**CHAPTER THREE**

**SYSTEM ANALYSIS AND DESIGN**

**3.1 Introduction**

This chapter provides a comprehensive analysis of the system analysis and design for the Online Assignment Submission Platform (OASP). It identifies the weaknesses and limitations of existing systems, outlines the necessary requirements for the proposed platform, and elaborates on its functional specifications. Additionally, it includes use case diagrams along with their explanations, an Entity-Relationship (E-R) diagram, and a detailed walkthrough of the proposed system’s implementation steps. The chapter concludes with an extensive discussion on the platform's system design.

**3.2 Analysis of the Existing System**

Assignment submission systems in educational institutions have evolved significantly, addressing challenges while introducing new complexities. Early email-based systems provided basic convenience but lacked organization, tracking, and verification features. Learning Management Systems (LMS) like Moodle and Blackboard improved submission workflows and organization but lacked specialized tools for complex assignment management.

Dedicated submission platforms introduced file management and deadline tracking, which simplified basic submissions but offered limited support for detailed feedback and plagiarism detection. Cloud-based platforms enhanced accessibility, storage, and backup capabilities, but concerns arose regarding data security, internet dependency, and compliance with institutional data policies.

Modern systems incorporating plagiarism detection tools revolutionized academic integrity, reducing manual workloads. However, issues such as false positives, difficulty handling non-text submissions, high licensing costs, and integration challenges persisted. AI-powered grading tools expedited assessments but struggled with subjective evaluation, bias, and discipline-specific grading standards. Real-time collaboration features improved interactions and peer reviews but introduced complexity in managing multiple feedback channels.

While existing systems have made progress, many still face interface design issues that affect both lecturers and students. For lecturers, overly complicated interfaces and limited customization options make creating assignments, grading, and giving feedback more time-consuming than necessary. Students often struggle with cluttered dashboards, making it hard to find assignments, track deadlines, or view feedback. Additionally, inconsistent navigation, poorly optimized mobile views, and unclear or outdated icons further frustrate users. A more intuitive, accessible, and user-friendly design is essential for better engagement and efficiency.

**3.2.1 Existing System Input**

The current online assignment submission platforms rely on the following inputs:

**i. Assignment Details**: Title, description, deadlines, and instructions provided by instructors.

**ii. User Data**: Students' and instructors' login credentials and personal details.

**iii. Submissions**: Uploaded files in supported formats (e.g., PDF, Word, and images).

**iv. Feedback Data**: Instructors' comments, grades, and suggestions for improvement.

**3.2.2 Existing System Procedure**

Existing systems follow a semi-automated process to handle assignments, as outlined below:

**i. Assignment Creation**: Instructors manually enter assignment details and upload them to the platform.

**ii. Submission Process**: Students submit assignments through file uploads, often without confirmation mechanisms.

**iii. Grading and Feedback**: Instructors download submissions, manually grade them, and provide feedback either as a separate document or within the platform.

**iv. Record Management**: Submissions and grades are stored in simple databases, often lacking advanced analytics or reporting capabilities.

**3.2.3 Existing System Output**

The outputs generated by existing systems are limited and often static. These include:

**i. Submission Logs**: Basic lists of assignments submitted by students.

**ii. Grades**: Numeric or letter grades without detailed breakdowns.

**iii. Feedback Files**: Static documents containing instructor comments.

**iv. Reports**: Manually compiled summaries of grades or submission statuses.

**3.2.4 Problems Associated with the Existing System**

Current systems face the following challenges:

**i. Limited Automation**: Manual processes for grading and feedback increase workload for instructors.

**ii. Scalability Issues**: Systems often experience performance lags during peak usage.

**iii. Integration Gaps**: Many platforms fail to integrate with plagiarism detection tools or external LMS.

**iv. User Accessibility**: Non-intuitive designs and poor mobile optimization hinder usability.

**v. Data Security Risks**: Weak encryption and authentication mechanisms expose user data to breaches.

**vi. Limited Reporting**: Lack of advanced analytics or performance tracking reduces insights into student progress.

**3.3 Propose System Design**

The proposed system design for the Online Assignment Submission Platform (OASP) seeks to overcome the limitations of existing systems by integrating cutting-edge features, intuitive interfaces, and automated processes. The platform is built to streamline assignment management, providing a faster, scalable, and more accessible experience for both students and instructors. To enhance responsiveness, the system will adopt lightweight frameworks optimized for performance, ensuring fast load times and smooth navigation even under high-traffic conditions. Improved icons are introduced, replacing outdated designs with modern, visually appealing alternatives that enhance usability and make the interface more engaging and intuitive. The system also prioritizes advanced plagiarism detection, incorporating enhanced APIs that support a broader range of file types and improve accuracy in identifying copied content while minimizing false positives. The feedback mechanism will also be upgraded, allowing instructors to provide detailed comments, and rubric-based grading directly within the platform.

