System Decomposition

3.2 System Decomposition

The Gradia system consists of six subsystems, each with a specific function to ensure modularity, scalability, and maintainability.

Subsystems & Responsibilities

Subsystem	Responsibilities
Frontend Web Application	Provides user interfaces for teachers and students, including dashboards for grades, essay submissions, and class management.
User Management Module	Manages authentication, registration, and password resets to ensure secure access.
Class & Assignment Manager	Handles class creation, assignments, student enrollment, deadlines, and submissions.
Rubric Engine	Stores, retrieves, and formats grading rubrics to ensure consistency with teacher-defined criteria.
Grading Controller	Coordinates essay grading, communicates with the LLM Service, retrieves feedback, and stores results.
Notification Service	Sends in-app and email notifications for grade updates, deadlines, and teacher announcements.

Component Diagrams for Subsystems

Frontend Web Application

- Interfaces: REST API (communicates with Backend).
- Responsibilities:
- - Renders dashboards and user interfaces.
- - Handles user interactions (submissions, rubric creation).
- - Displays grades, feedback, and announcements.

Backend Server

- Subsystems:
- - User Management Module: Authentication and password management.
- - Class & Assignment Manager: Class and assignment handling.
- - Rubric Engine: Rubric storage and retrieval.
- - Grading Controller: Essay grading coordination.
- - Notification Service: Sends notifications.
- Interfaces: REST API, LLM Service API, SQL Interface, Email API.

LLM Grading Service

- Interfaces: LLM Service API.
- Responsibilities:
- - Evaluates essays based on rubrics.
- - Provides feedback on grammar, coherence, and content.
- - Returns feedback and scores to the Backend.

Database

- Interfaces: SQL Interface.
- Responsibilities:
- - Stores and retrieves user data, rubrics, grades, and announcements.
- - Ensures data security and integrity.

Email Service

- Interfaces: Email API.
- Responsibilities:
- - Sends password resets and notifications.
- - Uses SMTP with TLS encryption for secure delivery.

Plagiarism Service

- Interfaces: Plagiarism API.
- Responsibilities:
- - Validates essays for plagiarism.
- - Ensures academic integrity compliance.

3.3 Hardware/Software Mapping

This section details the assignment of subsystems to hardware and software components, considering scalability and security aspects.

Nodes & Artifacts

Node	Hardware/Software	Artifact	Subsystems Hosted
Web Server	Apache/Nginx	frontend.war	Frontend Web Application
Application Server	Flask	backend.jar	Backend Server (all subsystems)
LLM Server	Python + CUDA	llm_service.pex	LLM Grading Service
Database Server	MySQL Server	mysql_db	Database

Plagiarism Server	Custom validation tool	plagiarism_check	Plagiarism Service
Email Server	SMTP service	smtp_service	Email Service
Security Gateway	Firewall, TLS 1.3	N/A	Manages secure communication

Connections & Protocols

From Node	To Node	Protocol	Purpose
Web Server	Application Server	HTTPS	Frontend-Backend communication.
Application Server	LLM Server	HTTPS	Send essays for AI grading (UC-7).
Application Server	Database Server	SQL	Store/retrieve data (UC-5, UC-10).
Application Server	Email Server	SMTP+TLS	Send password resets (UC-11).
Application Server	Plagiarism Server	HTTPS	Validate essays (UC-6).
Client Devices	Security Gateway	HTTPS	Secure user access (teachers/students).
Security Gateway	Web Server	HTTPS	Encrypted traffic routing.