member in evaluating student projects.
4	Student	The primary user is responsible for forming teams, submitting topics, and managing project progress.

2.2 Use Cases

2.2.1 Diagram(s)



2.2.2 Descriptions

ID	Use Case	Actors	Use Case Description
01	Login	Authenticated user	User logs into the syllabus management system
02	Logout	Authenticated user	User logs out of the system
03	View profile	Authenticated user	Users view their personal profile information
04	Update profile	Authenticated user	Users update personal profile details
05	Change password	Authenticated user	Users change their account password

06	Receive notification	Authenticated user	Users receive system notifications
07	Manage users	System Admin	Admin manages user accounts in the system
08	Assign roles	System Admin	Admin assigns roles and permissions to users
09	Configure workflow	System Admin	Admin configures the syllabus approval workflow
10	Publish syllabus	System Admin	Admin publishes approved syllabi
11	Create syllabus	Lecturer	Lecturer creates a new syllabus
12	Update syllabus	Lecturer	Lecturer edits an existing syllabus
13	Manage syllabus versions	Lecturer	Lecturer manages different versions of a syllabus
14	Submit syllabus for review	Lecturer	Lecturer submits syllabus for approval
15	Receive review feedback	Lecturer	Lecturer receives feedback from reviewers
16	Review syllabus	Head of Department (HoD)	HoD reviews submitted syllabi
17	Approve syllabus	Head of Department	HoD approves the syllabus

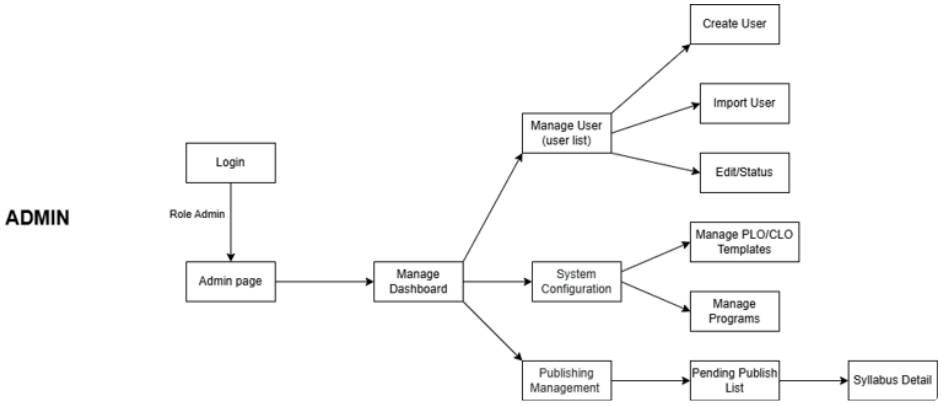
		(HoD)	
18	Reject syllabus	Head of Department (HoD)	HoD rejects the syllabus with comments
19	Manage syllabus workflow	Head of Department (HoD)	HoD manages the syllabus review process
20	Review CLO–PLO	Academic Affairs	Academic Affairs reviews CLO–PLO mapping
21	Approve academic syllabus	Academic Affairs	Academic Affairs approves academic content
22	Manage CLO–PLO	Academic Affairs	Academic Affairs manages CLO–PLO alignment
23	Final approve syllabus	Principal	Principal gives final approval for the syllabus
24	View statistics reports	Principal	Principal views syllabus statistics and reports
25	Search syllabus	Student	Student searches for syllabi
26	View syllabus	Student	Student views syllabus details
27	Follow syllabus	Student	Student follows syllabus updates
28	Send feedback	Student	Student sends feedback on syllabus

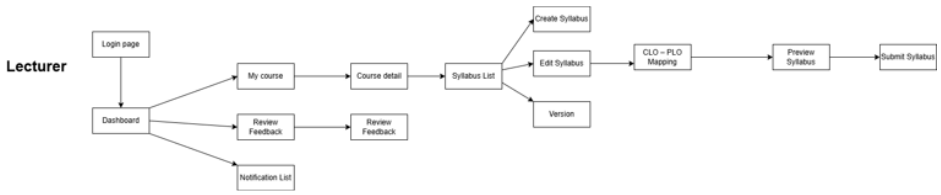
29	Detect syllabus changes	AI Service	AI detects changes in the syllabus
30	Validate CLO–PLO consistency	AI Service	AI validates CLO–PLO consistency
31	Auto summarize syllabus	AI Service	AI automatically summarizes syllabus content

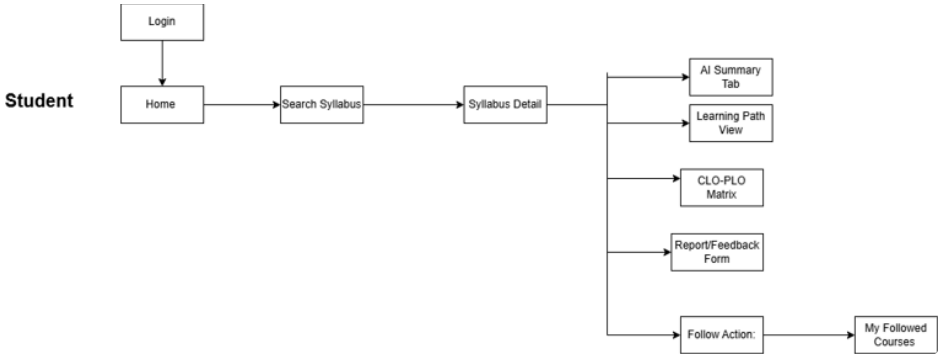
3.Functional Requirements

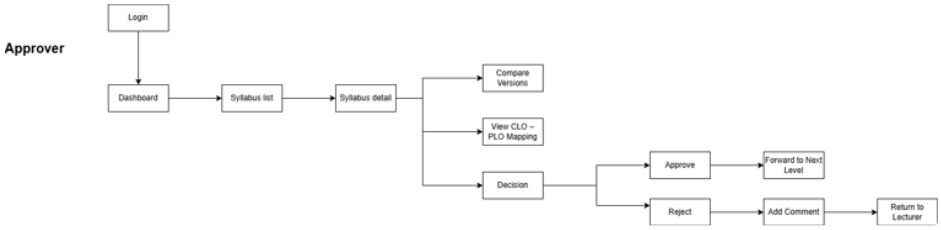
3.1 System Functional Overview

3.1.1 Screens Flow









3.1.2 Screen Descriptions

#	Feature	Screen	Description
1	Authentication	Login Page	Allows users (Admin, Lecturer, Student, Approver) to log into the system using their credentials.
2	User Management	Admin Page	Provides access to administrative functions after successful login.
3	Dashboard Management	Manage Dashboard	Displays system overview and provides navigation to management features.
4	User Management	Manage User (User List)	Allows admin to view, search, and manage system users.
5	User Creation	Create User	Allows admin to create new user accounts and assign roles.
6	User Import	Import User	Allows admin to import multiple users into the

			system via file upload.
7	User Update	Edit / Status	Allows admin to edit user information and manage account status.
8	System Configuration	System Configuration	Allows admin to configure academic and system settings.
9	Academic Configuration	Manage PLO/CLO Templates	Allows admin to manage PLO and CLO template definitions.
10	Program Management	Manage Programs	Allows admin to create and manage academic programs.
11	Publishing Control	Publishing Management	Manages the syllabus publishing workflow.
12	Publish Review	Pending Publish List	Displays a list of syllabi pending publication.
13	Syllabus View	Syllabus Detail	Displays detailed syllabus information for review.
14	Course Management	Dashboard (Lecturer)	Displays lecturer overview and access to assigned courses.
15	Course List	My Course	Allows lecturers to view courses they are responsible for.
16	Course Detail	Course Detail	Displays detailed information about a selected course.
17	Syllabus Management	Syllabus List	Displays all syllabi related to a course and their versions.
18	Syllabus Creation	Create Syllabus	Allows lecturers to create a new syllabus.
19	Syllabus Editing	Edit Syllabus	Allows lecturers to update syllabus content.

