

Software Design Specifications Document

TALIM - An AI Based Virtual Educational Assistant

By

Abdullah Tahir - 393056

Imaan Ibrar - 373982

Laiba Atiq - 372187

Supervisor:

Dr. Seemab Latif

Co-Advisor:

Dr. Aimal Rextin

Bachelor of Engineering in Software Engineering

Department of Computing

School of Electrical Engineering and Computer Science

National University of Sciences & Technology

1 Introduction

1.1 Purpose

This **System Design Specification (SDS)** document outlines the architectural and design blueprint of **TALIM**, an AI-based virtual educational assistant developed as part of the Final Year Project titled “*TALIM: An AI-Based Virtual Educational Assistant*”. This document corresponds to the second release of the product, version 2.0.

TALIM is designed to assist educators and students by automating key educational processes, including generating quizzes, assignments, and their corresponding assessment rubrics, as well as providing interactive, AI-driven responses to students’ queries related to course material. The system processes inputs such as lecture notes

and presentations in PDF format and outputs context-aware content like quizzes, assignments, and answers to student questions.

This SDS defines the **system architecture**, **design methodology**, **data representation**, and **functional design models** required to implement TALIM. It serves as a comprehensive reference for the development team to ensure that TALIM's design aligns with the specified requirements in the SRS, enabling seamless development and deployment of the system's core functionalities.

1.2 Document Conventions

1. Default Font
 - Font Style: Computer Modern
 - Font Size: 10pt (for article)
2. Section Heading
 - Font Size: 12pt (when the base font size is 10pt)
 - Format: Bold
3. Subsection Heading
 - Font Size: 11pt (when the base font size is 10pt)
 - Format: Bold
4. Subsubsection Heading
 - Font Size: 10pt (when the base font size is 10pt)
 - Format: Bold and italic

1.3 Scope

TALIM - AI-based Virtual Educational Assistant will transform learning by integrating advanced technology into educational practices. It will automatically process lecture notes and presentations, creating a comprehensive knowledge base that students can ask questions on and receive instant clarifications. The software would further automate the generation of quizzes and assignments, streamlining the process of marking for educators. Making such an investment will incorporate objectives of innovation and access to education, enhancing the strategies for increased engagement of the students, better learning results, and reducing administrative burdens on teachers. The project intends to lead the way in the educational technology space by moving forward the mission of quality education, making it more effective and accessible for all learners.

2. Design Methodology and Software Process Model

2.1 Modular Design Methodology

The design and development of **TALIM: An AI-Based Virtual Educational Assistant** follows a **Modular Design Methodology**, dividing the system into manageable modules with clearly defined purposes and interactions. This ensures scalability, maintainability, and ease of integration for TALIM's complex educational functionalities.

Key Modules and Their Roles:

1. User Interface Layer

- a. Provides dashboards for both students and teachers.
- b. Offers functionalities such as course enrollment, content access, and Virtual Teaching Assistant (VTA) chat.
- c. Teachers can create, manage, and access course materials, while students engage with learning content and receive AI-powered query support.

2. Authentication and User Management Module

- a. Manages user roles (students and teachers), handles registration and login processes, and ensures access control.
- b. Implements role-based permissions and ensures secure course enrollment and session management.

3. Query Analyzer Module

- a. Validates and categorizes queries to determine whether they are within the system's scope.
- b. Filters out unsupported queries, reducing computational overhead on the RAG layer by identifying those requiring visual or textual context.

4. RAG (Retrieval Augmented Generation) Module

- a. Retrieves relevant course material stored as embeddings in the vector database using MultiQuery retrievers.
- b. Combines retrieved content with prompt templates engineered for specific query intents like summaries, examples, and explanations.
- c. Leverages chain-of-thought reasoning and Google's Gemini LLM for generating accurate, contextually appropriate responses.
- d. Maintain a chat history to support interactive sessions.

5. Database Layer

- a. Stores and retrieves semantic chunks of processed content using ChromaDB for vectorized embeddings.
- b. Maintains metadata and ensures persistent storage for course materials and user-generated data.

This modular structure allows parallel development of components and ensures seamless interactions between system subsystems.

2.2 Justification for Modular Design Methodology

The Modular Design Methodology was chosen for TALIM due to its suitability for building AI-driven, scalable, and maintainable systems.

- **Scalability and Parallel Development:**
Each module is developed independently, such as the Query Analyzer and Course Management modules, ensuring scalability and efficient debugging.
- **Simplification of Complexity:**
TALIM's advanced features, including NLP processing, RAG integration, and LLM-powered response generation, are divided into smaller tasks for easier implementation.
- **Reusability:**
Modular components, such as the Query Analyzer and RAG module, can be enhanced or reused for future system updates or similar applications.
- **Maintainability:**
Independent modules ensure that updates or changes in one module do not affect others, making TALIM easy to maintain and improve.

2.3 Key Features of the Incremental Model for TALIM

The Incremental Software Development Model was selected for TALIM to ensure functionality is delivered progressively, enabling early deployment and feedback collection.

- **Phased Development:**
Development occurs in increments, with each delivering specific features:
 - a. **Increment 1:** Core modules, including Input Processing and the VTA Module.
 - b. **Increment 2:** Quiz and Assignment Generation modules with integrated assessment rubrics.
 - c. **Increment 3:** User Interface integration and Output Presentation features.
- **Prioritization of Core Functionalities:**
Early increments focus on critical features, such as content processing and VTA responses, ensuring the system's core is operational from the beginning.
- **Early User Feedback:**
Educators and students provide feedback on early increments, enabling refinements in subsequent phases.

- **Risk Mitigation:**

Incremental development allows iterative validation of AI features like LLM response accuracy, vector retrieval.

Why the Incremental Model Was Chosen:

1. **AI Feature Complexity:**

TALIM's AI components, such as the Query Analyzer and RAG pipeline, require incremental validation to ensure accuracy and performance.

2. **Flexibility:**

The model accommodates changes in design or functionality based on user feedback or evolving requirements.

3. **Timely Deliverables:**

Early delivery of functional increments allows stakeholders to experience TALIM's capabilities and provide actionable insights for improvement.

3. System Overview

3.1 Functionality

TALIM is an AI-based virtual educational assistant designed to automate and streamline various educational processes, enhancing teaching efficiency and facilitating student learning. The system processes lecture notes, presentations, and other course-related material to generate context-aware content, including:

- Automated Quiz Generation: Creates quizzes with multiple question formats (MCQs, short answers) tailored to course content.
- Assignment and Assessment Rubric Generation: Generates assignments and corresponding rubrics for evaluation.
- AI-Powered Student Support: Provides real-time responses to students' questions related to course material using NLP techniques.
- Interactive Learning Environment: Bridges the gap between educators and students by offering personalized assistance and reducing manual workload.

3.2 Context

With the growing demand for intelligent solutions in education, TALIM addresses the challenges faced by both educators and students:

- Educators: Reduces time spent on manual content creation (quizzes, assignments) and grading. Provides tools to focus more on interactive teaching.
- Students: Offers on-demand access to learning resources and answers to their academic questions, fostering self-learning and improving understanding of concepts.

TALIM processes inputs such as PDF lecture notes, slides, and course outlines to extract relevant information using Natural Language Processing (NLP). Outputs are generated in the form of:

- Structured quizzes and assignments.
- Assessment rubrics for grading assignments.
- AI-generated answers to student queries, enhancing their engagement with the course material.

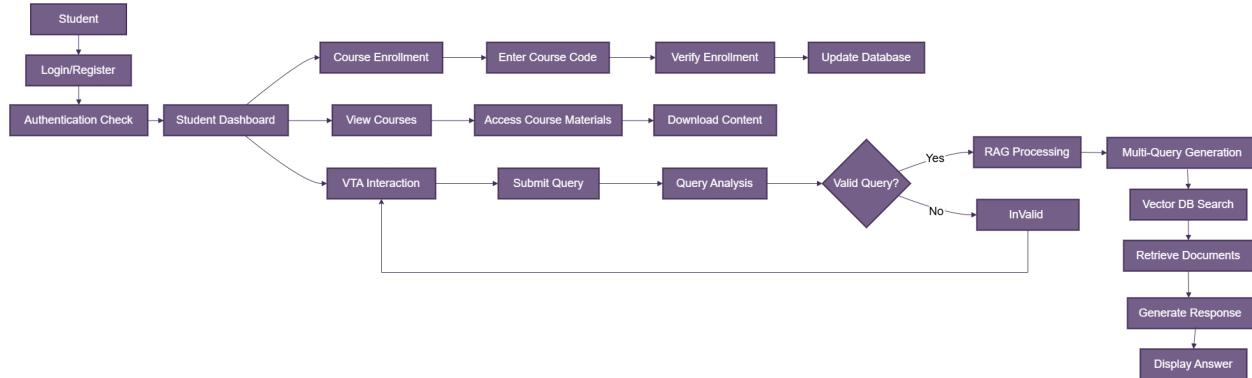
3.3 Design

The TALIM system employs a modular architecture built primarily with Python and the LangChain framework, leveraging Google's Gemini 1.5 Flash LLM for intelligent processing. The system starts with a document processing pipeline that uses PyPDFium2Loader to parse PDF lecture materials with OCR capabilities, followed by semantic chunking to create context-aware text segments. These chunks are then converted into vector embeddings using Google's text-embedding-004 model and stored in ChromaDB collections organized by subject matter.

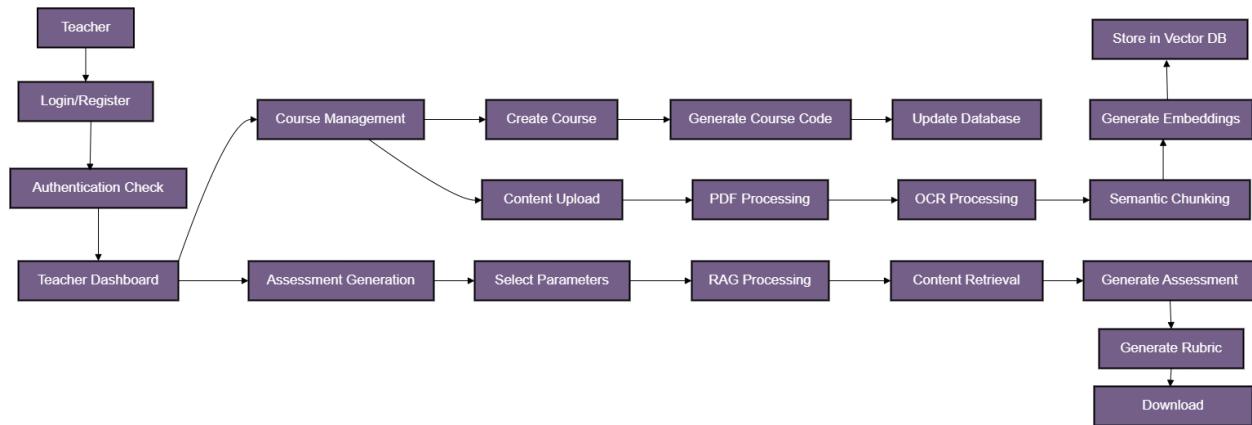
The system's core functionality is implemented through a sophisticated RAG (Retrieval Augmented Generation) approach, where user queries first pass through a query analyzer that evaluates if the query requires visual context or is within the system's scope. Valid queries are then processed through a MultiQuery Retriever to fetch relevant content from the vector store. The retrieved content is combined with carefully engineered prompt templates that utilize one-shot learning and chain-of-thought reasoning to generate contextually appropriate responses. The user interface follows a modern design with separate dashboards for students and teachers, featuring course windows, enrollment management, virtual educational assistant chat, content management, and assessment generation tools as specified in the UI design specifications.

3.4 Workflow Diagram

3.4.1 Student Workflow



3.4.2 Teacher Workflow



4. Architectural Design

The architectural design of TALIM follows a modular and systematic approach, ensuring that all subsystems work cohesively to deliver the desired functionalities. This modularity simplifies the development, maintenance, and scalability of TALIM's features. Each subsystem is designed to handle specific tasks while maintaining clear communication pathways with other subsystems. This approach supports the seamless integration of complex AI-driven features such as Retrieval-Augmented Generation (RAG) and Natural Language Processing (NLP) for educational content generation.

The architecture emphasizes robust user interaction, secure data management, and efficient retrieval and generation processes, making TALIM a comprehensive virtual educational assistant. Below is a detailed description of TALIM's modular program structure, subsystem relationships, and high-level interaction flow.

4.1 Modular Program Structure

1. User Interface Layer

The interface layer serves as the primary point of interaction for both students and teachers. The layer provides comprehensive interfaces including student and teacher dashboards that enable course access, enrollment features, and VTA chat functionality. Teachers can access specialized interfaces for course creation, content management, and assessment generation, while students interact with course windows for content viewing and the Virtual Educational Assistant for queries.

2. Authentication and User Management Module

The authentication module handles all aspects of user identity and access control within the system. It manages the complete user lifecycle from registration through login processes, while maintaining active session management. The module implements role-based access control to differentiate between student and teacher privileges, handles course enrollment verification, and provides user profile management capabilities.

3. Query Analyzer Layer

The Query Analyzer acts as the first line of processing for all incoming queries, performing crucial validation to determine if queries are within system scope or require visual context. The aim of this layer is to reduce LLM call and retrieval computations by filtering baseless and out of scope queries.

4. RAG (Retrieval Augmented Generation) Layer

The RAG layer implements a sophisticated content retrieval and generation system. The Document Retriever employs MultiQuery technology to expand the original query into multiple variations, enhancing the search coverage across the vector database. The Response Generator utilizes carefully engineered prompts with one-shot examples and chain-of-thought reasoning, tailored to different query intents like explanations, summaries, or examples. This layer interacts

closely with the Gemini LLM to produce contextually accurate and educationally appropriate responses.

6. Database Layer

The document processing component forms the foundation of content handling, using OCR-capable processing to parse PDF materials and implementing semantic chunking to preserve context. It generates embeddings for all processed content while managing metadata extraction and organization. The vector database management component provides sophisticated storage and retrieval of document embeddings, organizing collections by subject matter, implementing efficient vector similarity search capabilities, and maintaining persistent storage with automated indexing. It also uses another database to store other details taken from the user while signing, adding etc.

4.2 Subsystem Relationships

The modular structure of TALIM ensures seamless interaction between subsystems. The relationships between subsystems are as follows:

Database Layer ↔ RAG Layer

The Database subsystem provides processed and chunked content to the RAG Layer. When course materials are uploaded, they are processed into semantic chunks and embedded, then stored in the vector database. The RAG Layer later retrieves these processed documents for query responses and assessment generation. The relationship ensures that high-quality, context-aware content is available for retrieval and generation tasks.

Query Analyzer Layer ↔ RAG Layer

The Query Analyzer subsystem sends validated and analyzed queries to the RAG Layer. After analyzing query intent and scope, the Query Processor forwards qualified queries to RAG for document retrieval and response generation. The RAG Layer then uses the query intent to select appropriate response templates and retrieval strategies. This relationship ensures that queries are properly understood and answered with relevant context.

RAG Layer ↔ User Interface Layer

The RAG Layer sends generated responses and content to the Output Presentation subsystem for formatting and display. Generated responses, whether from student queries or assessment creation, are formatted appropriately for the user interface. This relationship ensures that AI-generated content is presented in a user-friendly and accessible manner.

User Interface Layer ↔ Authentication Layer

The User Interface subsystem interacts with Authentication for user access control. All user actions first pass through authentication verification before accessing other subsystems. This relationship maintains system security and proper role-based access control.

Authentication Layer ↔ Database Layer

The Authentication subsystem works with Database to enforce course access permissions. It verifies whether students have proper enrollment and teachers have appropriate course management rights. This relationship ensures that course content and features are only accessible to authorized users.

Database Layer ↔ Assessment Generation Layer

The Database subsystem provides course content to Assessment Generation for creating educational materials. When teachers request assessments, the course content is used as context for generating relevant questions and rubrics. This relationship ensures that generated assessments align with course materials.