**3.3.1 Propose System Input**

The proposed system will accept the following inputs to support its advanced functionalities and ensure seamless operation:

i. User Information: Detailed registration data for students, instructors, and administrators, including login credentials and personal details.

ii. Assignment Details: Comprehensive information provided by instructors, such as assignment titles, rubrics, instructions, deadlines, and supporting materials.

iii. Submission Files: Assignments uploaded by students in supported formats (e.g., PDFs, Word documents, or multimedia files).

iv. Feedback Data: Instructors' comments, rubric-based grades (i.e grades assigned using a predefined scoring guide), inline annotations, and suggestions for student improvement.

v. Plagiarism Reports: Data generated from integrated plagiarism detection tools to ensure academic integrity.

vi. System Customization Preferences: User inputs to personalize dashboard layouts, interface themes, and navigation styles for an optimized user experience.

vii. Notifications and Alerts: User-configured preferences for receiving updates on assignment deadlines, submission confirmations, feedback availability, and system reminders.

viii. Communication Data: Inputs for instructor-student communication, including messages, discussion threads, and peer-review feedback.

ix. Course Information: Data on course structures, schedules, enrolled students, and linked assignments to enable contextual assignment management.

x. Usage Analytics Data: Input from system usage tracking to generate insights on submission trends, user activity, and system performance for instructors and administrators.

**3.3.2 Propose System Procedure**

The proposed Online Assignment Submission Platform (OASP) will follow an efficient and streamlined workflow designed to enhance user experience, optimize system performance, and improve assignment management processes:

i. User Registration: Students, instructors, and administrators register with role-based access, allowing customized dashboards and tailored user experiences.

ii. Assignment Creation: Instructors create assignments by adding titles, instructions, deadlines, and grading criteria using a simple interface with clear, modern icons.

iii. Submission Management: Students upload assignments through a responsive portal, receiving automatic submission confirmations and supporting various file formats.

iv. Plagiarism Detection: Submissions are automatically scanned using advanced tools for originality, generating accurate reports and reducing false positives.

v. Grading and Feedback: Instructors evaluate submissions with tools for inline comments, structured grading criteria, and detailed feedback to enhance learning.

vi. Report Generation: The system provides dynamic reports on submissions, grades, and performance trends through intuitive dashboards.

vii. Communication and Notifications: Users receive real-time alerts for deadlines, submissions, and feedback, with an integrated messaging feature for direct communication.

viii. System Optimization: The platform uses lightweight frameworks for fast load times and seamless navigation, ensuring high performance during peak usage.

**3.3.3 Propose System Output**

The proposed Online Assignment Submission Platform (OASP) will provide the following outputs, designed to be clear, helpful, and easy to use:

i. Submission Logs: A real-time list of assignment submissions showing status updates like pending, completed, or late, along with timestamps, making it easy for both students and instructors to track progress.

ii. Plagiarism Reports: Detailed originality reports for each submission, clearly highlighting flagged content and providing useful metrics to ensure academic integrity.

iii. Grades and Feedback: Structured grading outputs that include scores, detailed feedback, and inline comments, all aimed at helping students improve their performance.

iv. Performance Analytics: User-friendly dashboards displaying insights into student progress, submission trends, and overall performance, tailored for instructors and administrators.

v. Customized Notifications: Timely alerts for deadlines, submission confirmations, and feedback availability, personalized for each user to keep them informed.

vi. Dynamic Assignment Status: Visual status updates for assignments (e.g., upcoming, submitted, graded), displayed using clean and intuitive icons for a quick overview.

vii. Course-Level Reports: Summaries of assignment performance, student engagement, and grading trends, offering administrators valuable insights to make informed decisions.

viii. Communication Logs: Easy-to-access records of messages and discussions between students and instructors, promoting clear communication and accountability.

**3.4.1 Requirement Analysis of Proposed System**

It includes functional, non-functional.

**3.4.2 Functional Requirements**

The functional requirements of the proposed Online Assignment Submission Platform (OASP) outline the essential features to ensure smooth operation and user satisfaction:

**User Registration and Login:**

Students, instructors, and administrators must be able to create accounts and log in securely.

**Assignment Management**:

Instructors can create assignments with titles, instructions, deadlines, and grading rubrics.

Students can view available assignments and submit their work in multiple file formats.

**Submission Confirmation:**

Students receive instant confirmation upon successful submission.