20	Version Control	Version	Allows lecturers to manage and view syllabus versions.
21	Outcome Mapping	CLO–PLO Mapping	Allows lecturers to map Course Learning Outcomes to Program Learning Outcomes.
22	Syllabus Preview	Preview Syllabus	Allows lecturers to preview syllabus before submission.
23	Submission	Submit Syllabus	Allows lecturers to submit syllabus for approval.
24	Feedback Review	Review Feedback	Allows lecturers to view feedback from approvers.
25	Notification	Notification List	Displays system notifications for lecturers.
26	Student Home	Home	Provides students with access to syllabus search and navigation.
27	Syllabus Search	Search Syllabus	Allows students to search for syllabi by course or program.
28	Syllabus Exploration	Syllabus Detail	Displays syllabus details for students.
29	AI Support	AI Summary Tab	Provides AI-generated summary of syllabus content.
30	Learning Guidance	Learning Path View	Displays recommended learning paths for students.
31	Outcome Visualization	CLO–PLO Matrix	Shows CLO–PLO mapping matrix for students.
32	Feedback Submission	Report / Feedback Form	Allows students to submit feedback or report issues.

33	Follow Feature	Follow Action	Allows students to follow selected courses.
34	Followed Courses	My Followed Courses	Displays the list of courses followed by the student.
35	Approval Dashboard	Dashboard (Approver)	Displays overview of syllabi awaiting approval.
36	Approval List	Syllabus List	Displays list of syllabi pending review.
37	Approval Review	Syllabus Detail	Allows approvers to review syllabus details.
38	Version Comparison	Compare Versions	Allows approvers to compare different syllabus versions.
39	Outcome Review	View CLO–PLO Mapping	Allows approvers to review CLO–PLO alignment.
40	Approval Decision	Decision	Allows approvers to make approval decisions.
41	Approval Action	Approve	Allows approvers to approve and forward syllabus to next level.
42	Rejection Action	Reject	Allows approvers to reject syllabus submissions.
43	Commenting	Add Comment	Allows approvers to add comments for lecturers.
44	Return Workflow	Return to Lecturer	Returns rejected syllabus back to lecturer for revision.

3.1.3 Screen Authorization

#	Screen	Admin	Lecturer	Student	Approver
1	Login Page	X	X	X	X
2	Register			X	

3	Reset Password			X	
4	Admin Page	X			
5	Manage Dashboard	X			
6	Manage User (User List)	X			
7	Create User	X			
8	Import User	X			
9	Edit / Status	X			
10	System Configuration	X			
11	Manage PLO / CLO Templates	X			
12	Manage Programs	X			
13	Publishing Management	X			
14	Pending Publish List	X			
15	Syllabus Detail (Admin)	X			
16	Dashboard (Lecturer)		X		
17	My Course		X		
18	Course Detail		X		

19	Syllabus List (Lecturer)		X		
20	Create Syllabus		X		
21	Edit Syllabus		X		
22	Version		X		
23	CLO – PLO Mapping		X		
24	Preview Syllabus		X		
25	Submit Syllabus		X		
26	Review Feedback		X		
27	Notification List		X		
28	Home (Student)			X	
29	Search Syllabus			X	
30	Syllabus Detail (Student)			X	
31	AI Summary Tab			X	
32	Learning Path View			X	
33	CLO – PLO Matrix			X	
34	Report / Feedback Form			X	

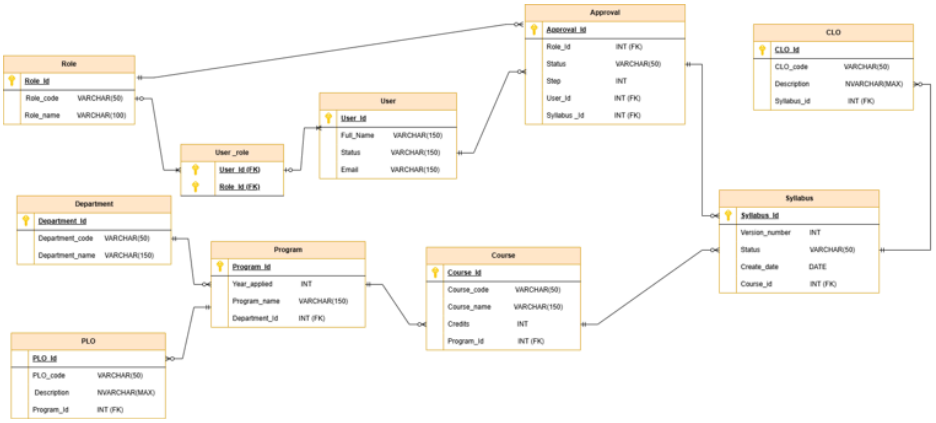
35	Follow Action			X	
36	My Followed Courses			X	
37	Dashboard (Approver)				X
38	Syllabus List (Approver)				X
39	Syllabus Detail (Approver)				X
40	Compare Versions				X
41	View CLO – PLO Mapping				X
42	Decision				X
43	Approve				X
44	Reject				X
45	Add Comment				X
46	Return to Lecturer				X

3.1.4 Non-Screen Functions

#	Feature	System Function	Description
1	Automated Syllabus Approval	Cron Job	Scheduled job that checks and updates the final approval status of syllabi when all required criteria are met.

2	Automated Review Creation	Cron Job	Scheduled job that automatically creates syllabus review schedules when a new semester starts.
3	Automated Syllabus Status Update	Cron Job	Scheduled job that updates syllabus status according to academic timelines (Draft, Review, Approved, Published).
4	Export Review Schedule Template	API Endpoint	Allows Academic Affairs or Admin to export an Excel template for importing syllabus review schedules.
5	Automated CLO–PLO Validation	AI Background Job	Automatically checks the alignment between CLOs and PLOs to support reviewers.
6	Automated Syllabus Summary	AI Background Job	Automatically generates syllabus summaries for student viewing.