4.3 High-Level Flow of Subsystem Interaction

Student Query Flow

1. User Interface Layer → Authentication Layer (verifies user access)
2. Authentication Layer → Query Analyzer Layer
3. Query Analyzer Layer → RAG Layer (if query is valid)
4. RAG Layer → Database Layer (retrieves relevant content)
5. RAG Layer → User Interface Layer (displays response)

Course Material Upload Flow

1. User Interface Layer → Authentication Layer
2. Authentication Layer → Database Layer
3. Database Layer → User Interface Layer (confirms upload)

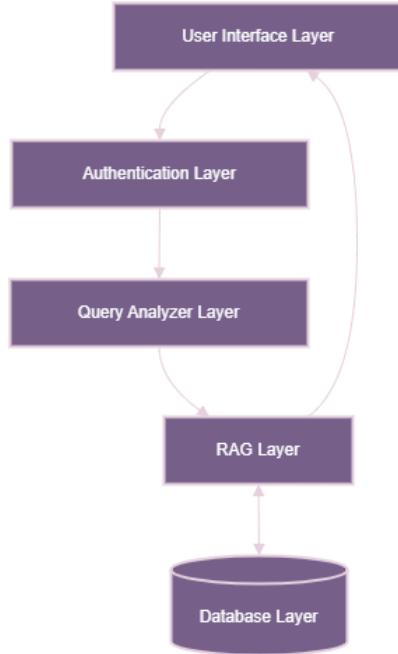
Assessment Generation Flow

1. User Interface Layer → Authentication Layer
2. Authentication Layer → Database Layer
3. Database Layer → RAG Layer
4. RAG Layer → Database Layer (retrieves course content)
5. RAG Layer → Database Layer
6. Database Layer → User Interface Layer (displays assessment)

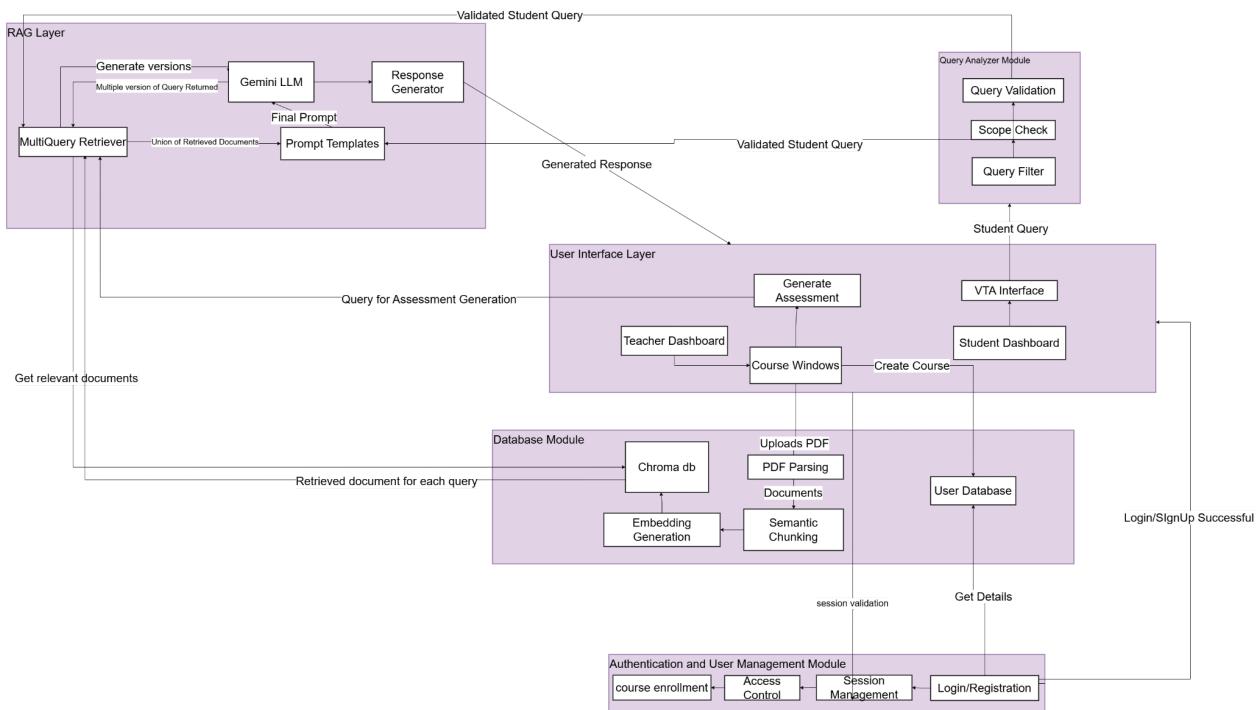
Course Enrollment Flow

1. User Interface Layer → Authentication Layer
2. Authentication Layer → Database Layer
3. Database Layer → User Interface Layer (confirms enrollment)