**Plagiarism Detection:**

The system scans all submissions for originality and generates detailed plagiarism reports.

**Grading and Feedback:**

Instructors provide feedback using inline comments, rubrics, and overall remarks directly on the platform.

**Notification System:**

Alerts are sent for assignment deadlines, submission confirmations, and feedback availability.

**Dynamic Reports:**

The system generates reports on submission statuses, student progress, and overall assignment performance.

**3.4.3 Non- Functional Requirements**

The non-functional requirements include the software and hardware specifications needed for implementing the proposed system:

**Software Requirements:**

**Operating System:**

The system should run on a Linux or Windows server for deployment.

Programming Languages and Frameworks:

Backend: Python (Django/Flask) or PHP (Laravel).

Frontend: HTML5, CSS3, JavaScript, and libraries like React or Vue.js.

**Database Management System:**

MySQL, PostgreSQL, or MongoDB for managing system data.

Web Server:

Apache or Nginx for hosting the application.

**Plagiarism Detection Tool:**

Integration with tools like Turnitin or open-source solutions for content originality checks.

**Other Software:**

Email API for notification and communication services.

**Hardware Requirements:**

**Server Hardware:**

Minimum 8 GB RAM, Quad-Core Processor, 500 GB SSD for storage.

Scalability for cloud-based deployment (AWS, Azure, or Google Cloud).