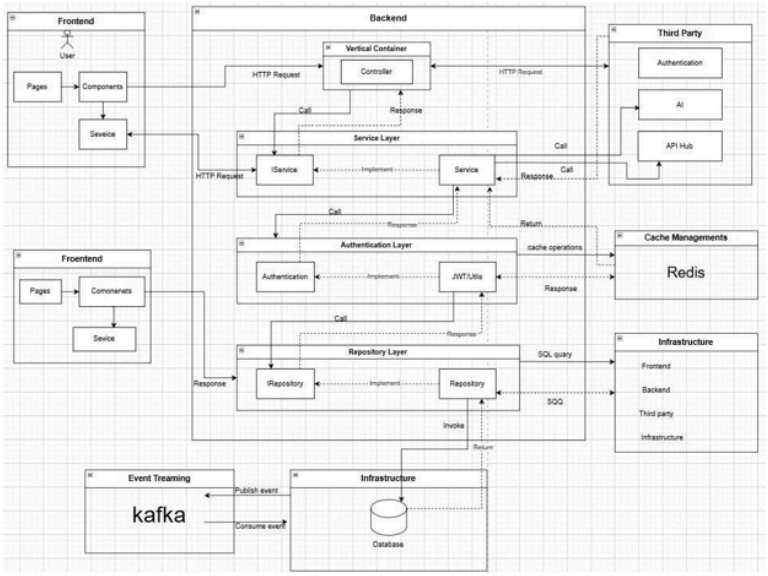
3.1.5 Entity Relationship Diagram



IV. Software Design Description

1. System Design

1.1 System Architecture



1.1.1 Frontend

Google Cloud Platform

The frontend is deployed on Google Cloud Platform to ensure stable performance, scalability, and automated deployment support.

ReactJS / NextJS (TypeScript)

These are the core technologies used to build the Web interface for System Admins, Lecturers, Reviewers/Approvers, and Students. TypeScript improves type safety, reduces errors, and enhances system maintainability.

React Native

Used to develop the mobile application for students, allowing them to look up public syllabuses, view AI-generated summaries, and receive real-time notifications about syllabus updates.

1.1.2 Backend

Python – FastAPI

FastAPI is the main framework for building the Web API, responsible for handling business logic such as user management, syllabus management, approval workflows, and communication with the AI Microservice.

MySQL

A relational database system that stores business data including user information, syllabus content, workflow status, and subject relationships.

Redis

Used as a cache for frequently accessed data and as a message broker for asynchronous AI processing tasks.

Docker

All system components are containerized to ensure fast, consistent deployment and easy scalability.

1.1.3 AI & Crawler Microservice

FastAPI + Celery (Event-driven Architecture)

The AI module is deployed as an independent microservice, handling background tasks such as Semantic Difference detection, CLO–PLO checking, and content summarization.

AI Core & NLP

LangChain is used for orchestration together with PhoBERT, ViBERT, SentenceTransformers, KeyBERT, and VnCoreNLP to perform semantic analysis, keyword extraction, and CLO–PLO logic validation.

GenAI – Llama 3 / OpenAI / Gemini API

These models are used to automatically generate syllabus summaries and support academic content analysis.

Crawler & OCR

Selenium and BeautifulSoup are used to crawl reference materials, while VietOCR and Tesseract are used to extract text from PDF and image files.

PostgreSQL + pgvector, Elasticsearch

Used to store vector data and support full-text and semantic search on syllabuses.

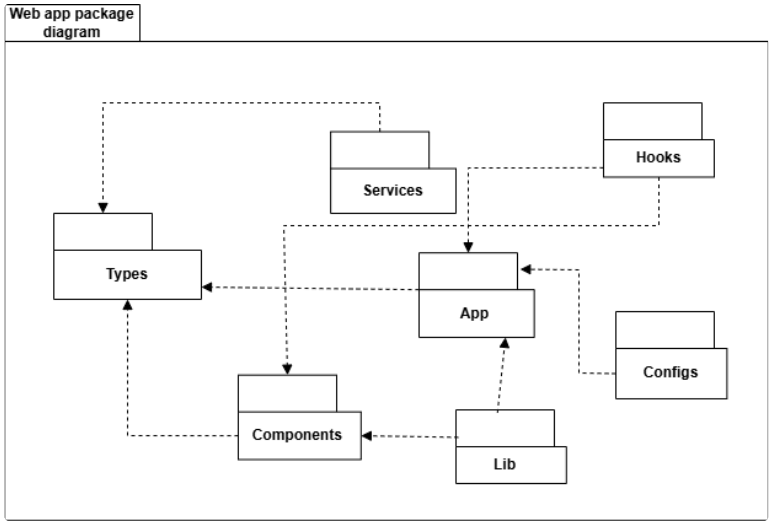
1.1.4 Third-party

Postman

An API testing tool used to validate backend services during system development and integration.

1.2 Package Diagram

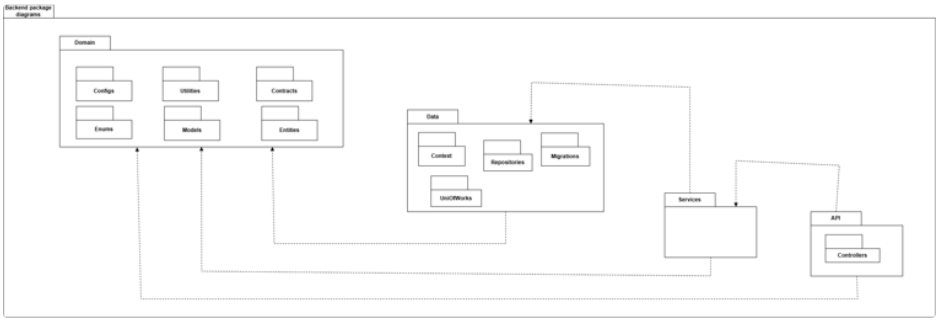
1.2.1 Frontend



No	Package	Description
1	`src`	The root directory contains the entire source code of the application. This is where modules, UI components, processing logic, and project configuration are located.
2	`public`	Contains static resources such as images, icons (favicons), or files that need to be served directly from the server without going through the build process (e.g., HTML, JSON).
3	`src/app`	Contains the core components or main structure of the application. This is where main modules such as pages, features, or important contexts are managed.
4	`src/context`	Contains the core components or main structure of the application. This is where main modules such as pages, features, or important contexts are managed.
5	`src/component`	Contains common UI components such as buttons, modals, or input forms. Often combined with TailwindCSS for consistent styling.
6	`src/assets`	Store static resources related to the interface such as images, fonts, or files that do not change frequently.

7	`src/hooks`	Includes custom React Hooks for reusing complex logic, such as API management, dynamic state, or synchronous actions.
8	`src/config`	Contains the main configuration of the application, including environment variables definition file, API endpoint configuration, or theme configuration with TailwindCSS.
9	`src/lib`	Contains libraries or generic logic source code, for example utility functions or logic processing classes that are not dependent on any specific component.
10	`src/layouts`	Defines the general layout for pages (such as pages with navigation bars, sidebars, or footers). Helps ensure UI consistency across the application.
11	`src/routes`	Manage the application's routing system. Contains files that define URL paths as well as the components that will render for each path.