4.4 Box and Line Diagram



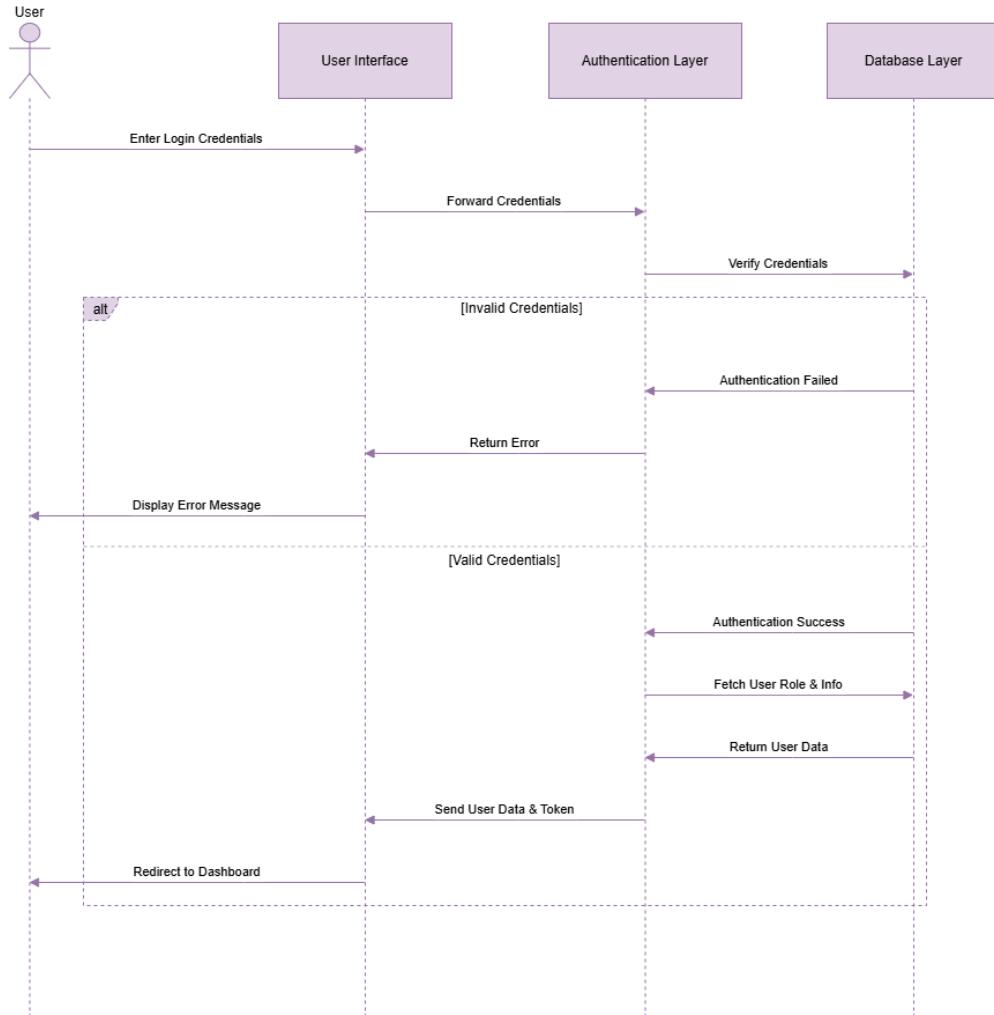
4.5 Architecture Diagram



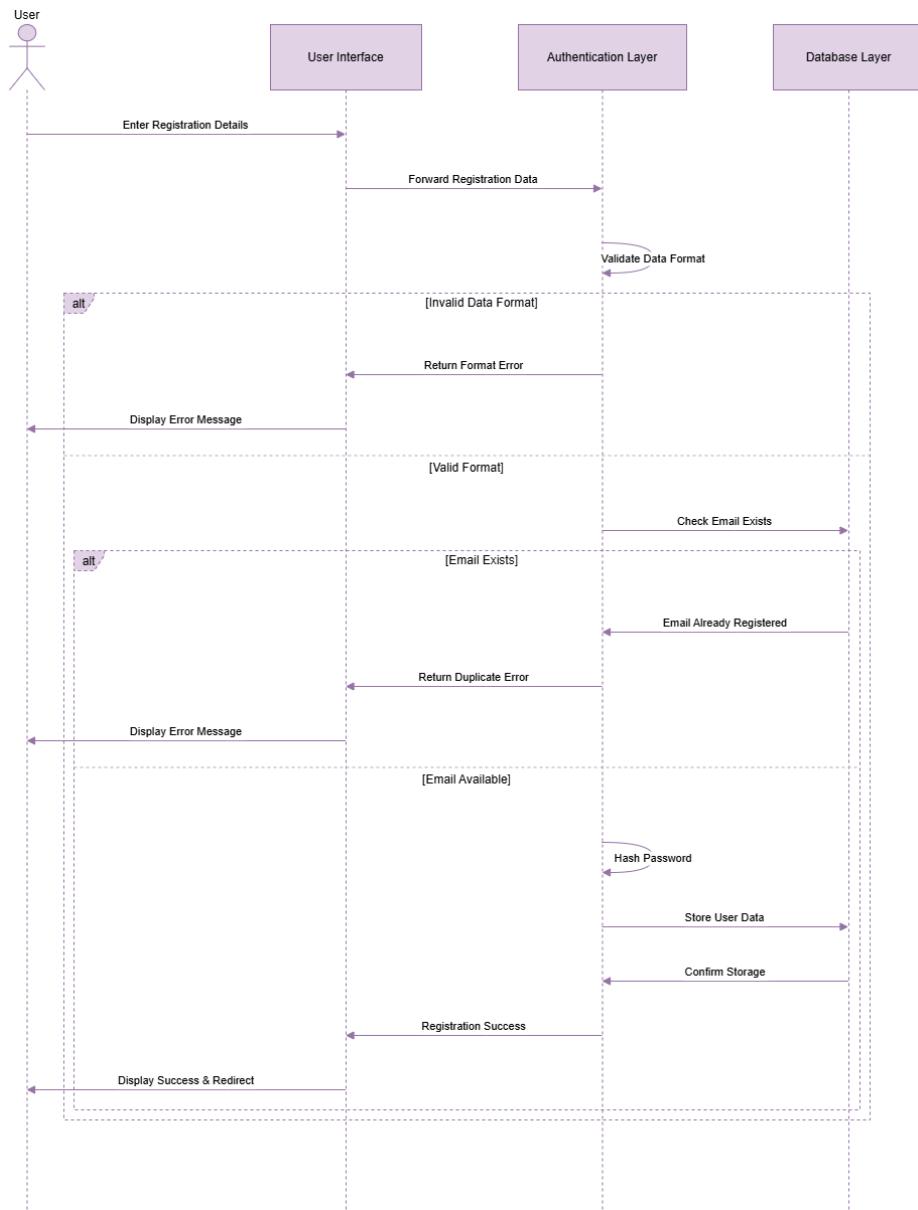
5. Design Models

5.1 Sequence Diagram

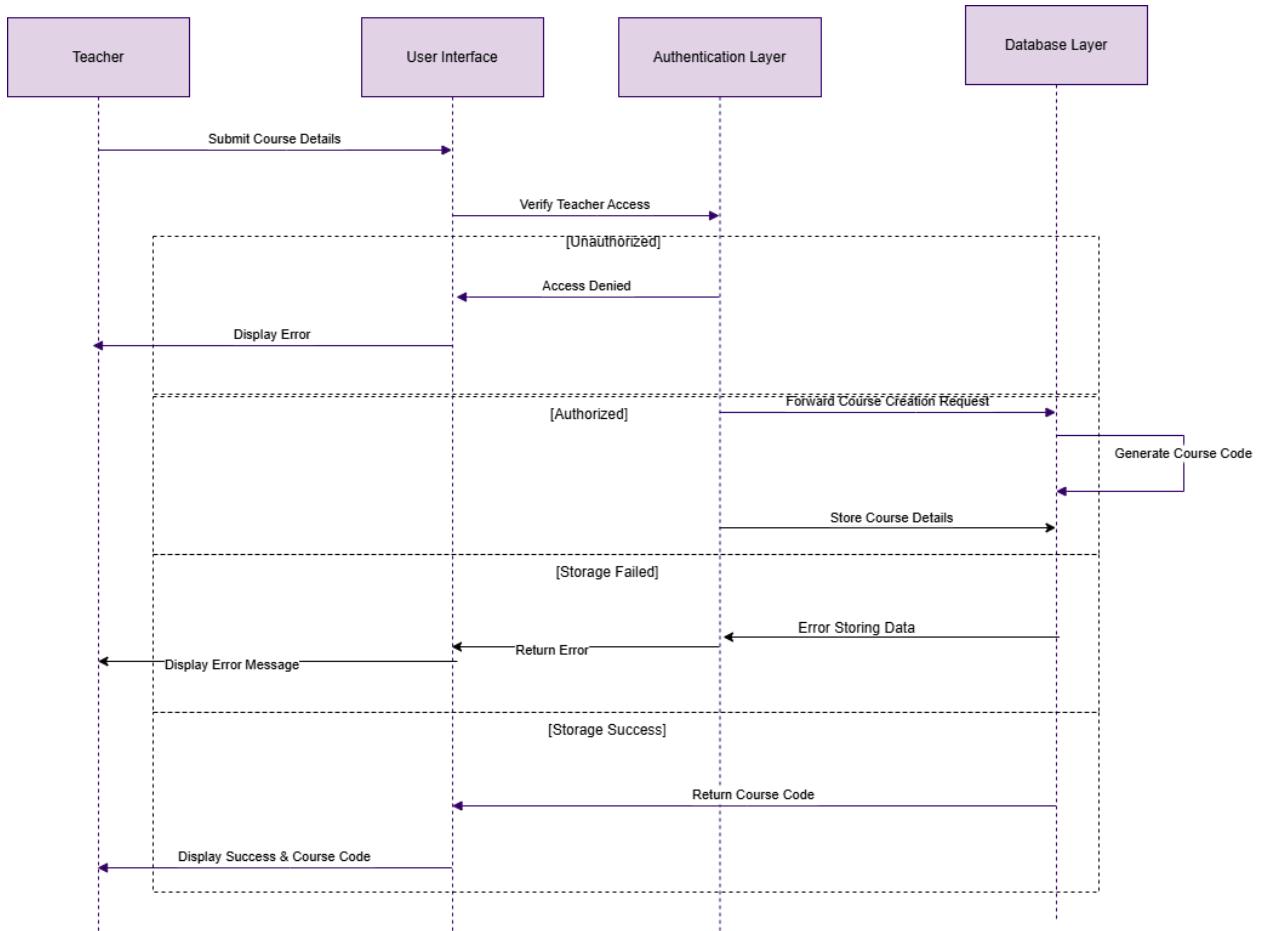
A. Login Sequence Diagram



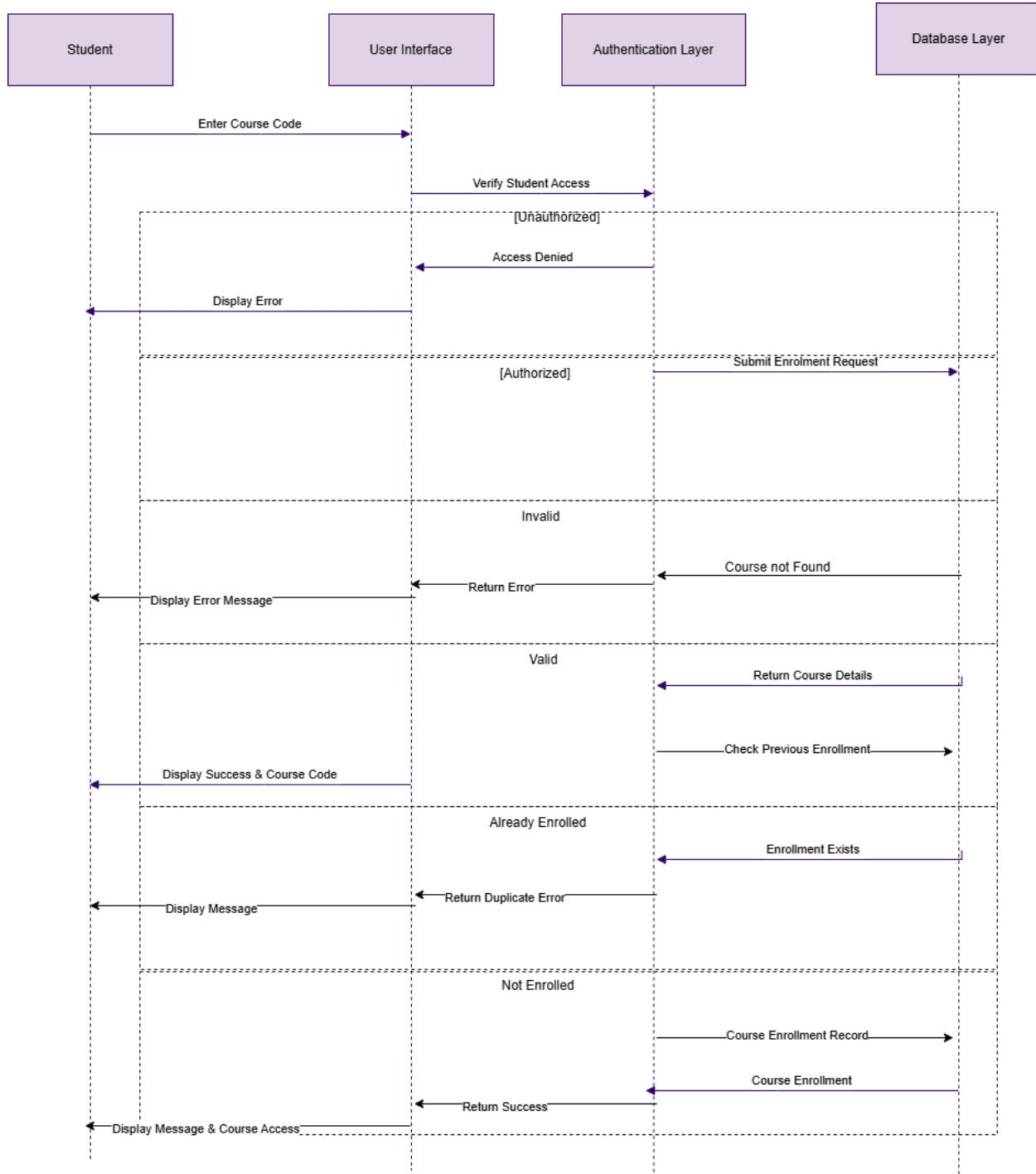
B. Signup Sequence Diagram



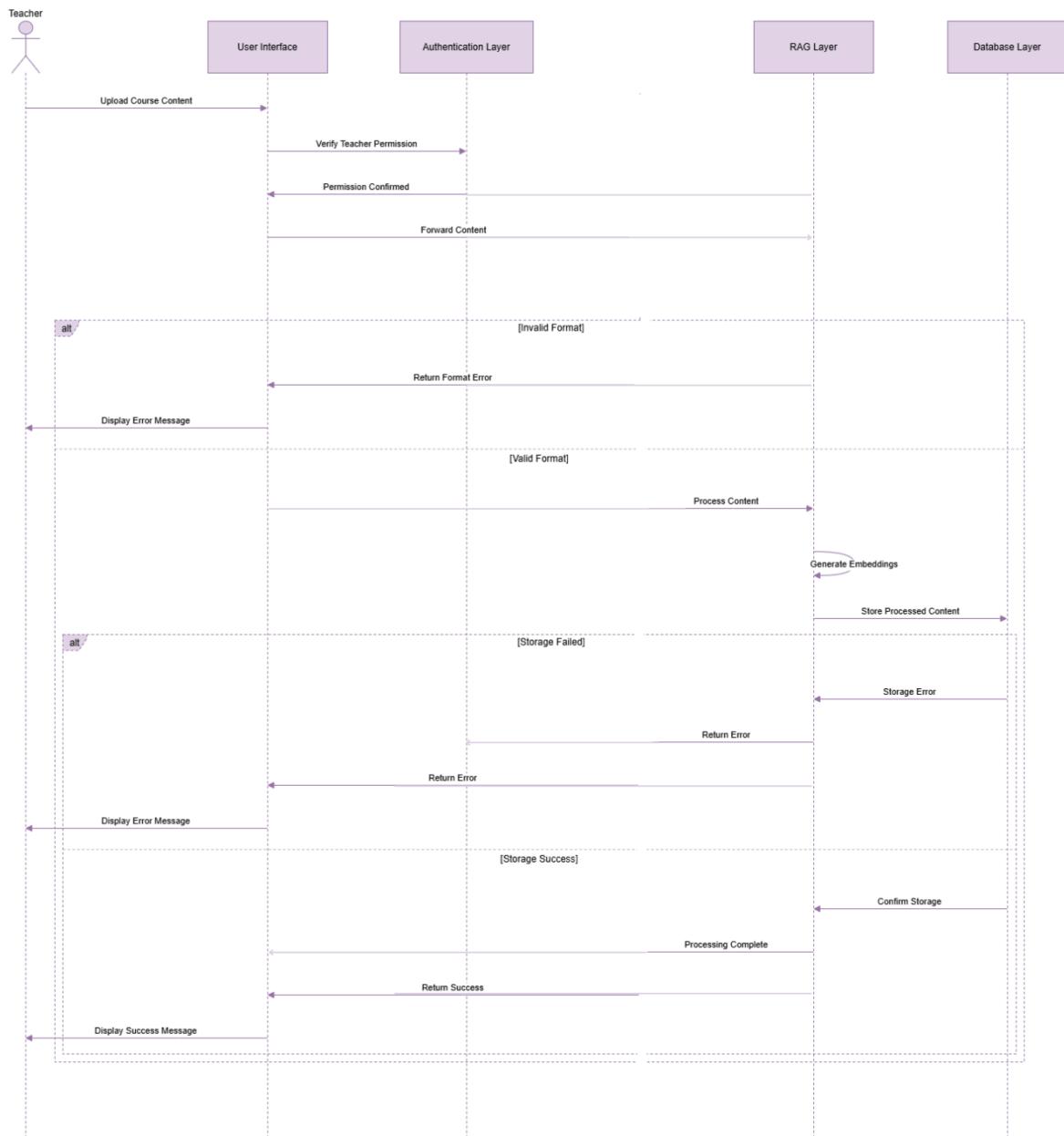
C. Creating a Course Sequence Diagram



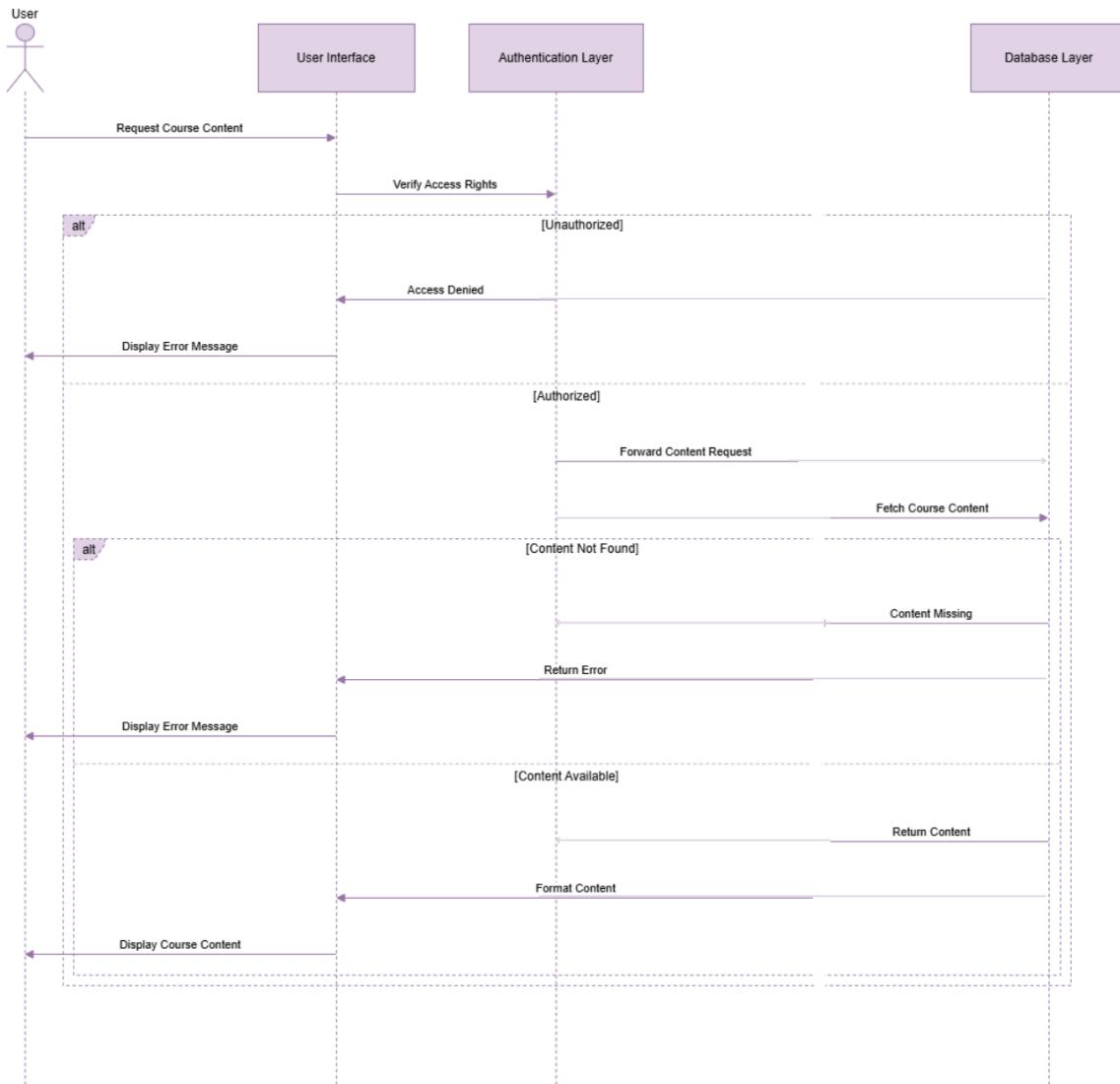
D. Enrolling in a Course Sequence Diagram