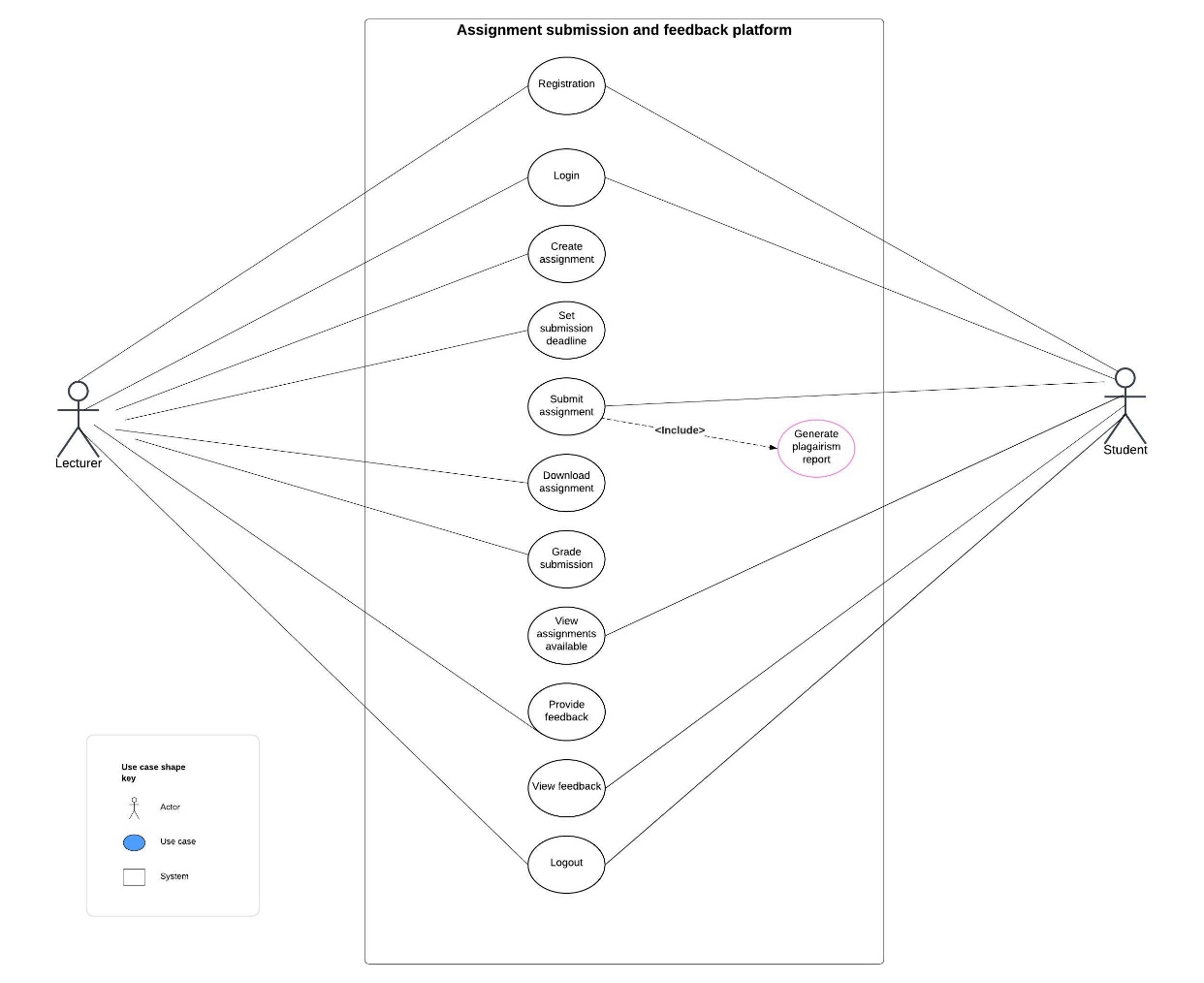
**Client Devices:**

Students and instructors can access the system using desktops, laptops, or mobile devices with at least 2 GB RAM and modern browsers.

**Network Infrastructure:**

High-speed internet connection with at least 10 Mbps bandwidth to ensure seamless file uploads and downloads.

**3.5 Use Case Diagram**

****

The Use Case Diagram highlights user interactions with the Assignment Submission and Feedback Platform, focusing on:

**i. Lecturer Actions:**

Registration: Sign up on the platform to access its functionalities.

Login: Access the system using their credentials.

Create Assignment: Design and upload assignments with detailed instructions.

Set Submission Deadline: Specify due dates for assignment submissions.

Download Assignment: Retrieve submitted assignments for review.

Grade Submission: Evaluate and assign grades to student submissions.

Provide Feedback: Offer detailed feedback on student assignments.

Logout: Exit the platform securely.

**ii. Student Actions:**

Registration: Register on the platform to participate in assignments.

Login: Log in to the platform to access their dashboard.

Submit Assignment: Upload completed assignments for evaluation.

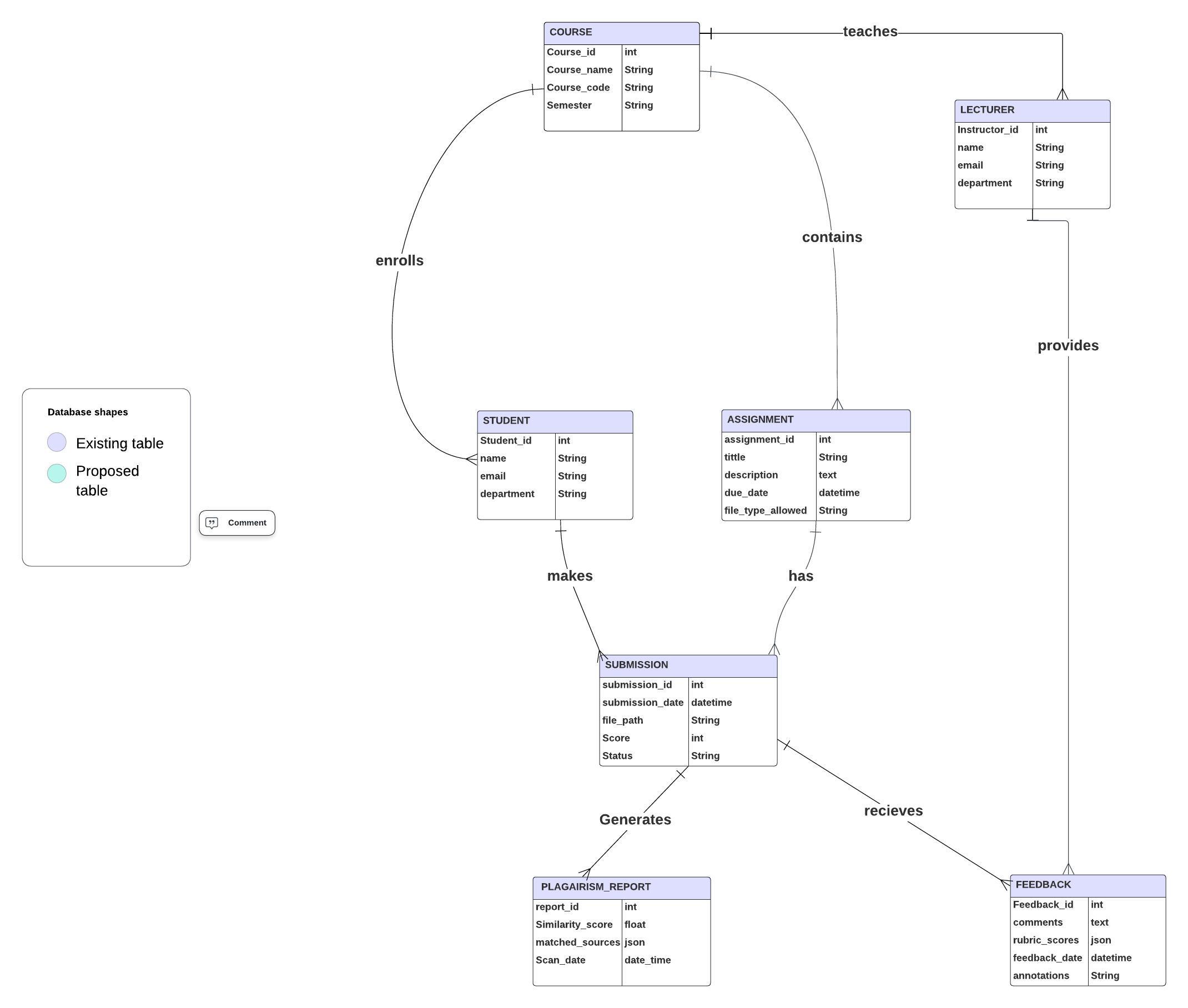
Generate Plagiarism Report: Access an originality report for their submission (if enabled).