1.2.2 Backend

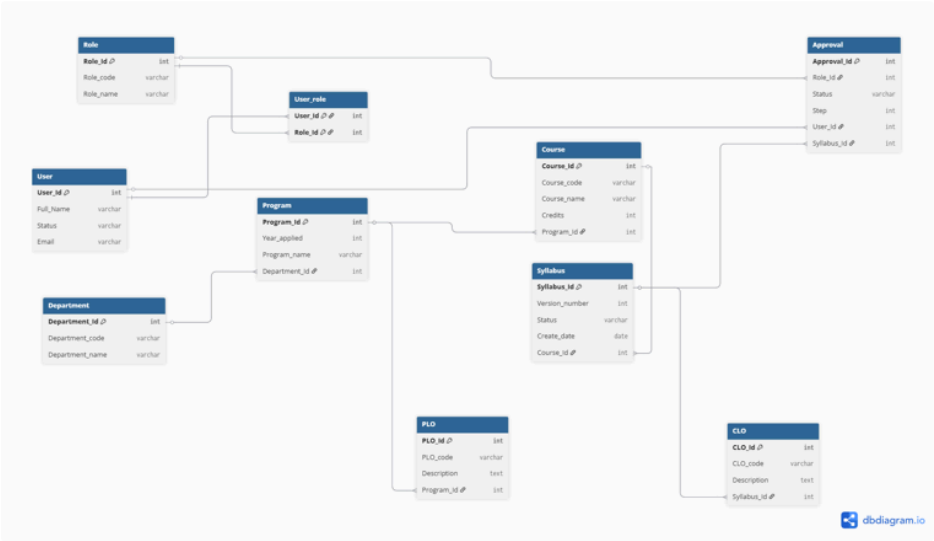


No	Package	Description
1	`src/configuration`	Contains configuration files required for the SMD system, including environment settings (development/production), database configuration, JWT settings, CORS configuration, and global system parameters.
2	`src/controllers`	Handles HTTP requests and responses of the SMD system. This layer receives requests

		from the frontend (authentication, syllabus management, approval workflows), invokes corresponding services, and returns API responses to clients.
3	`src/security`	Manages system security concerns such as user authentication, role-based authorization (Student, Lecturer, Council), JWT token handling, password encryption, and API access control.
4	`src/services`	Contains the core business logic of the SMD system. Responsible for processing operations such as syllabus creation, syllabus revision, submission, approval/rejection, and enforcing business rules.
5	`src/utils`	Provides shared utility functions used across the system, including string processing, input validation, date and time handling, encryption helpers, and syllabus file processing utilities.
6	`src/mappers`	Responsible for transforming data between different layers, such as converting request DTOs to domain entities and mapping domain entities to response DTOs, ensuring separation between API and business models.
7	`src/exceptions`	Defines and handles application-level exceptions, including authentication errors, authorization errors, data-not-found exceptions, and business rule violations related to syllabus processing.
8	`src/repositories`	Acts as the Data Access Layer (DAL). Handles CRUD operations and database interactions for entities such as User, Course, Syllabus, Approval, and Version.
9	`src/specification`	Defines query specifications and filtering criteria for complex data retrieval, supporting syllabus search and filtering by status, course, lecturer, or academic term.

10	`src/models`	Contains domain models and entities representing the core business objects of the SMD system, including User, Course, Syllabus, Approval, Version, and their relationships.
11	`src/resources`	Stores shared application resources such as templates, configuration assets, or common resources required during system execution and deployment.

2. Database Design



2.1 User Table

Field Name	Type	Description	Unique	Not null	PK/FK
User_Id	int	Unique identifier for user	yes	yes	PK
Full_Name	varchar(255)	Full name of the user	no	yes	
Email	varchar(255)	User email address	yes	yes	
Status	varchar(50)	Account status (active/inactive)	no	no	

2.2 Role Table

Field Name	Type	Description	Unique	Not null	PK/FK
Role_Id	int	Unique identifier for role	yes	yes	PK
Role_code	varchar(100)	Role code (ADMIN, LECTURER, STUDENT ...)	yes	yes	
Role_name	varchar(255)	Role name	no	yes	

2.3 UserRole Table

Field Name	Type	Description	Unique	Not null	PK/FK
User_Id	int	Reference to User	no	yes	FK
Role_Id	int	Reference to Role	no	yes	FK

2.4 Program Table

Field Name	Type	Description	Unique	Not null	PK/FK
Program_Id	int	Unique identifier for program	yes	yes	PK
Program_name	varchar(255)	Program name	no	yes	
Year_applied	int	Year the program is applied	no	yes	

Department_Id	int	Reference to Department	no	yes	FK

2.5 Course Table

Field Name	Type	Description	Unique	Not null	PK/FK
Course_Id	int	Unique identifier for course	yes	yes	PK
Course_code	varchar(100)	Course code	yes	yes	
Course_name	varchar(255)	Course name	no	yes	
Credits	int	Number of credits	no	yes	
Program_Id	int	Reference to Program	no	yes	FK

2.6 Syllabus Table

Field Name	Type	Description	Unique	Not null	PK/FK
Syllabus_Id	int	Unique identifier for syllabus	yes	yes	PK
Version_number	int	Version number of syllabus	no	yes	
Status	varchar(50)	Syllabus status	no	yes	
Create_date	date	Creation date	no	yes	

Course_Id	int	Reference to Course	no	yes	FK

2.7 CLO Table

Field Name	Type	Description	Unique	Not null	PK/FK
CLO_Id	int	Unique identifier for CLO	yes	yes	PK
CLO_code	varchar(100)	CLO code	yes	yes	
Description	text	CLO description	no	no	
Syllabus_Id	int	Reference to Syllabus	no	yes	FK

2.8 PLO Table

Field Name	Type	Description	Unique	Not null	PK/FK
PLO_Id	int	Unique identifier for PLO	yes	yes	PK
PLO_code	varchar(100)	PLO code	yes	yes	
Description	text	PLO description	no	no	
Program_Id	int	Reference to Program	no	yes	FK

2.9 Department Table

Field Name	Type	Description	Unique	Not null	PK/FK

Department_Id	int	Unique identifier for department	yes	yes	PK
Department_code	varchar(100)	Department code	yes	yes	
Department_name	varchar(255)	Department name	no	yes	

2.10 Approval Table

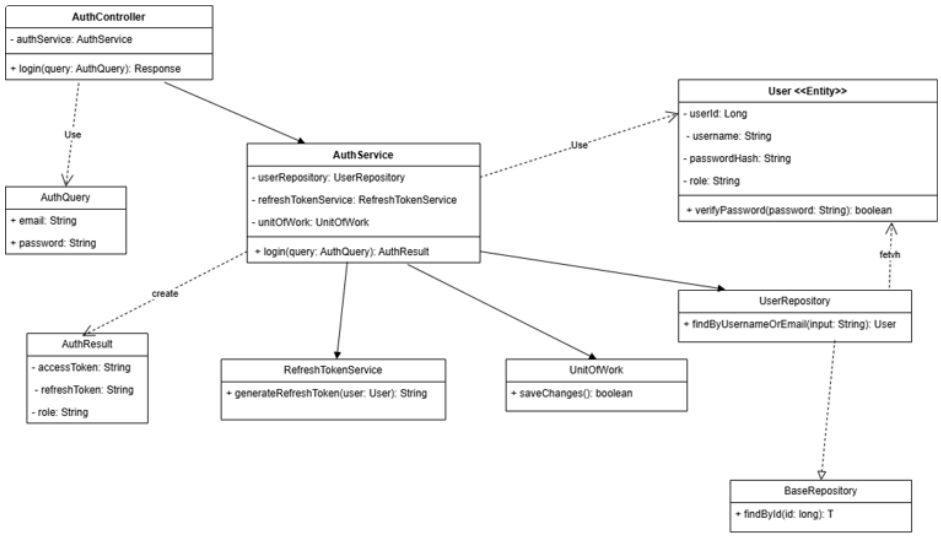
Field Name	Type	Description	Unique	Not null	PK/FK
Approval_Id	int	Unique identifier for approval	yes	yes	PK
Role_Id	int	Role responsible for approval	no	yes	FK
User_Id	int	User who performs approval	no	yes	FK
Syllabus_Id	int	Reference to syllabus	no	yes	FK
Step	int	Approval step number	no	yes	
Status	varchar(50)	Approval status (pending/approved/rejected)	no	yes	