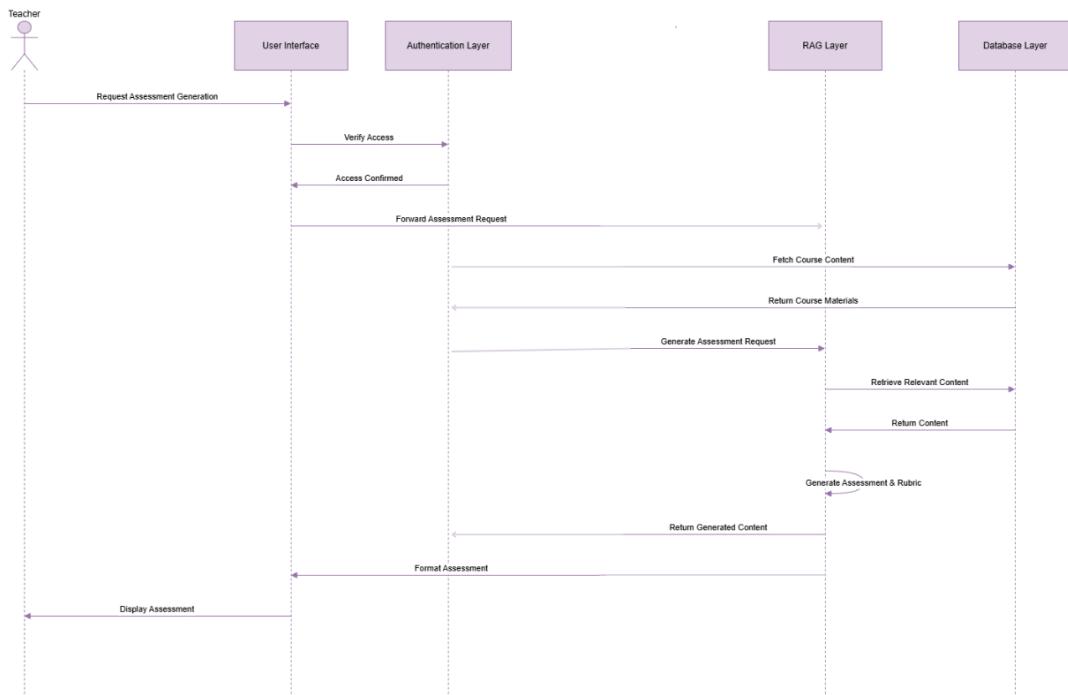
E. Adding Content Sequence Diagram



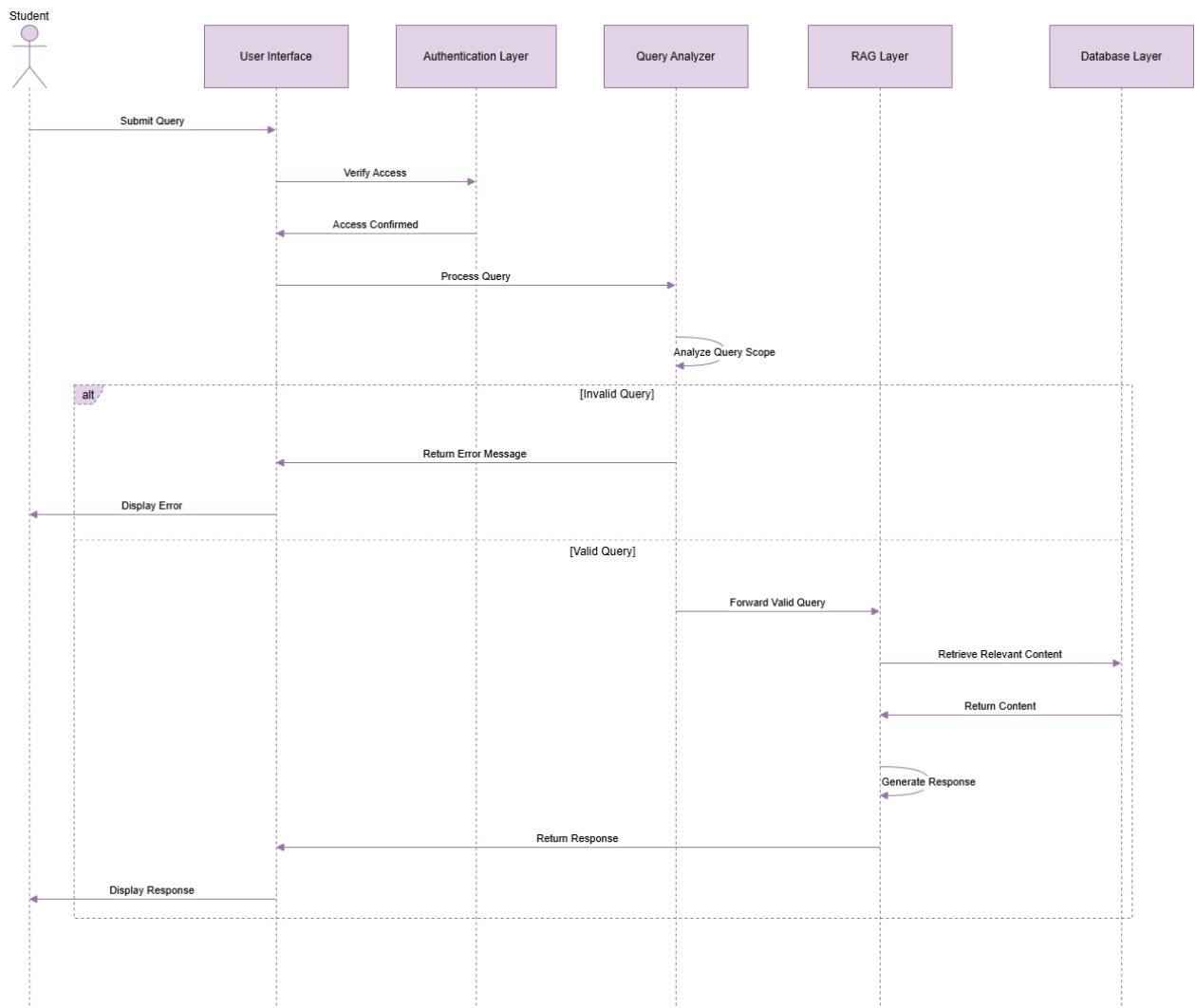
F. Viewing Content Sequence Diagram



G. Assessment Generation Sequence Diagram



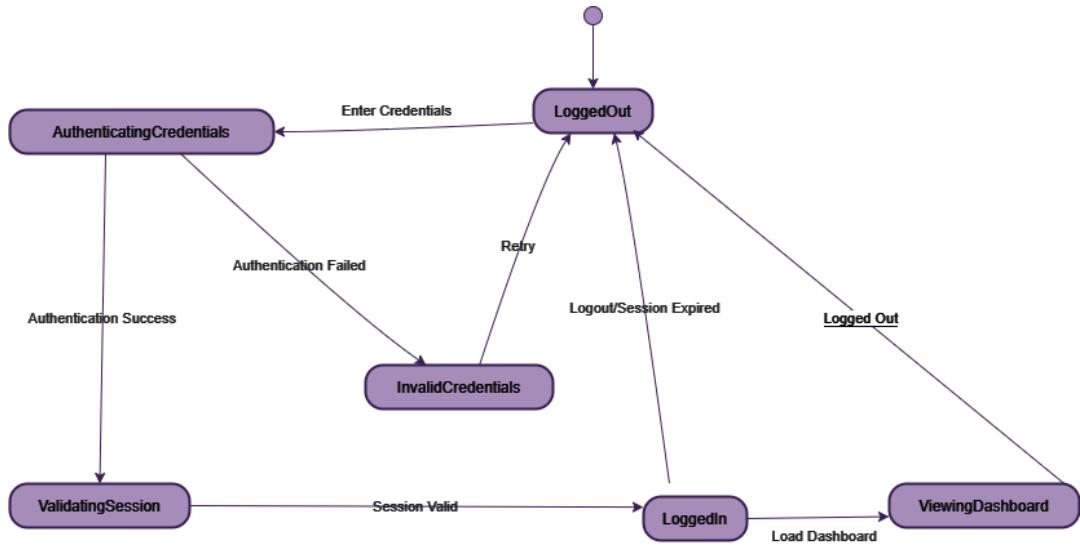
H. Content Querying Sequence Diagram



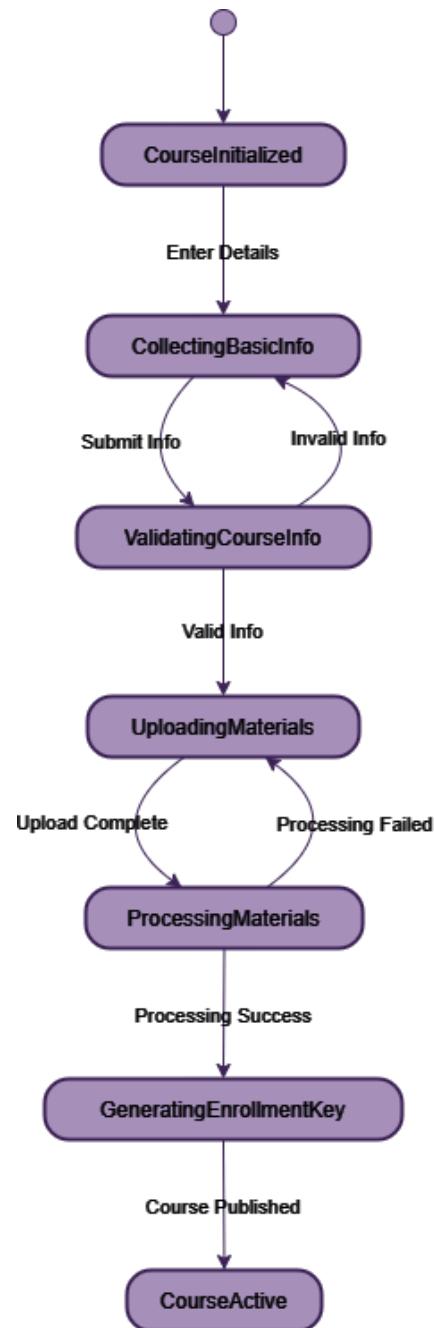
5.2 Data Flow Diagram

5.3 State Transition Diagram

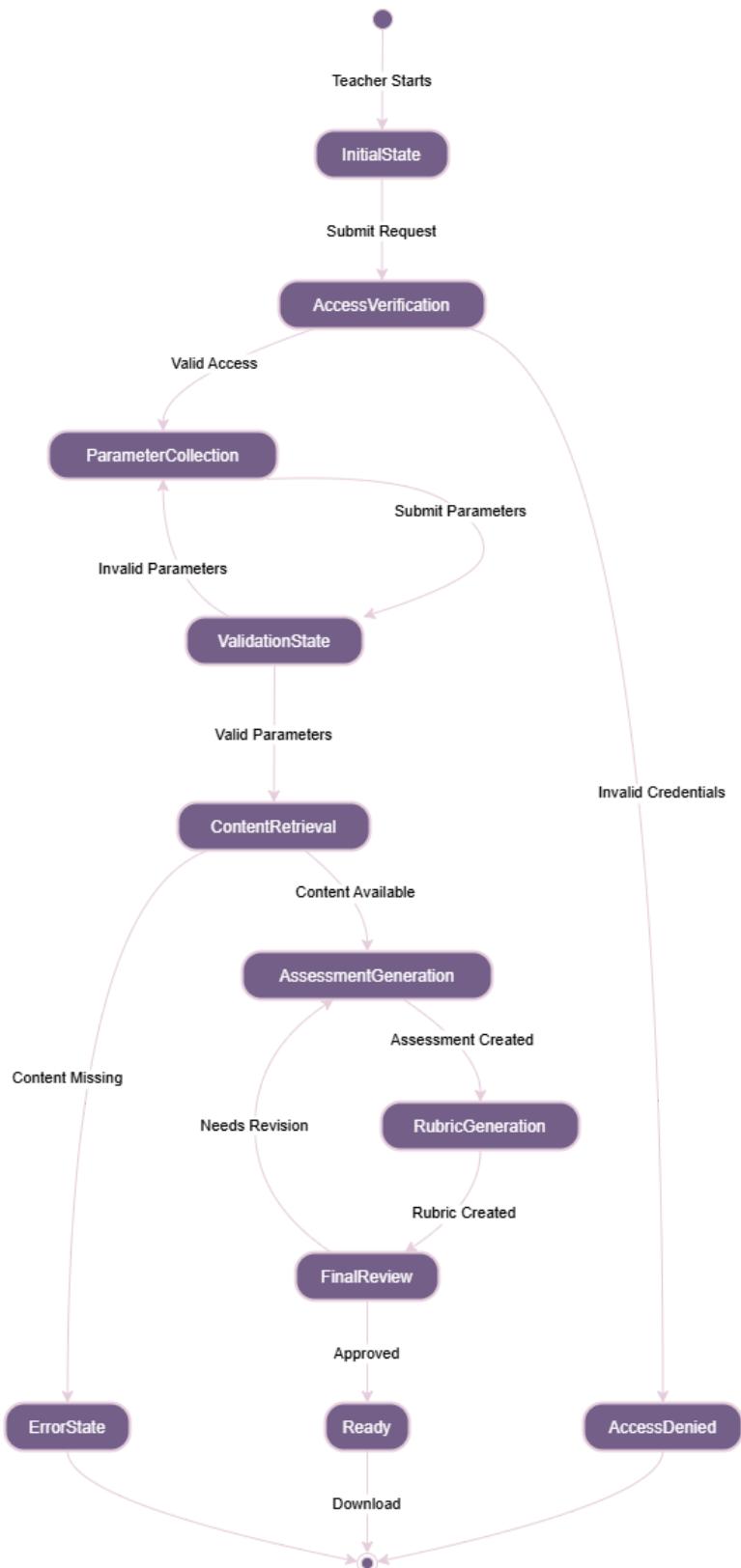
5.3.1. Authentication State Flow



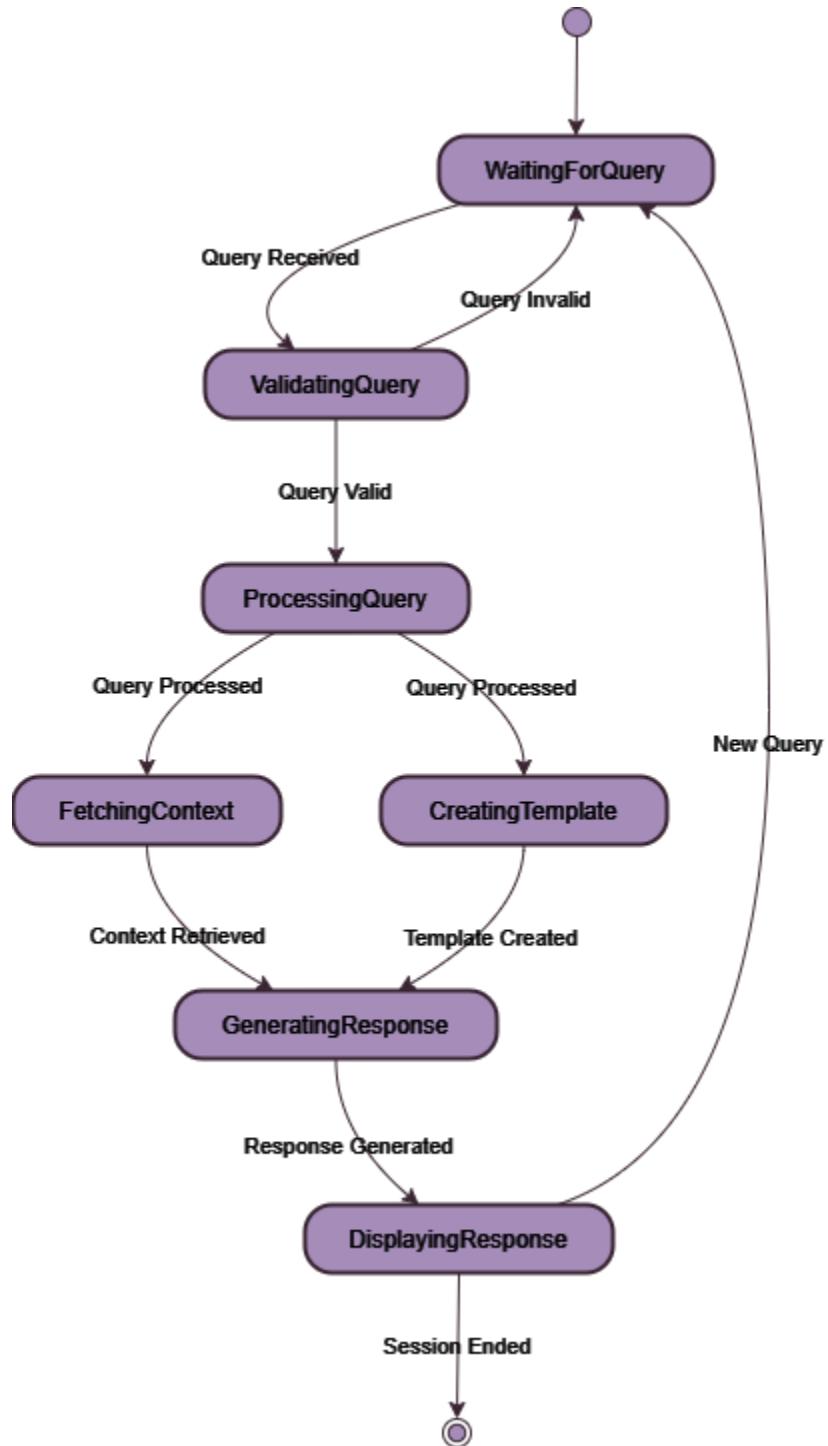
5.3.2 Course Creation State Flow



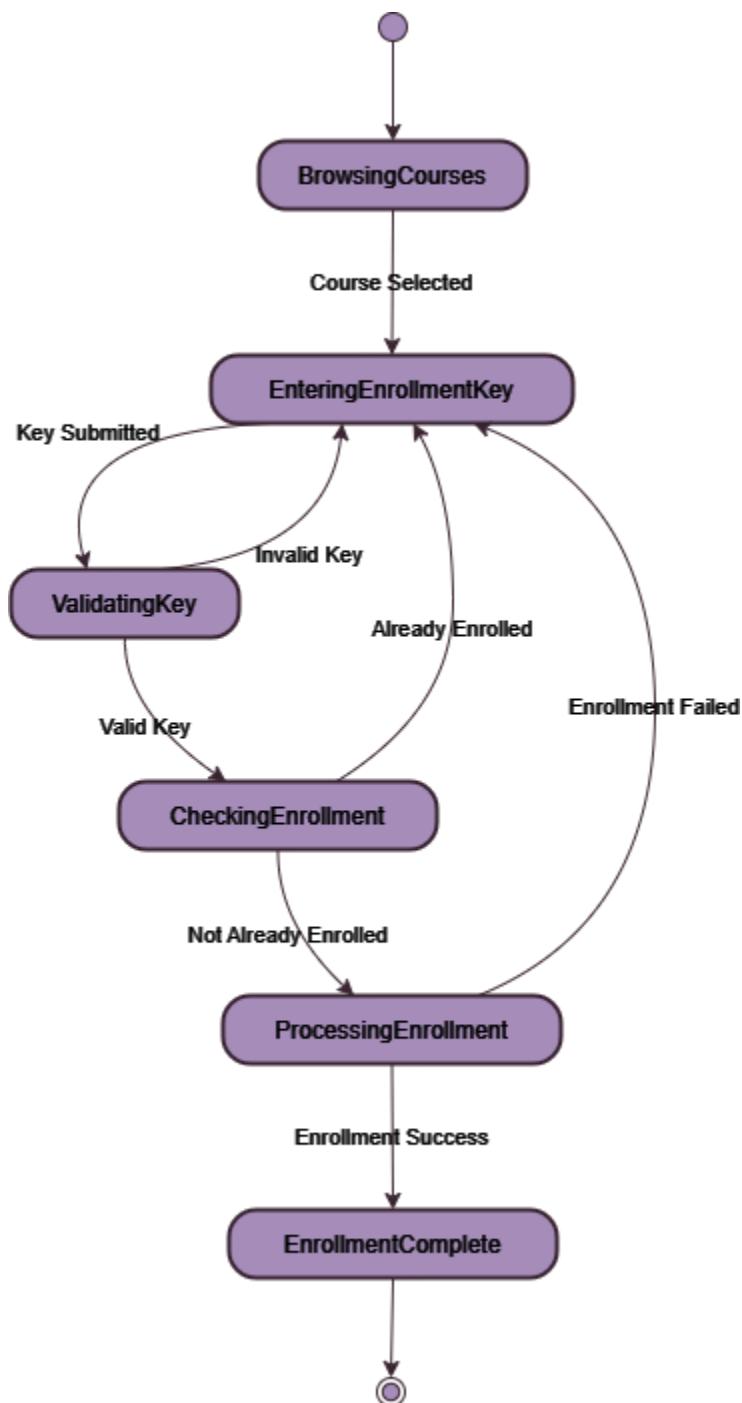
5.3.3 Assessment Generation State Flow:



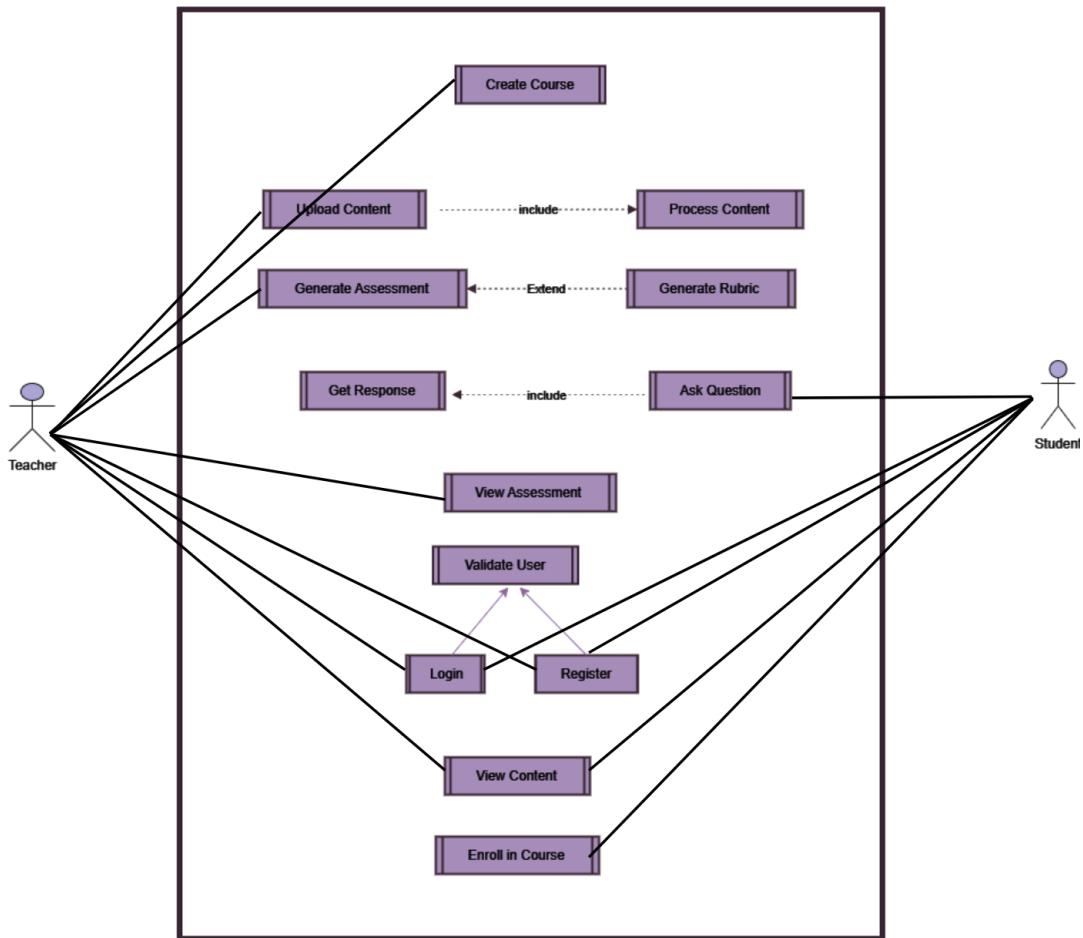
5.3.4 VTA Query State Flow:



5.3.5 Course Enrollment State Flow:

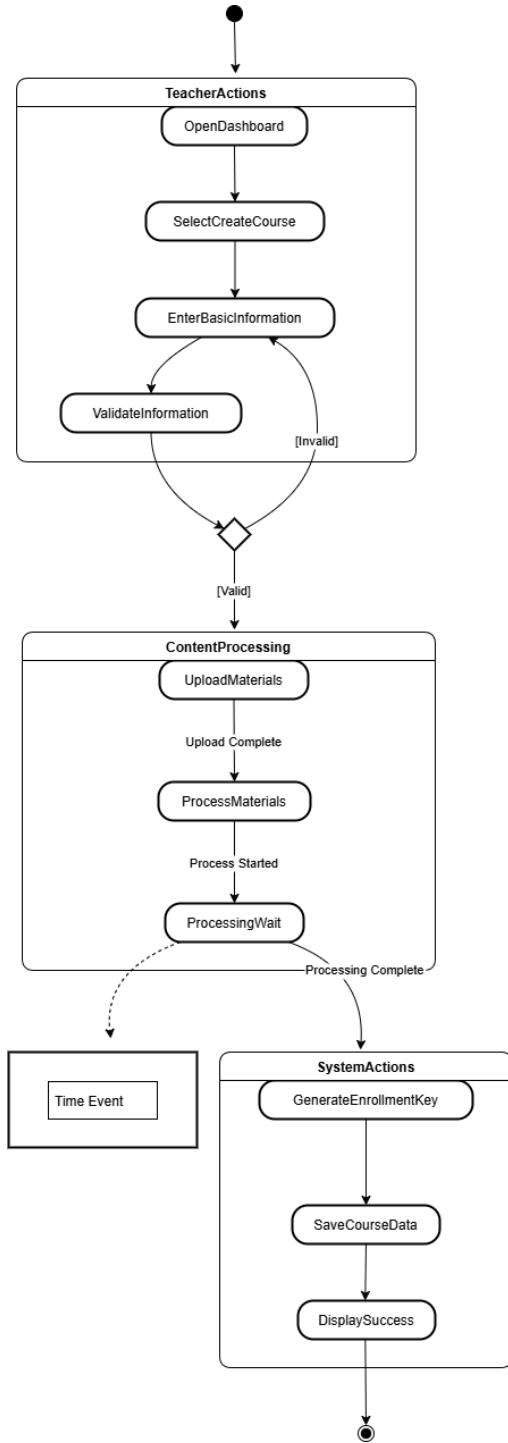


5.4 Use Case Diagram

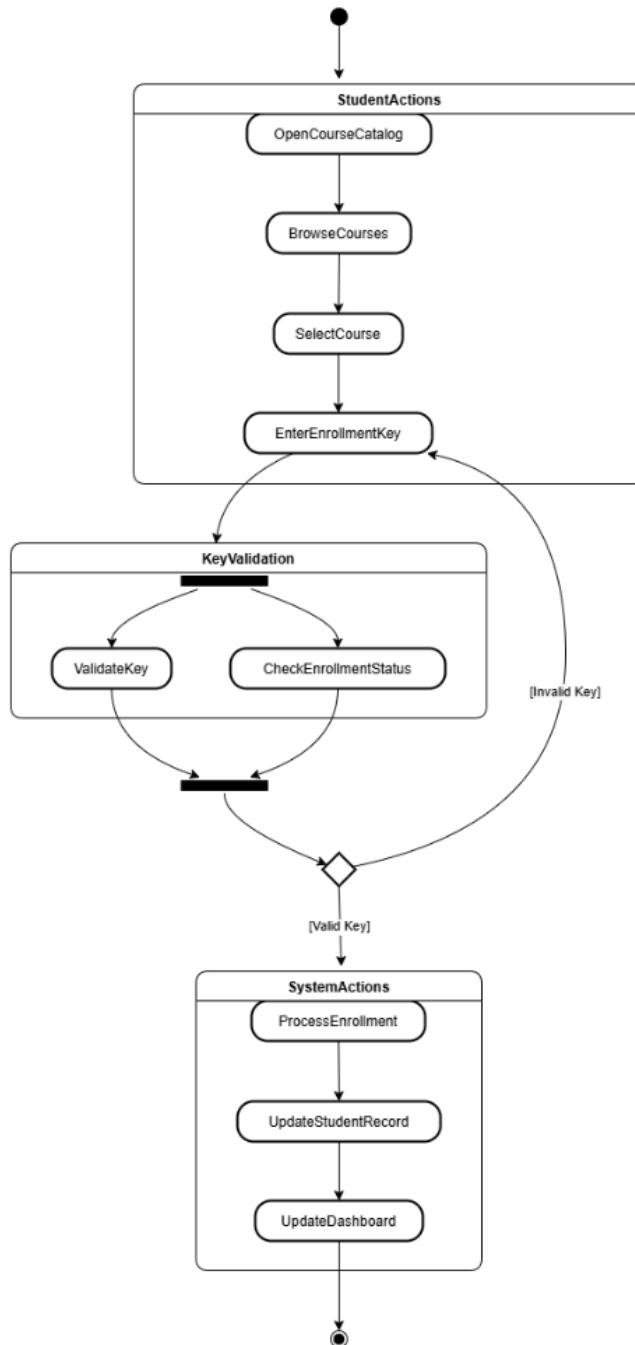


5.5 Activity Diagram

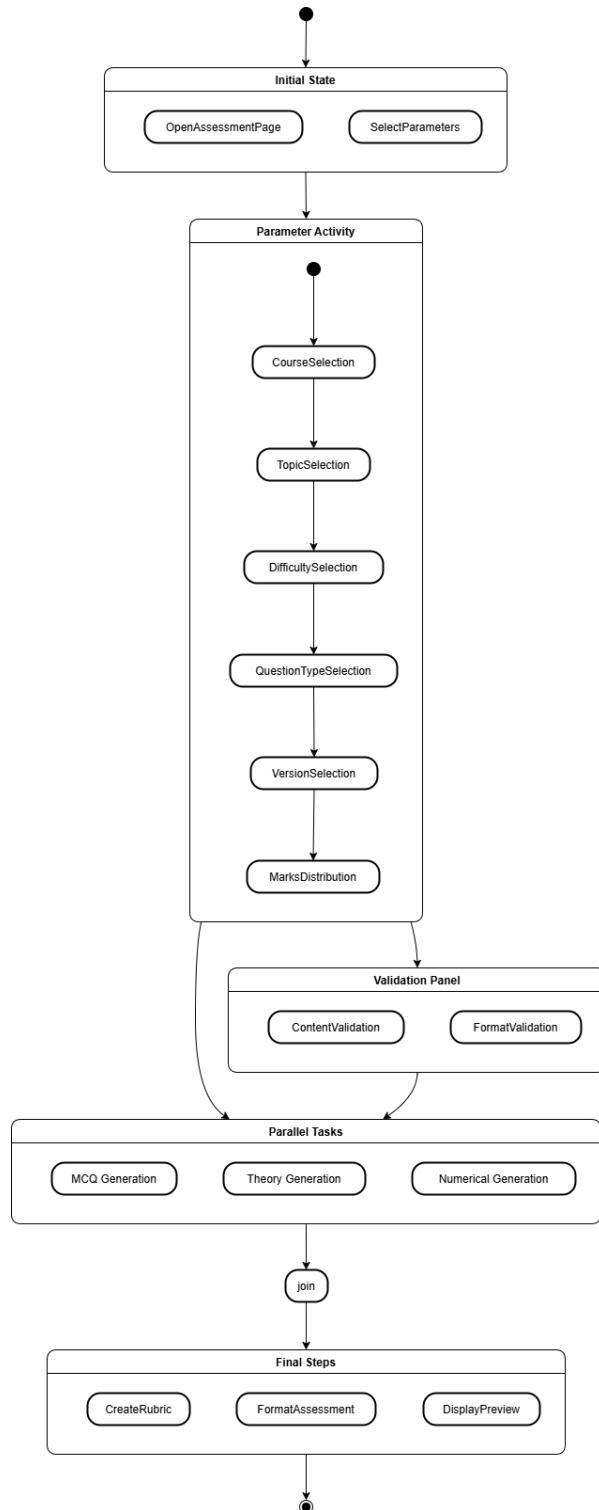
5.5.1 Course Creation Process



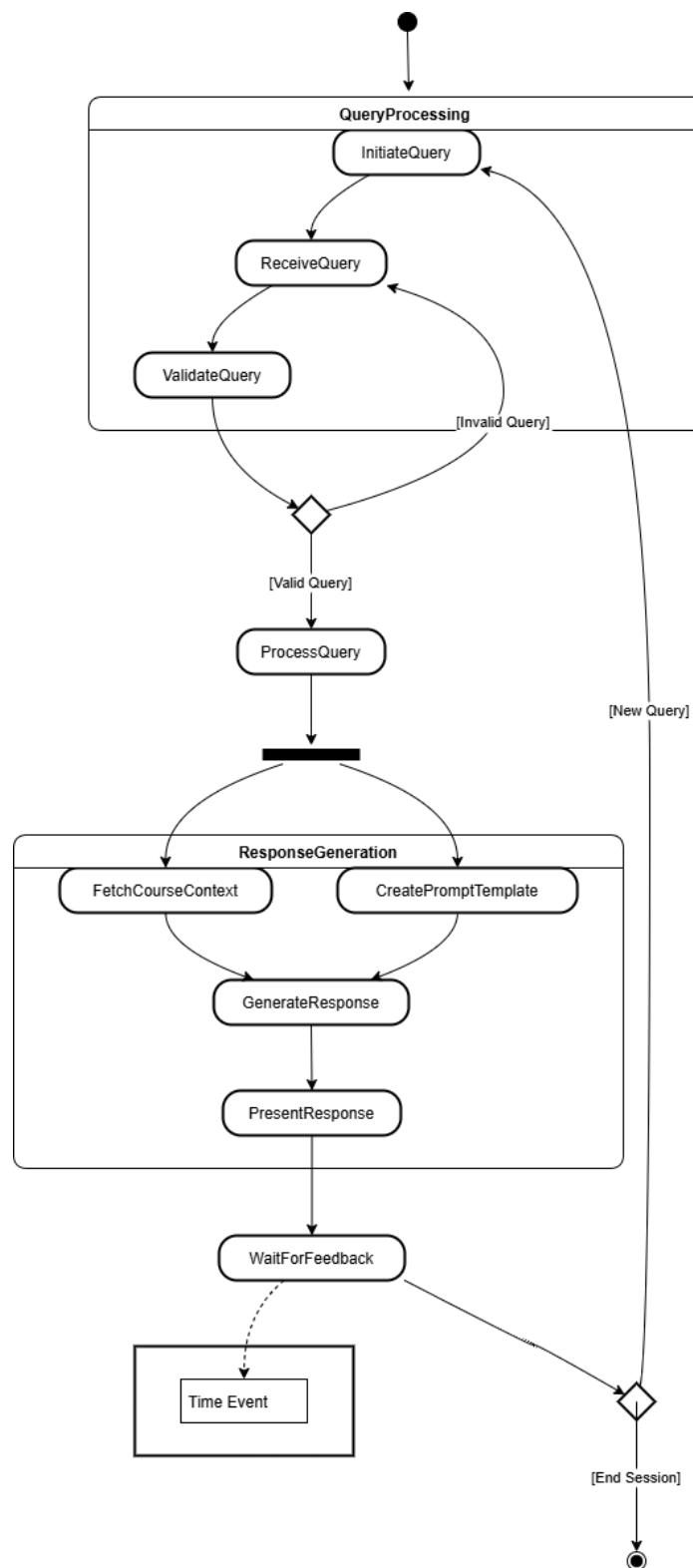
5.5.2 Student Enrollment



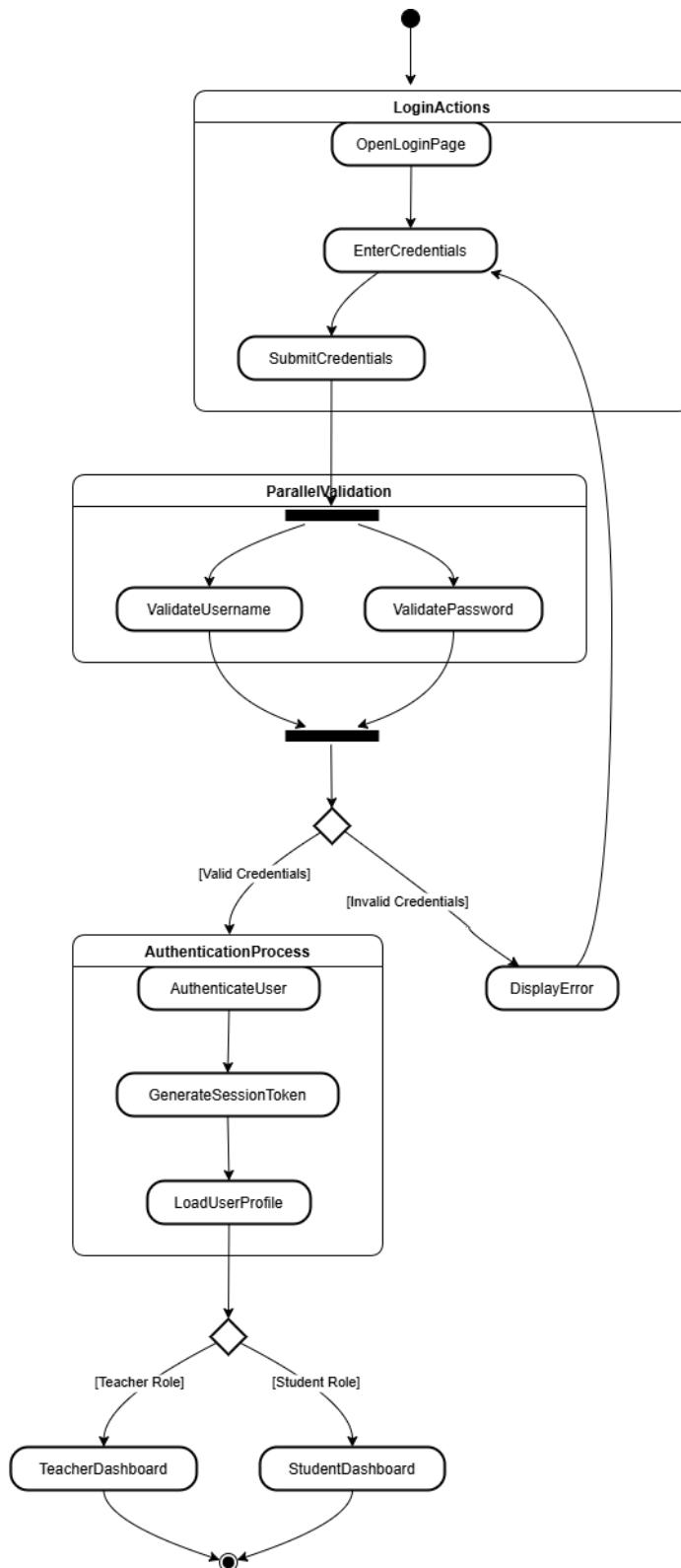
5.5.3 Assessment Generation



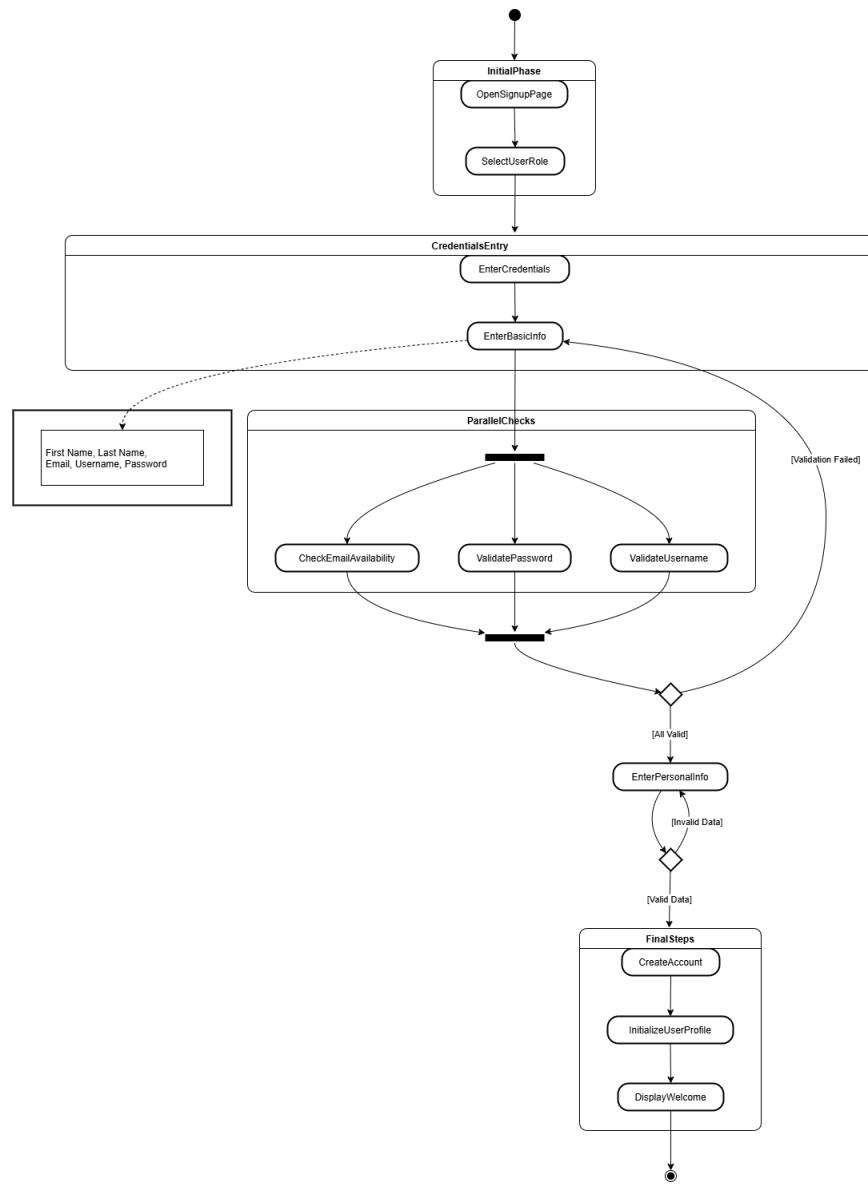
5.5.4 VTA Query



5.5.5 Login



5.5.6 Sign Up



6. Data Design

Information Domain Transformation and Data Structure Organization

1. **User Management Domain** The user management domain is transformed into a hierarchical data structure with a base **User** class and specialized role-based classes:
 - **Base User Entity:**
 - Captures fundamental user information
 - Provides a universal authentication and management mechanism

- Stores core identity and authentication details
- Supports both student and teacher roles through inheritance
- Key transformations:
 - User identity → UUID-based identification
 - Authentication → Secure password hashing
 - User status tracking → Active/inactive flags and timestamps
- **Role-Specific Specialization:**
 - **Student** class extends base user with academic-specific attributes
 - Stores university details
 - Tracks academic progression (semester, domain)
 - Enables course enrollment operations
 - **Teacher** class extends base user with instructional attributes
 - Captures professional details (department, specialization)
 - Enables course creation and management capabilities

2. **Course Management Domain** Transformed into a flexible, normalized data structure:

- **Course Entity:**
 - Represents the core educational offering
 - Unique identification via UUID
 - Metadata storage for course details
 - Supports dynamic content management
 - Implements enrollment control mechanisms
 - Key transformations:
 - Course information → Structured, queryable format
 - Content management → Flexible, extensible storage
 - Access control → Enrollment key mechanism
- **Course Content Representation:**
 - Separated from core course entity for flexibility
 - Supports multiple content types (lecture notes, slides, videos)
 - Enables granular content management
 - Allows easy content addition, update, and deletion

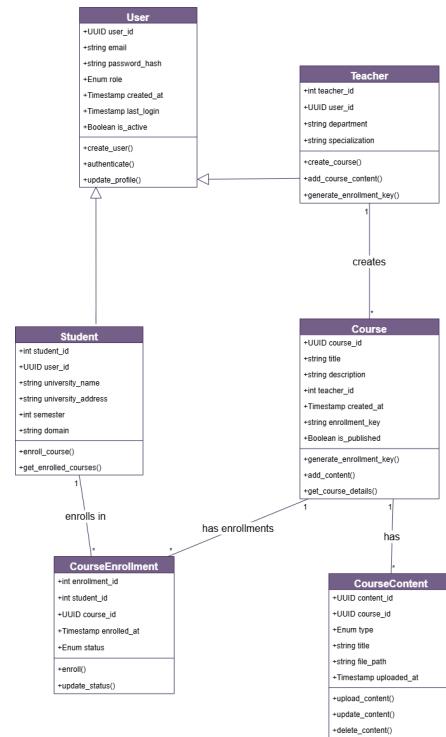
3. **Enrollment Management Domain** Implemented through a dedicated relationship entity:

- **Course Enrollment:**
 - Manages many-to-many relationships between students and courses
 - Tracks enrollment status and timestamp
 - Enables complex enrollment workflows
 - Key transformations:
 - Enrollment tracking → Detailed status management
 - Relationship representation → Normalized database structure

6.1 ERD Diagram

The class diagram for the Educational Platform represents a sophisticated, normalized data model that systematically transforms the educational information domain into structured entities.

At its core, a base User class serves as the foundational element, with specialized Student and Teacher classes inheriting core user attributes and extending them with role-specific characteristics. The diagram illustrates complex relationships between entities: teachers can create multiple courses, each course can have multiple contents and enrollments, and students can enroll in various courses through a many-to-many CourseEnrollment relationship. Key design principles include using UUID-based identifiers for enhanced security, implementing comprehensive metadata tracking with timestamps and status flags, and providing method-level operations that support authentication, profile management, course creation, content management, and enrollment processes. The class structure ensures data normalization by separating concerns, minimizing redundancy, and enabling flexible, scalable data management across the educational platform's ecosystem, with each class possessing carefully designed attributes and methods that reflect the specific functional requirements of user roles, course management, and academic interactions.



6.2 Data Dictionary

6.2.1 Course

Attribute	Type	Description
course_id	UUID	Unique identifier for each course
created_at	Timestamp	Date and time of course creation
description	TEXT	Detailed description of the course
enrollment_key	VARCHAR	Unique key for course enrollment
is_published	Boolean	Indicates if course is publicly available
teacher_id	Integer	Foreign key linking to course creator
title	VARCHAR	Name of the course

6.2.2 CourseContent

Attribute	Type	Description
content_id	UUID	Unique identifier for course content
course_id	UUID	Foreign key linking to parent course
file_path	VARCHAR	Storage path for content file
title	VARCHAR	Name of the content item
type	ENUM	Type of content (e.g., lecture_notes, slides, video)
uploaded_at	Timestamp	Date and time of content upload

6.2.3 CourseEnrollment

Attribute	Type	Description
course_id	UUID	Foreign key linking to enrolled course
enrolled_at	Timestamp	Date and time of course enrollment
enrollment_id	Integer	Unique identifier for enrollment record
status	ENUM	Enrollment status (e.g., active, completed, dropped)
student_id	Integer	Foreign key linking to enrolled student

6.2.4 Student

Attribute	Type	Description
domain	VARCHAR	Academic domain or field of study
semester	Integer	Current academic semester
student_id	Integer	Unique identifier for the student
university_address	VARCHAR	Address of the student's university
university_name	VARCHAR	Name of the student's university
user_id	UUID	Foreign key linking to the user account

6.2.5 Teacher

Attribute	Type	Description
department	VARCHAR	Academic department of the teacher

<code>specialization</code>	VARCHAR	Teacher's area of expertise
<code>teacher_id</code>	Integer	Unique identifier for the teacher
<code>user_id</code>	UUID	Foreign key linking to the user account

6.2.6 User

Attribute	Type	Description
<code>created_at</code>	Timestamp	Account creation timestamp
<code>email</code>	VARCHAR	User's email address
<code>is_active</code>	Boolean	Account active status
<code>last_login</code>	Timestamp	Most recent login time
<code>password_hash</code>	VARCHAR	Encrypted password
<code>role</code>	ENUM	User role (e.g., student, teacher)
<code>user_id</code>	UUID	Unique identifier for the user

7. User Interface Design

7.1 TALIM System Functionality - User Perspective

TALIM provides two distinct user experiences through its web interface: one for teachers and one for students. Each user starts by choosing their role during registration and accessing their personalized dashboard after authentication.

7.1.1 Teacher Experience

Teachers interact with TALIM through an organized dashboard that displays their course statistics and teaching materials. The system offers two primary functions:

1. Course Management

- Create and manage courses through a step-by-step process
- Upload and organize course materials with real-time progress tracking
- Generate unique enrollment keys for student access

2. Assessment Generation

The assessment generation process is streamlined through a single interface where teachers can:

- Select question types, counts, and difficulty levels
- Generate multiple versions of assessments and rubrics
- Download generated materials immediately

7.1.2 Student Experience

Students access TALIM's learning environment through a comprehensive dashboard that shows their enrolled courses and progress. The learning experience includes:

1. Course Interaction

TALIM provides an intuitive learning environment where students can browse available courses and enroll using provided keys. Once enrolled, they can access:

- Organized course materials
- Downloadable lecture notes
- Interactive learning resources

2. Virtual Teaching Assistant

The VTA feature offers immediate learning support by:

- Answering course-related queries
- Providing context-aware explanations
- Supporting continuous learning outside class hours

7.1.3 System Feedback

TALIM will ensure users stay informed through:

- Success/error messages for all actions
- Progress indicators for ongoing operations
- Clear navigation breadcrumbs
- Helpful tooltips for new users

The interface will adapt to different devices while maintaining consistency and ease of use across all features.

7.2 Screens

7.2.1 Signup

Objects: Input fields (Name, Email, Password), Signup button, Login Button.

Actions: Enter user details, click the Create Account button to create an account, or login if you already have an account.

Already have an Account? [Sign In](#)

Create your account

Student Teacher

Full Name

First name... Last name...

Username

Username...

Email

Email address

Password

Create password Confirm Password

Create Account →

7.2.2 Personal Information

Objects: Input fields (University Name, University Address, Semester, Domain), Create Account button, Login Button.

Actions: Fill personal details, and save changes or Login if you already have an account.



TALIM

Already have an Account? [Sign In](#)

PERSONAL INFORMATION

University Name

University Address

Semester

Domain

[Create Account →](#)

Email Password Confirm password [Forgot password?](#)

[Create Account →](#)

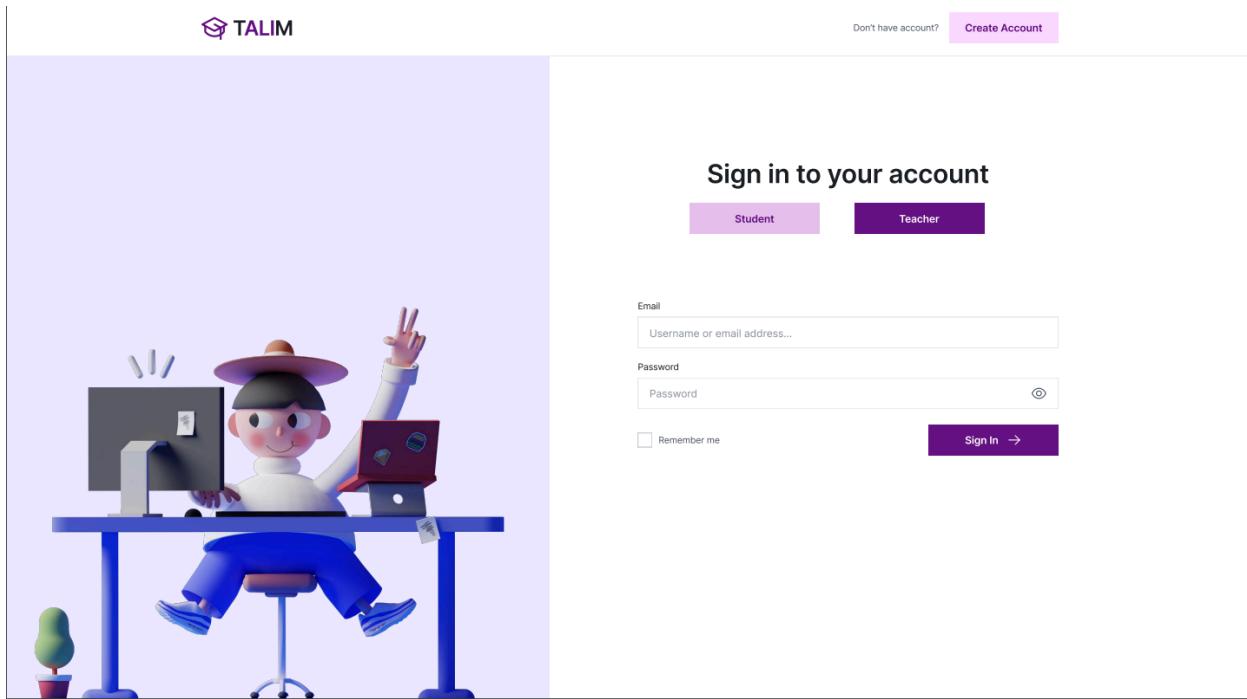


 Apple

7.2.3 SignIn

Objects: Input fields (Email, Password), SignIn button, Create Account Button.