View Assignments Available: Check for assignments uploaded by lecturers.

View Feedback: Access feedback provided by lecturers.

Logout: Log out of the system securely after use.

**3.6 E-R Diagram**

****

The E-R Diagram illustrates the relationships among key entities within the system:

i. Course Enrolls Student: Links students to courses they register for, enabling assignment participation.

ii. Lecturer Teaches Course: Connects lecturers to the courses they manage and instruct.

iii. Course Contains Assignment: Associates assignments with their respective courses for proper organization.

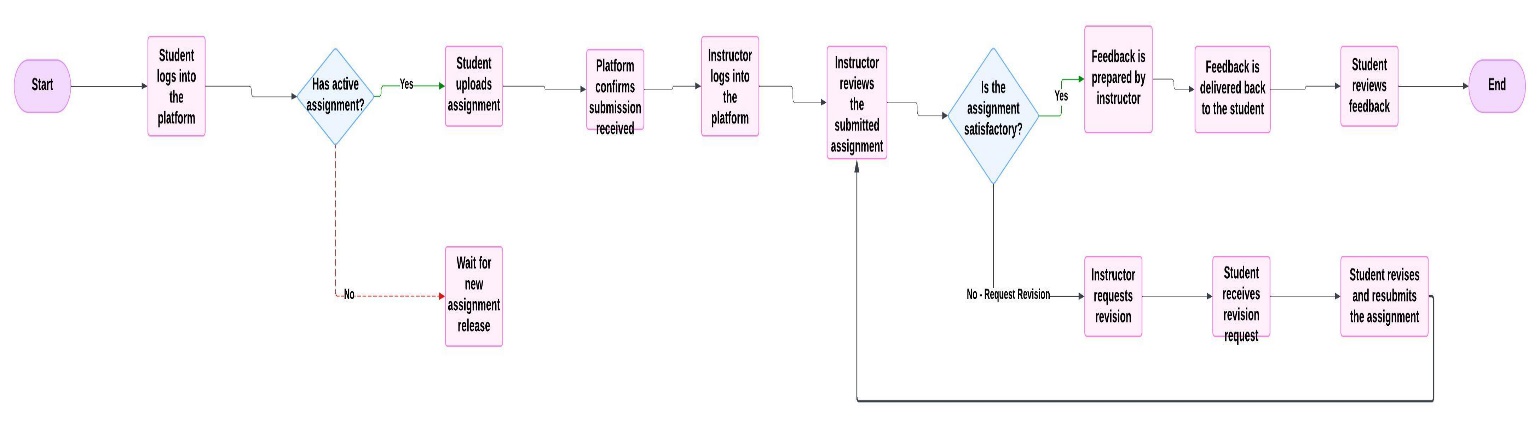
iv. Student Makes Submission: Tracks student assignment submissions for evaluation.

v. Submission Generates Plagiarism Report: Ensures originality by generating a report for each submission.

vi. Lecturer Provides Feedback: Allows lecturers to give detailed evaluations for submitted assignments.

vii. Submission Receives Feedback: Ensures students receive insights on their performance and progress.

**3.3.7 Activity diagram**

****

The activity diagram outlines the steps in the assignment submission and feedback process:

1. Student logs into the platform and checks for active assignments.
2. If an assignment is active, the student uploads the assignment, and the platformconfirmsreceipt.
3. The assignment undergoes plagiarism detection, ensuring originality.
4. The instructor reviews the submission to assess its quality.
   * If satisfactory, feedback is prepared and sent to the student.
   * If revisions are needed, the instructor requests changes, and the student resubmits after revisions.
5. The process ends with the student reviewing the instructor's feedback.