3. Detailed Design

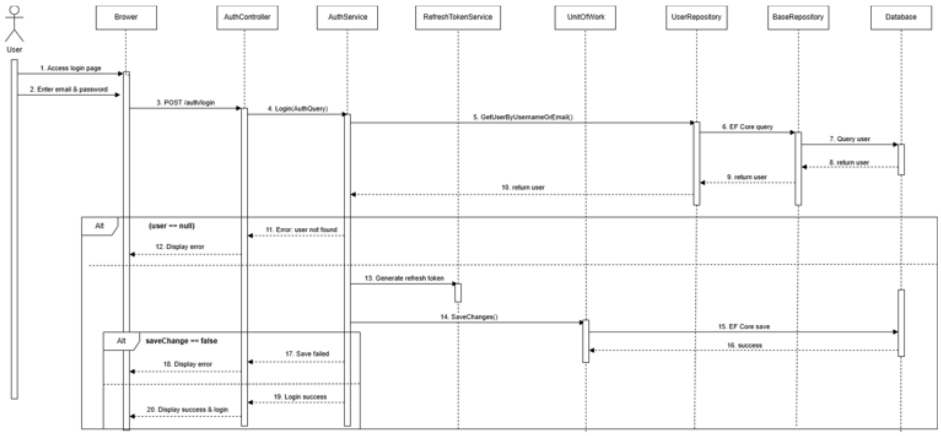
3.1 Authentication

3.1.1 Login

3.1.1.1 Class Diagram



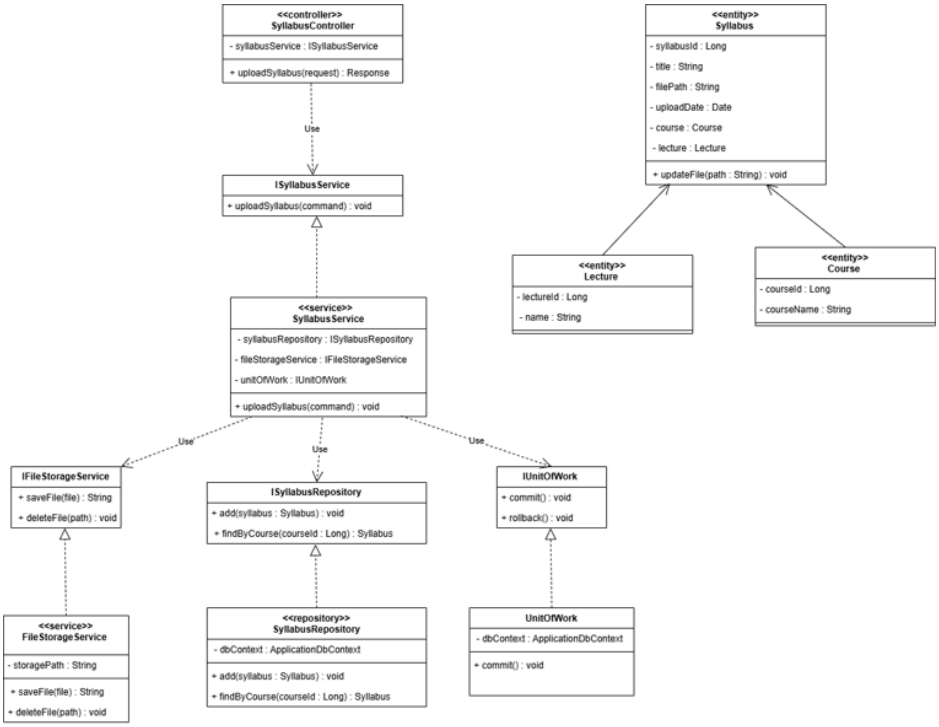
3.1.1.2 Sequence Diagram



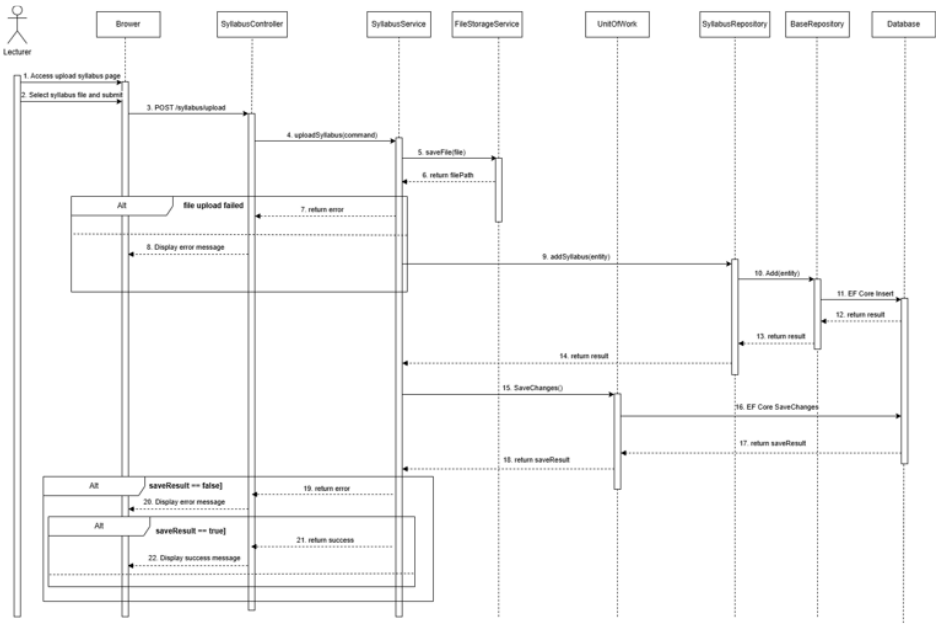
3.2 Syllabus Management Module

3.2.1 Lecture upload syllabus

3.2.1.1 Class Diagram

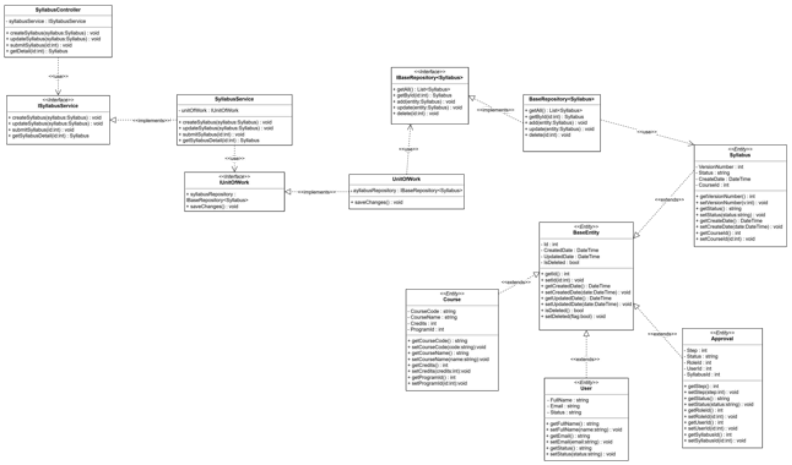


3.2.1.2 Sequence Diagram

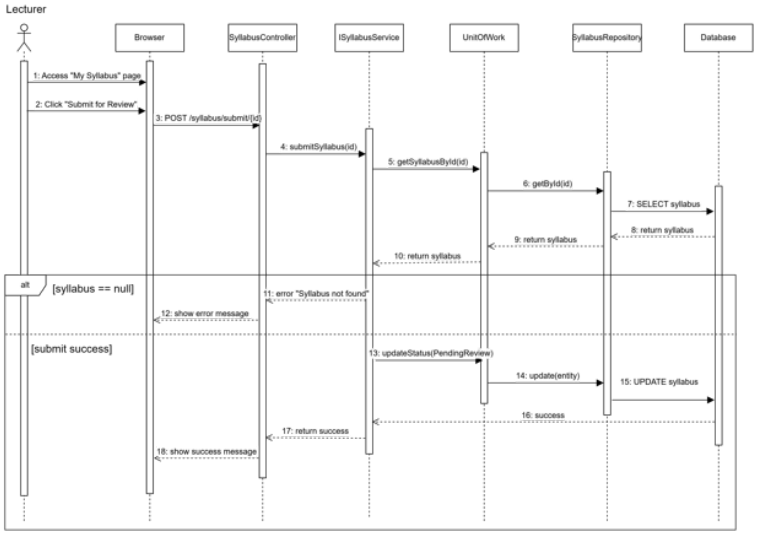


3.2.2 Lecture submit syllabus

3.2.2.1 Class Diagram

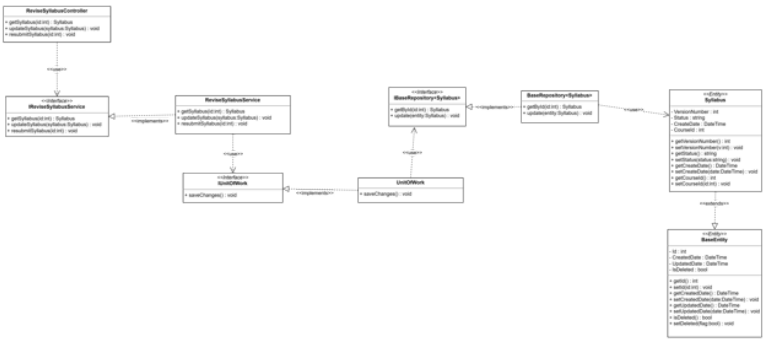


3.2.2.2 Sequence Diagram

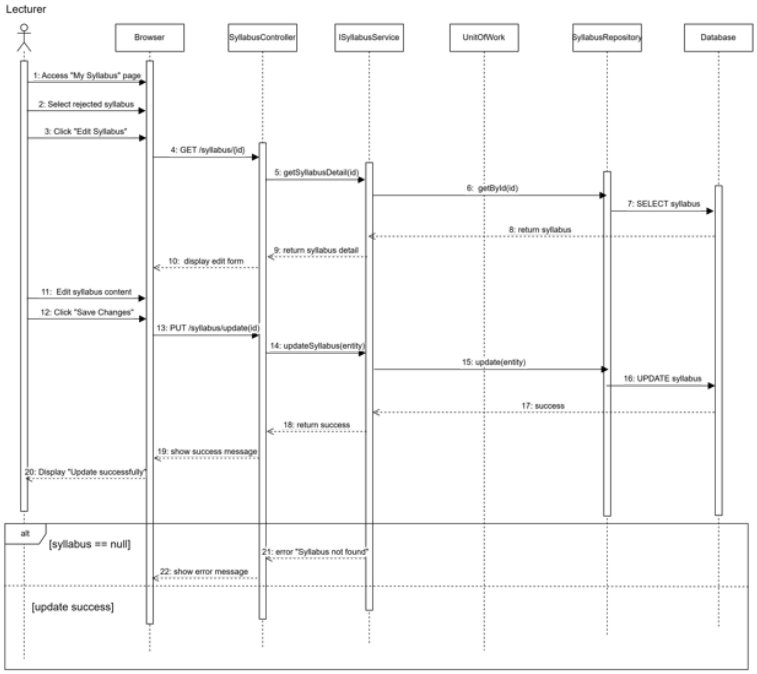


3.2.3 Lecturer revise syllabus