Actions: Enter credentials, sign in, or Create Account if not already signed up.



7.2.4 Student Dashboard

Objects: Course List, Dashboard Navigation Menu, All Courses Button, LogOut Button.
Actions: Navigate to different course details or view all courses or logout.

The dashboard for student Mia Parker shows her profile picture and name. It includes a 'Dashboard' section with three stats: 957 Enrolled Courses, 6 Active Courses, and 951 Completed Courses. Below this is a section titled 'Let's start learning, Mia' with two course cards: 'Reiki Level I, II and Master/Teacher Program' and 'Copywriting - Become a Freelance Copywriter...'. Both cards have an 'Open Course' button. The bottom of the dashboard includes links for 'FAQs', 'Privacy Policy', and 'Terms & Condition'.

7.2.5 Teacher Dashboard

Objects: Course Management Options (View Courses, Create New Course), LogOut Button, Dashboard Navigation Menu.

Actions: Manage courses, LogOut of Account, or navigate to relevant sections.

The screenshot shows the TALIM teacher dashboard. On the left is a dark sidebar with a logo, 'TALIM' text, and three navigation items: 'Dashboard' (selected), 'Create New Course', and 'My Courses'. The main content area has a header with a profile picture of 'MIA PARKER' (Assistant Professor) and a 'All Courses →' button. Below this is a 'Dashboard' section with two cards: one showing '957 Total Courses' and another showing '6 Active Courses'. A 'Lets Teach, Mia' section follows, featuring two course thumbnails: 'Reiki Level I, II and Master/Teacher Program' (with '1. Introductions') and 'Copywriting - Become a Freelance Copywriter...' (with '1. How to get started with figma'). Each thumbnail has an 'Open Course' button. At the bottom are links for 'FAQs', 'Privacy Policy', and 'Terms & Condition', along with a 'Sign-out' button.

7.2.6 Student's Courses

Objects: List of Enrolled Courses.

Actions: Select a course to view details or chat with AI.

The screenshot shows the TALIM platform interface. On the left, a dark sidebar features the TALIM logo, a dashboard link, and a purple "My Courses" button. The main content area has a white header with a user profile for "MIA PARKER" (7th Semester) and a purple "Enroll In A Course →" button. Below this is a section titled "My Courses" with two course cards. The first card, "Reiki Level I, II and Master/Teacher Program", shows a person sitting on a chair with floating papers and includes an "Open Course" button. The second card, "Copywriting - Become a Freelance Copywriter...", shows a desk with a cup and books and also includes an "Open Course" button. At the bottom, there are links for "FAQs", "Privacy Policy", and "Terms & Condition".

7.2.7 Enroll Courses

Objects: Course Search Bar, Enroll Button.

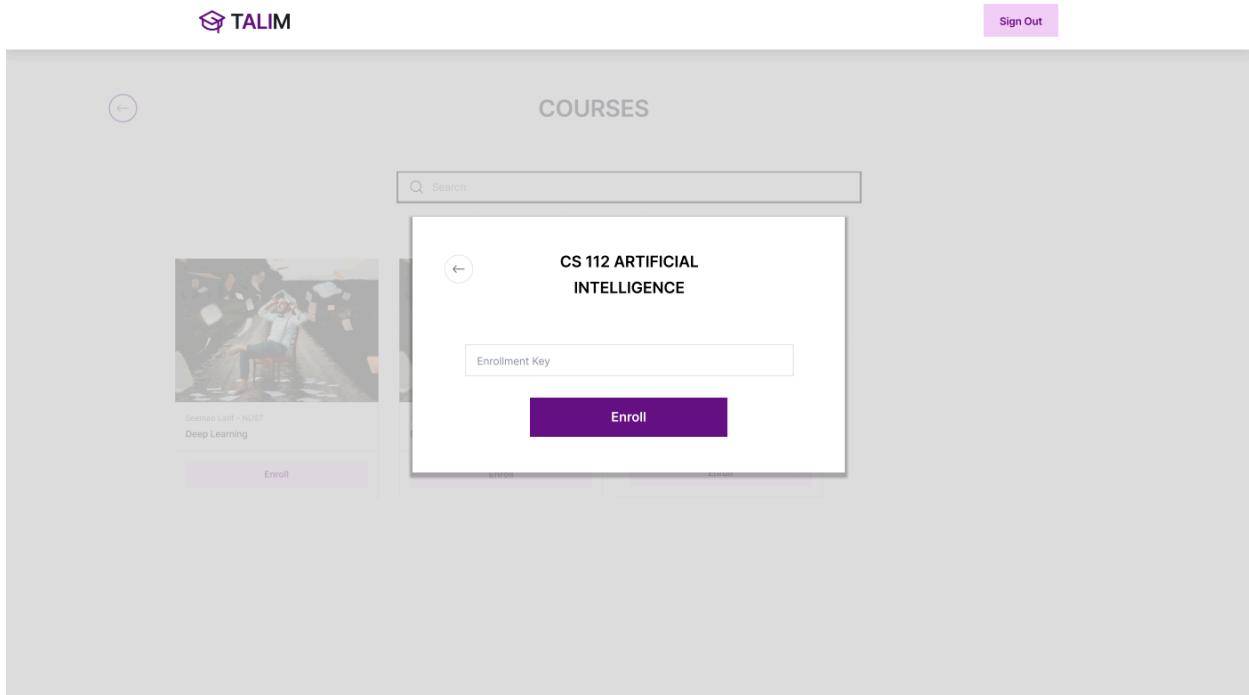
Actions: Search for courses, enroll by selecting the desired course.

The screenshot shows the TALIM platform interface. At the top, the TALIM logo and a "Sign Out" button are visible. Below is a header with a back arrow and the word "COURSES". A search bar with a magnifying glass icon and the placeholder "Search" is present. The main area displays three course cards for "Seemab Latif - NUST Deep Learning", each showing a person sitting on a chair with floating papers and an "Enroll" button. The cards are identical in layout and content.

7.2.8 Enrollment Key Popup

Objects: Input Field for Enrollment Key, Submit Button.

Actions: Enter the enrollment key to gain course access.



7.2.9 Course Window

Objects: Course Content List, Lecture Notes, Lecture Slides, VTA.

Actions: Access specific course materials or Chat with VTA.

Artificial Intelligence Course

Dashboard / Artificial Intelligence Course

COURSE

	Introduction to AI Course by: Seemab Latif	Lecture Slides Lecture Notes
	AI Intelligence Course by: Seemab Latif	Lecture Slides
	Figma in AI Course by: Seemab Latif	Lecture Slides

Hey There! Do you have
any questions?

7.2.10 Virtual Educational Assistant

Objects: Chat Interface, Input Field, Send Button.**Actions:** Ask questions, interact with the assistant for concept clarification or content access.

Wed 8:21 AM

TALIM Bot
Always active Hello! I am TALIM, your personal educational assistant. How may I help you? What is AI? Artificial intelligence is a technology that allows you to generate, classify, and perform tasks like image analysis and speech recognition.

Type a message...



7.2.11 Teacher's Courses

Objects: List of Courses Created by the Teacher.

Actions: Select a course to add content or generate assessments.

The screenshot shows the TALIM platform's main dashboard. On the left is a dark sidebar with the TALIM logo, navigation links for Dashboard, Create New Course, and My Courses, and a Sign-out button at the bottom. The main area features a user profile card for 'MIA PARKER' (Assistant Professor) with a photo, followed by two course cards. The first card is for 'Reiki Level I, II and Master/Teacher Program' with one lesson ('1. Introductions') and an 'Open Course' button. The second card is for 'Copywriting - Become a Freelance Copywriter...' with one lesson ('1. How to get started with figma') and an 'Open Course' button. At the bottom right of the main area are links for FAQs, Privacy Policy, and Terms & Condition.

7.2.12 Create New Course - Window 1

Objects: Course Title, Description Fields, Next Button.

Actions: Enter course details, proceed to the next step.

The screenshot shows the 'Create a new course' window. The left sidebar is identical to the previous dashboard screenshot. The main window has a header 'Good Morning' and 'Create a new course'. It features three tabs: 'Basic Information' (selected), 'Advance Information', and 'Curriculum'. The 'Basic Information' tab contains fields for 'Title' (with placeholder 'Your course title' and character count '0/80'), 'Course Category' (with placeholder 'Select...'), and 'Semester' (with placeholder 'Which Semester students do you teach your course?'). At the bottom are 'Cancel' and 'Save & Next' buttons. The footer includes links for FAQs, Privacy Policy, and Terms & Condition.

7.2.13 Create New Courses - Window 2

Objects: Course Poster and Description, Next Button.

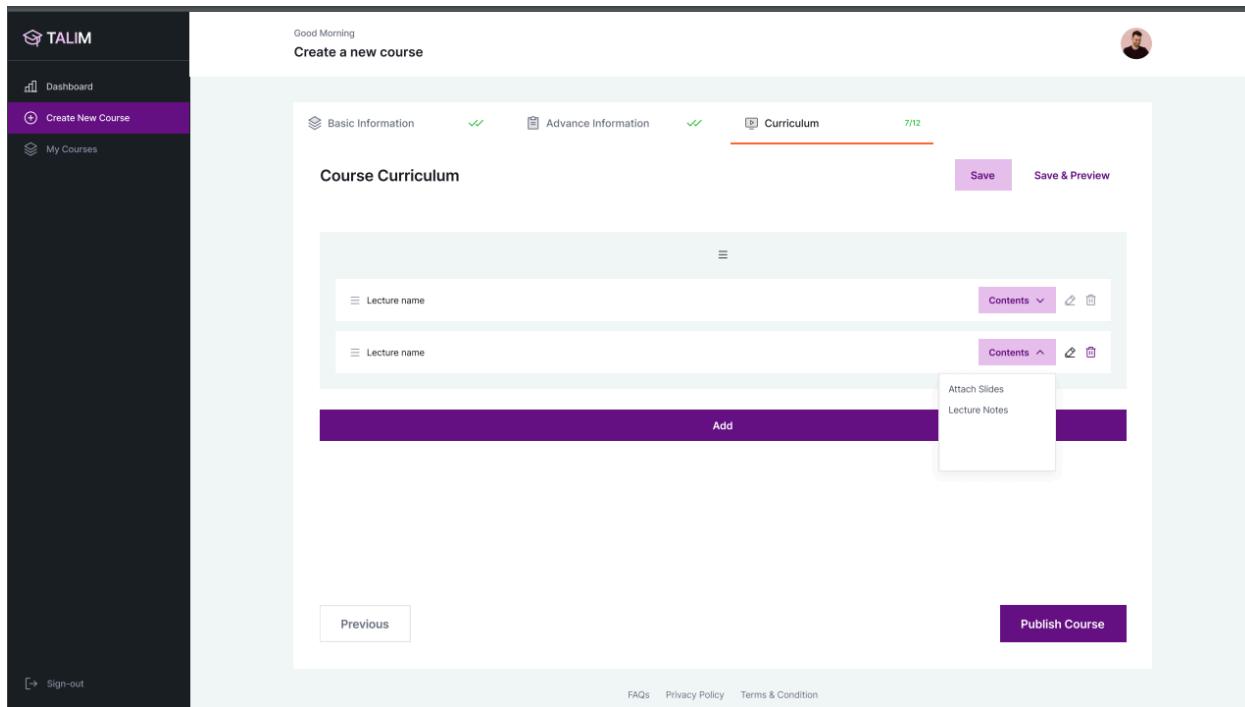
Actions: Enter course details, proceed to the next step.

The screenshot shows the TALIM platform's 'Create a new course' interface. On the left is a dark sidebar with icons for Dashboard, Create New Course (highlighted in purple), and My Courses. The main area has a header 'Good Morning' and 'Create a new course'. It's divided into three tabs: 'Basic Information' (green checkmark), 'Advance Information' (red underline, currently active), and 'Curriculum' (grey). The 'Advance Information' tab contains sections for 'Course Thumbnail' (with a placeholder image and 'Upload Image' button) and 'Course Descriptions' (with a rich text editor and 'Save & Next' button). Navigation buttons 'Previous' and 'Next' are at the bottom. The footer includes links for FAQs, Privacy Policy, and Terms & Condition.

7.2.14 Create New Courses - Window 3

Objects: Upload Content Section (Lecture Notes, Presentations), Publish Button.

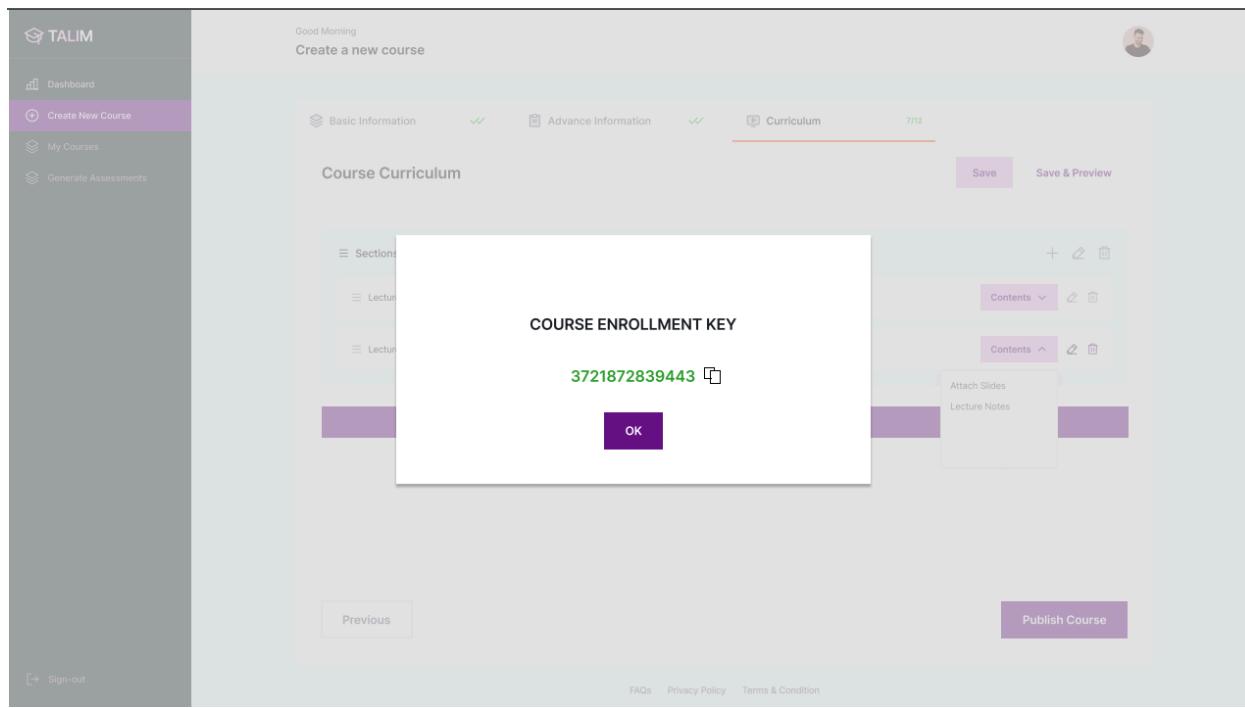
Actions: Upload course material and publish.