3.2.3.1 Class Diagram



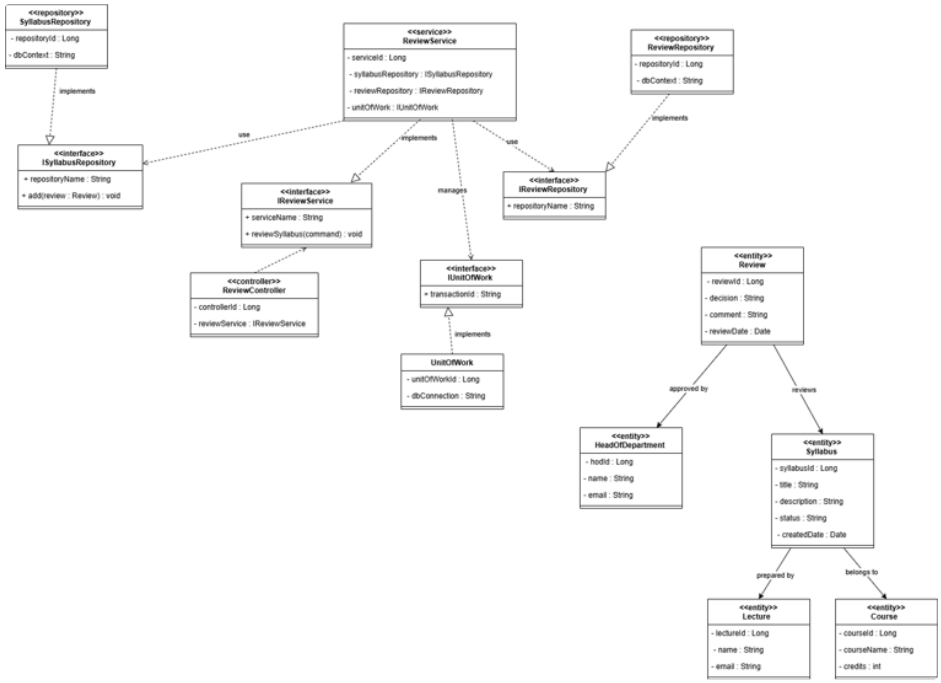
3.2.3.2 Sequence Diagram



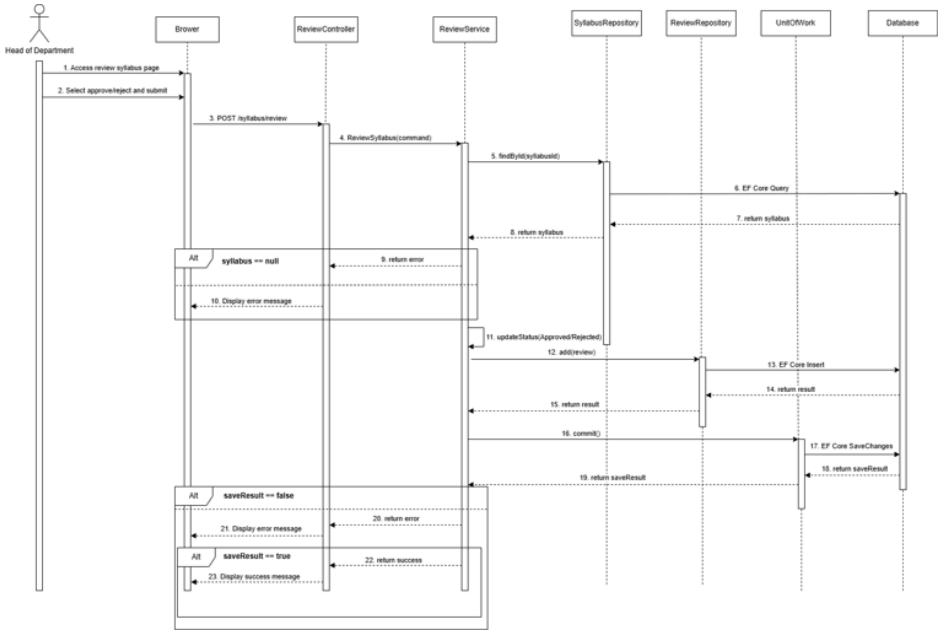
3.3 Review & Approval Workflow

3.3.1 Head of Department Review

3.3.1.1 Class Diagram

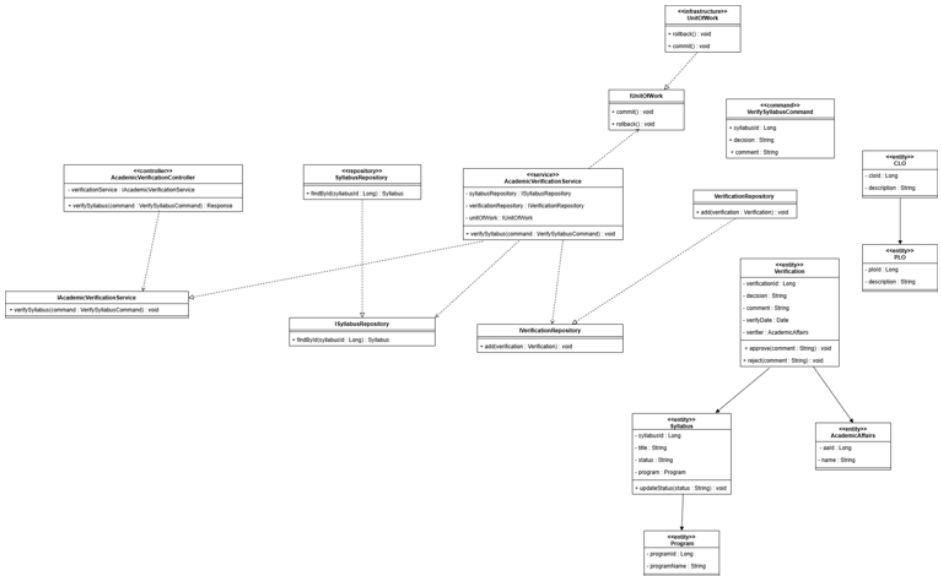


3.3.1.2 Sequence Diagram

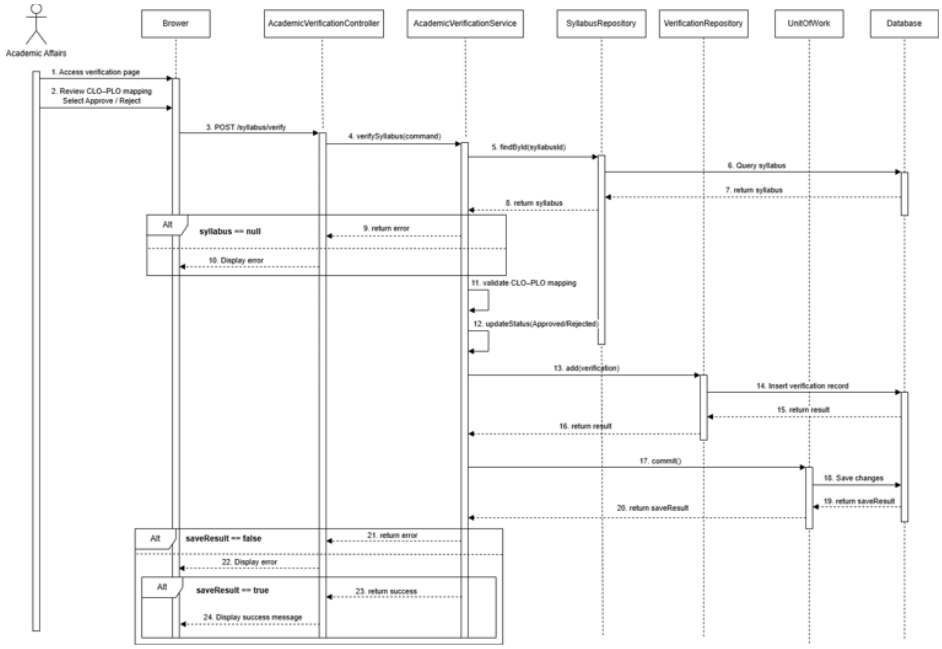


3.3.2 Academic Affairs Administration (AAA) Verification

3.3.2.1 Class Diagram

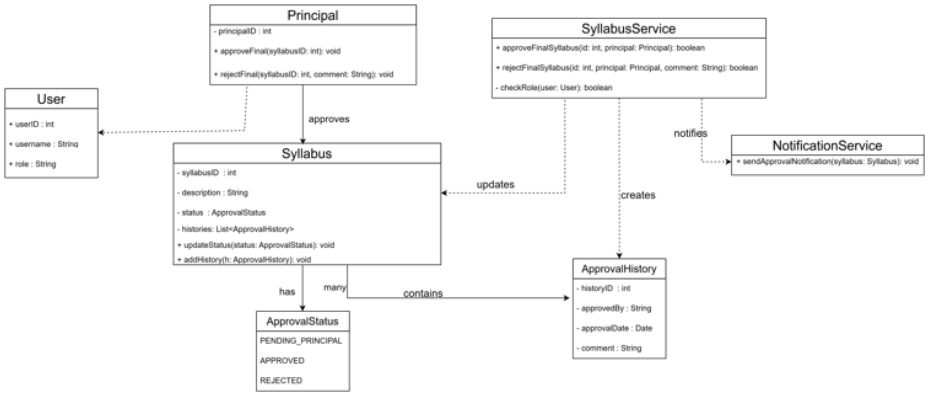


3.3.2.2 Sequence Diagram

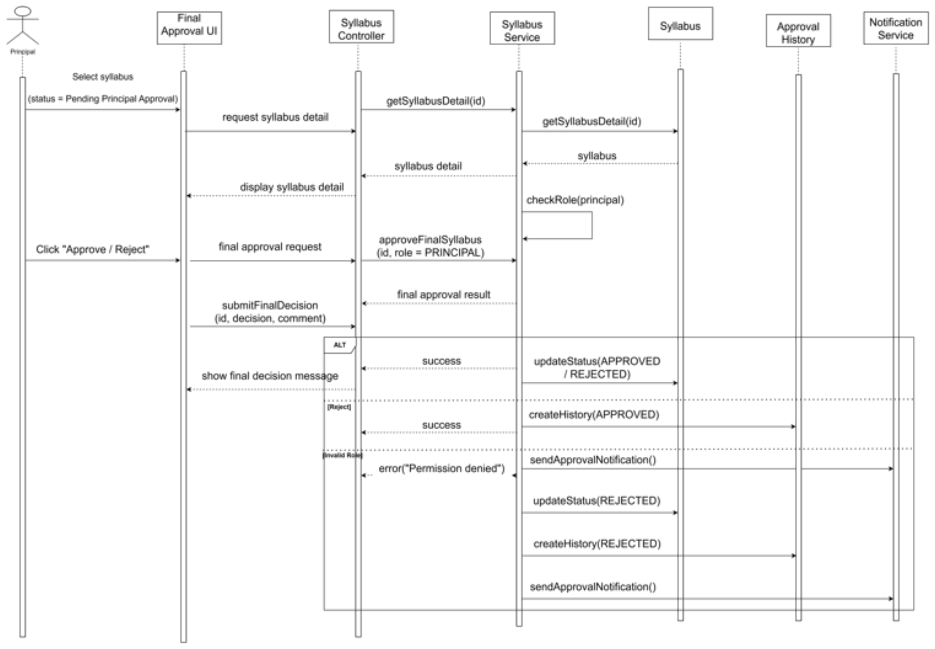


3.3.3 Principal Final Approval

3.3.3.1 Class Diagram



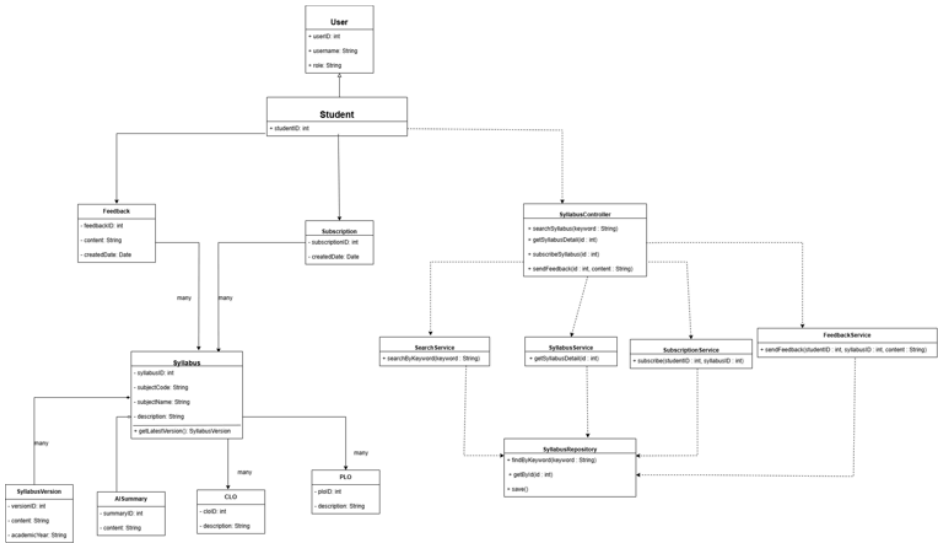
3.3.3.2 Sequence Diagram



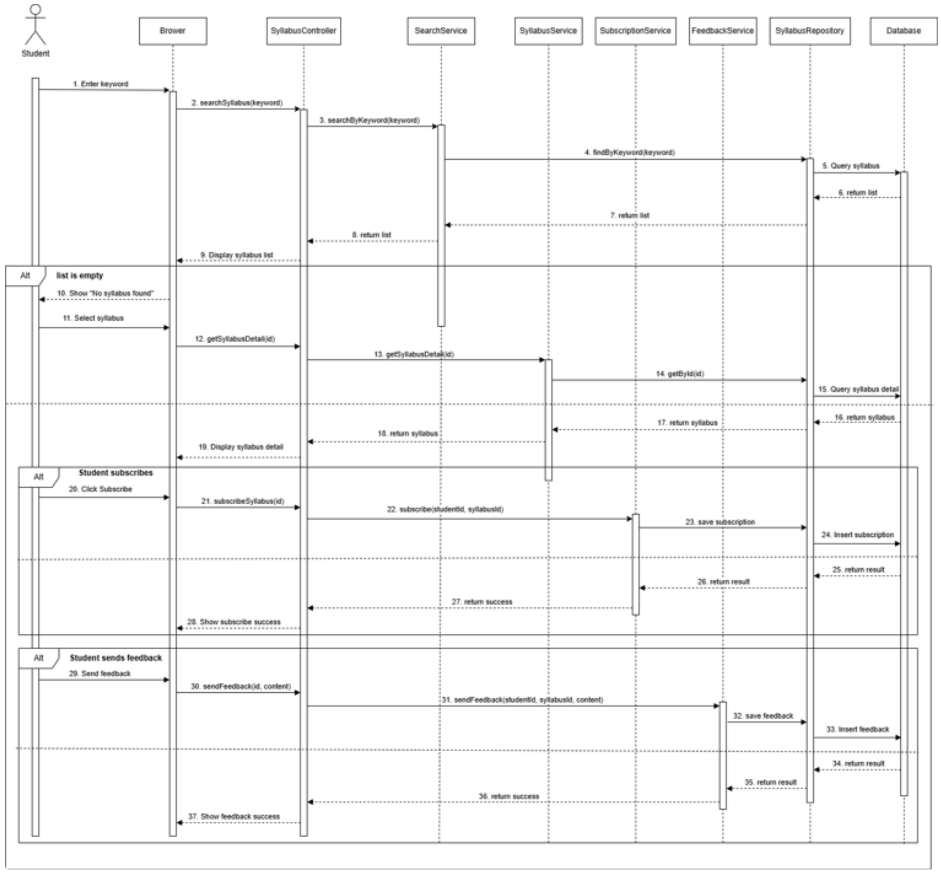
3.4 Syllabus Search & View

3.4.1 Student view syllabus

3.4.1.1 Class Diagram



3.4.1.2 Sequence Diagram


Student
Brower
SyllabusController
SearchService
SyllabusService
SubscriptionService
FeedbackService
SyllabusRepository
Database
1. Enter keyword
2. searchSyllabus(keyword)
3. searchByKeyword(keyword)
4. findByKeyword(keyword)
5. Query syllabus
6. return list
7. return list
8. return list
9. Display syllabus list
Alt
list is empty
10. Show "No syllabus found"
11. Select syllabus
12. getSyllabusDetail(id)
13. getSyllabusDetail(id)
14. getById(id)
15. Query syllabus detail
16. return syllabus
17. return syllabus
18. return syllabus
19. Display syllabus detail
Alt
Student subscribes
20. Click Subscribe
21. subscribeSyllabus(id)
22. subscribe(studentId, syllabusId)
23. save subscription
24. Insert subscription
25. return result
26. return result
27. return success
28. Show subscribe success
Alt
Student sends feedback
29. Send feedback
30. sendFeedback(id, content)
31. sendFeedback(studentId, syllabusId, content)
32. save feedback
33. Insert feedback
34. return result
35. return result
36. return success
37. Show feedback success