7.2.15 Enrollment Key Generated Popup

Objects: Generated Enrollment Key, Copy Button.

Actions: Copy the key for distribution to students.



7.2.16 Add Content to Existing Course

Objects: Upload Content Button, Content List.

Actions: Upload new materials.

The screenshot shows the TALIM platform interface. On the left is a dark sidebar with a logo, 'TALIM', and navigation links: 'Dashboard', 'Create New Course', and 'My Courses'. The 'My Courses' link is highlighted with a purple bar. The main content area has a light gray header 'Artificial Intelligence Course' with a breadcrumb 'Dashboard / Artificial Intelligence Course / Add Curriculum'. Below this is a large white panel containing two sections, each labeled 'Lecture name' with a dropdown arrow icon. To the right of each section is a 'Contents' button with a dropdown menu. The first dropdown menu contains 'Attach Slides' and 'Lecture Notes'. At the bottom of the panel are 'Previous' and 'Add' buttons. The footer of the page includes links for 'FAQs', 'Privacy Policy', and 'Terms & Condition'.

7.2.17 View Course

Objects: Course Overview, List of Uploaded Content, Buttons for Uploading Content and Generating Assessments.

Actions: View uploaded content and Add new content and generate assessments.

Artificial Intelligence Course

Dashboard / Artificial Intelligence Course

COURSE

	The Python Mega Course: Build 10 Real World Applications Course by: Seemab Latif	Lecture Slides Lecture Notes
	Machine Learning A-Z™: Hands-On Python & R In Data Science Course by: Seemab Latif	Lecture Slides
	Learn Ethical Hacking From Scratch Course by: Seemab Latif	Lecture Slides

[Generate Assignment](#) [Generate Quiz](#) [Add Curriculum](#)

7.2.18 Generate Quiz

Objects: Quiz Customization Options (Question Type, Difficulty, etc.), Generate Button.**Actions:** Set quiz parameters and generate the quiz.

Good Morning
Generate Quiz

Lecture Name
Lecture Title Here 0/80

Number of MCQs Select... Number of Theoretical Questions Select...

Number of Numericals Select... Difficulty Level Select...

Number of Versions Select... Generate Rubric Select...

[Cancel](#) [Save](#)

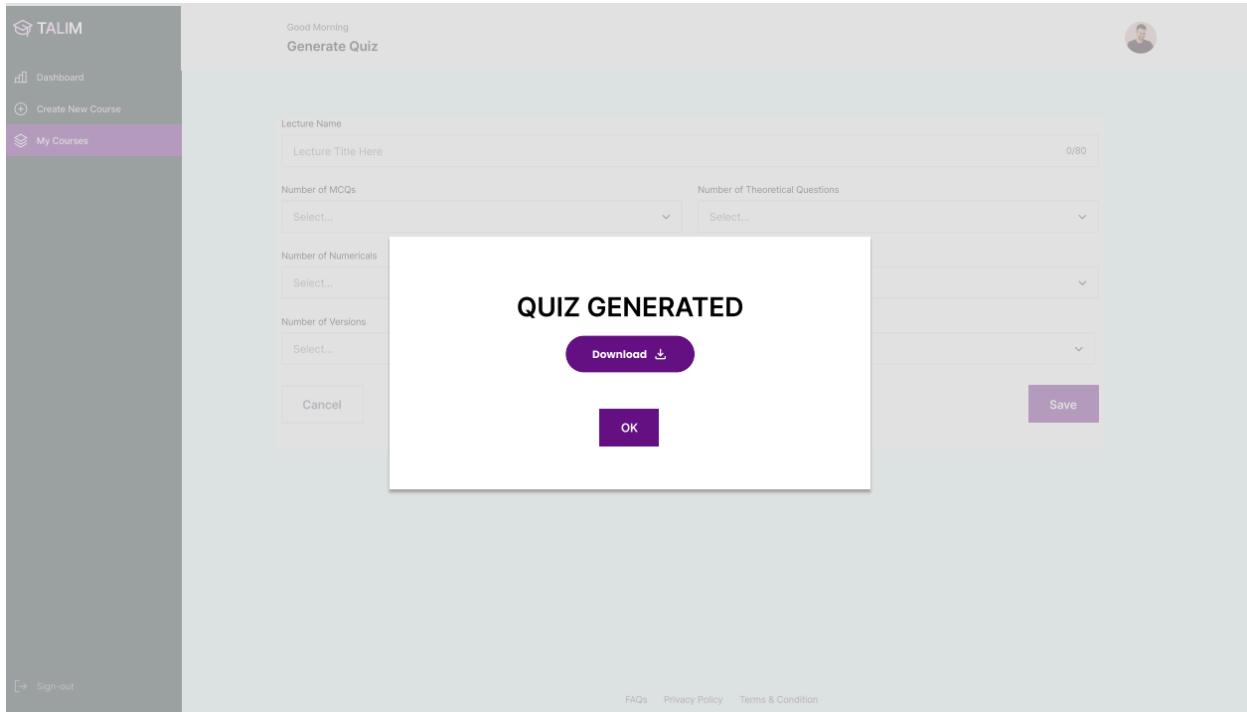
Sign-out

FAQs Privacy Policy Terms & Condition

7.2.19 Download Quiz Popup

Objects: Download Button.

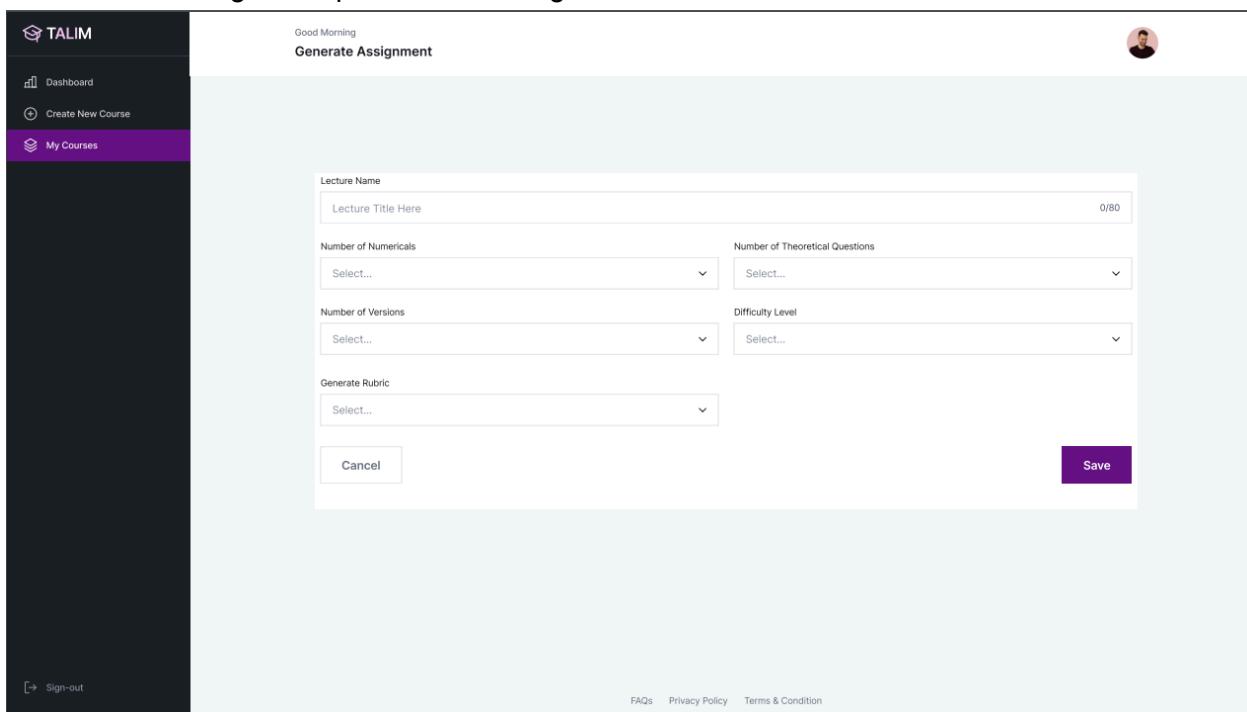
Actions: Download the generated quiz.



7.2.20 Generate Assignment

Objects: Assignment Customization Options (Topics, Type), Generate Button.

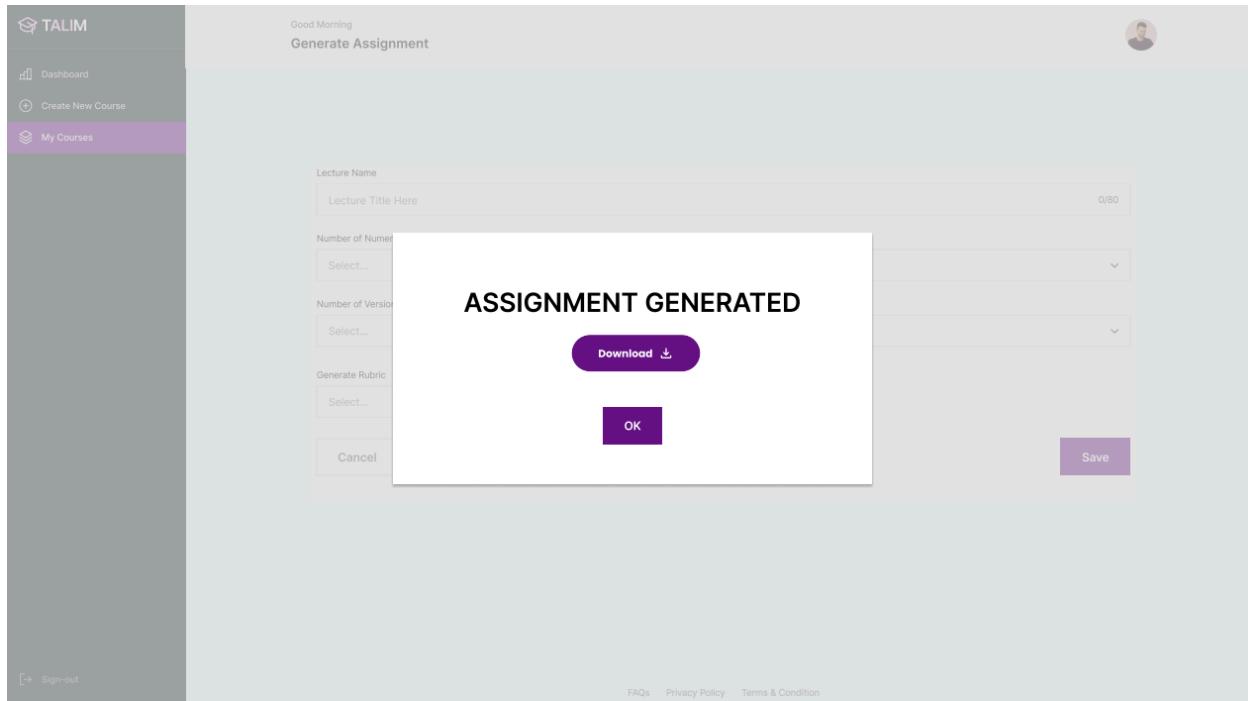
Actions: Set assignment parameters and generate it.



7.2.21 Download Assignment Popup

Objects: Download Button.

Actions: Download the generated assignment.



8. REFERENCES

1. Meta. *Llama-2-7b-chat-hf*. Available at: <https://huggingface.co/meta-llama/Llama-2-7b-chat-hf>. Accessed: October 2024.
2. AGI Sphere. *Retrieval-Augmented Generation with Llama2*. Available at: <https://agi-sphere.com/retrieval-augmented-generation-llama2/>. Accessed: October 2024.
3. The Brazilian Ba. *The Amazing ChatGPT: Functional and Non-Functional Requirements*. Available at: <https://thebrazilianba.com/2023/04/09/the-amazing-chatgpt-functional-and-non-functional-requirements/>. Accessed: October 2024.
4. Onlim. *Top Requirements for Chatbot Software*. Available at: <https://onlim.com/en/top-requirements-for-chatbot-software/>. Accessed: October 2024.
5. Yellow.ai. *Chatbot Features*. Available at: <https://yellow.ai/blog/chatbot-features/>. Accessed: October 2024.
6. Ivy.ai. *Example AI Chatbot Requirements*. Available at: <https://go.ivy.ai/hubfs/Marketing/Handouts%20and%201%20Pagers/Example%20AI%20%20Chatbot%20Requirements.pdf>. Accessed: October 2024.

7. Pakistan Credit Rating Agency (PACRA). *Educational Statistics*. Available at: https://www.pacra.com/view/storage/app/Education%20-%20PACRA%20Research%20-%20June%2723_1688738214.pdf. Accessed: August 2024.
8. Statista. *Projected Market Volume of EdTech Solutions*. Available at: <https://www.statista.com/outlook/emo/online-education/pakistan>. Accessed: September 2024.
9. ResearchGate. *Leveraging Chatbots to Improve Self-Guided Learning through Conversational Quizzes*. Available at: https://www.researchgate.net/publication/311508246_Leveraging_chatbots_to_improve_self-guided_learning_through_conversational_quizzes. Accessed: August 2024.
10. ResearchGate. *TA-bot: An AI Agent as a Teaching Assistant using Google's Conversational Technologies*. Available at: https://www.researchgate.net/publication/325332045_TA-bot_An_AI_agent_as_a_Teaching_Assistant_using_Google's_Conversational_Technologies. Accessed: August 2024.
11. Inria. *Rexy, a Configurable Application for Building Virtual Teaching Assistants*. Available at: <https://inria.hal.science/hal-02544593/document>. Accessed: August 2024.
12. ResearchGate. *Evolution in Education: Chatbots*. Available at: https://www.researchgate.net/publication/342357938_Evolution_in_Education_Chatbots. Accessed: August 2024.
13. ResearchGate. *Education 4.0: Artificial Intelligence-Assisted Task-and-Time Planning System*. Available at: https://www.researchgate.net/publication/359120286_Education_40_Artificial_Intelligence_Assisted_Task-and_Time_planning_system. Accessed: August 2024.
14. ResearchGate. *Kwame for Science: An AI Teaching Assistant for Science Education in West Africa*. Available at: https://www.researchgate.net/publication/361604415_Kwame_for_Science_An_AI_Teaching_Assistant_for_Science_Education_in_West_Africa. Accessed: August 2024.
15. arXiv. *Ethel: A Virtual Teaching Assistant*. Available at: <https://arxiv.org/abs/2407.19452>. Accessed: August 2024.
16. ResearchGate. *Virtual Teaching Assistant for Undergraduate Students using Natural Language Processing and Deep Learning*. Available at: https://www.researchgate.net/publication/378084679_Virtual_teaching_assistant_for_undergraduate_students_using_natural_language_processing_deep_learning. Accessed: August 2024.
17. UK Government. *Teacher Workload Survey*. Available at: https://assets.publishing.service.gov.uk/media/5e12fcb7e5274a0f9e82e4fd/teacher_workload_survey_2019_main_report_amended.pdf. Accessed: September 2024.
18. Pew Research. *How Teachers Manage Their Workload*. Available at: <https://www.pewresearch.org/social-trends/2024/04/04/how-teachers-manage-their-workload/>. Accessed: September 2024.
19. eGrove. *Forecasted Online Education Market*. Available at: <https://egrove.olemiss.edu/cgi/viewcontent.cgi?article=3167&context=etd>. Accessed: September 2024.

20. RSP Science Hub. *RSP Quiz Generation*. Available at: <https://rspsciencehub.com/index.php/journal/article/view/834/688>. Accessed: September 2024.