**DEVELOPMENT OF AN INTELLIGENT PLAGIARISM AND GRADING SYSTEM USING DEEP LEARNING TECHNIQUES**

**CHAPTER THREE**

**SYSTEM ANALYSIS AND DESIGN**

**3.1 PREAMBLE**

This section details the analysis and design of the proposed system. It presents the requirements for the system, the techniques taken to achieve the system as well as diagrams to model the proposed system.

**3.2 THE PROPOSED SYSTEM**

The proposed system will be a web-based plagiarism and grading system application running on Amazon Web Services (AWS). The application will allow users to Sign in, upload their documents to check for plagiarism or grade assignments. The student can download the detailed feedback of their plagiarism score or their assignment grade after checking is completed. The user can also check for plagiarism between two documents.

**3.3 REQUIREMENTS ANALYSIS**

Requirements analysis refers to the process of gathering, analysing, documenting and refining features or requirements for a software. It is also known as requirements engineering. Software requirements refer to the descriptions of functionalities provided by the software system as well as the constraints under which it operates. The requirements for this study are detailed in this section.

**3.3.1 Functional Requirements**

This refers to the core functionality or services to be provided by the proposed software. These requirements are as follows:

1. Students will be able to Sign up and Login to their accounts.
2. Students will be able to upload their documents for plagiarism check and Grading.
3. Students will be able to see history of their past submissions.
4. Educators will be able to upload assignments.
5. The system should be able to check for plagiarism accurately.
6. The system should be able to grade essays effectively.

**3.3.2 Non-Functional Requirements**

This refers to the descriptions of the system’s operational capabilities and constraints. They are as follows:

1. The system should be able to provide detailed feedback on document uploaded.
2. The system should have a simple and intuitive interface for end-users.
3. The system should have quick refresh rate.

**3.4 DATA COLLECTION**

The PAN 2011 dataset was used for training the LSTM and CNN plagiarism model. Hewlett Foundation dataset was used in training the Grading system model and the twitter dataset for sentiment analysis.

**3.4.1 Description of** **the PAN 2011 Dataset**

The PAN plagiarism corpus 2011 (PAN-PC-11) is a corpus for the evaluation of automatic plagiarism detection algorithms. The PAN-PC-11 contains documents in which plagiarism has been inserted automatically as well as documents in which plagiarism has been inserted manually. The dataset includes detailed annotations that specify the type of plagiarism and the source of the plagiarized content for each plagiarized section. The dataset provides metadata for each document, such as the document's title, author, publication date. The Pan PC 2011 dataset has been widely used in the evaluation and development of plagiarism detection systems, providing a standardized benchmark for comparing the performance of different approaches. The dataset consists of 11000 suspicious folders and 11000 source folders.

**3.4.2 Description of the Hewlett Dataset**

The dataset was gotten from the William and Flora Hewlett Foundation which was provided for Automated Student Assessment Prize Competition on Kaggle. It consists of 8 sets of essays written by students belonging from 7th to 10th grade. All the essays from these 8 sets are generated from a single prompt. The dataset consists of 12000 essays. The dataset provides metadata for each document, such as essay Id, essay set, essay, and scores of the two raters. Each essay in the dataset is approximately 150 to 550 words in length.

**3.4.3 Description of the Twitter Dataset**

The dataset has three sentiments namely, negative, neutral, and positive. It contains two fields for the tweet and label. It is a widely used dataset in the field of natural language processing and sentiment analysis, as it consists of a large dataset of real-world data. The dataset will be used to train and evaluate deep learning models for classifying the sentiment of essays.

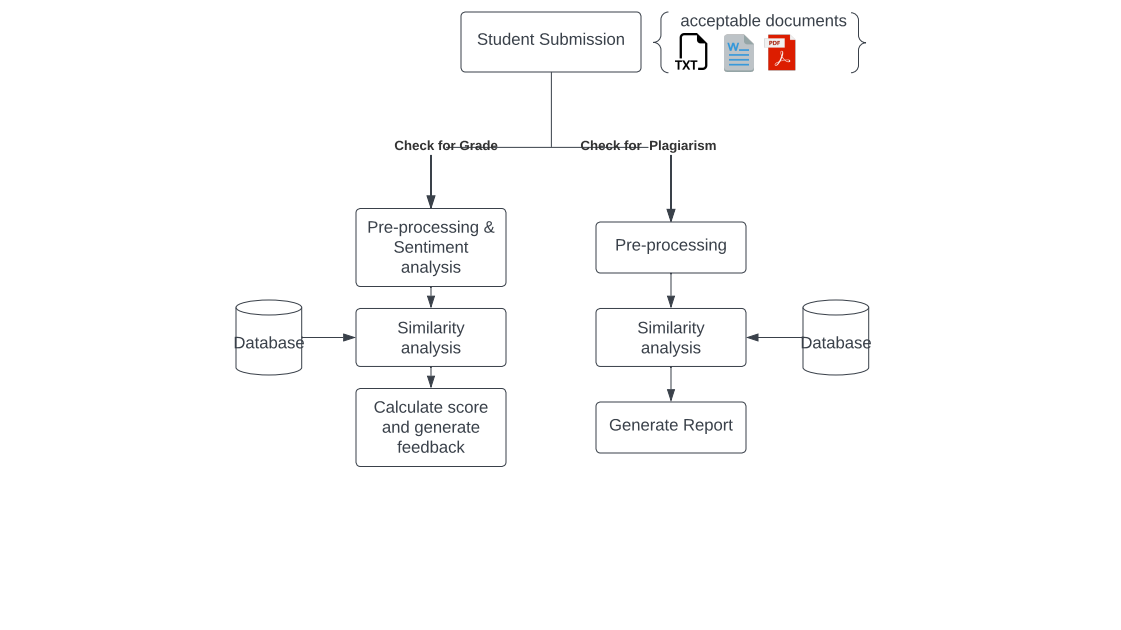
**3.5 PHYSICAL DESIGN**

Physical design includes the input and output mechanisms of the software. It has two major categories namely the input design and the output design. The student uploads the document for plagiarism check or grading the system then outputs the feedback.

**3.5.1 The Proposed System Architecture**

The proposed system architecture for the plagiarism and grading system involves several components including the

1. User interface- this is the interface the student interacts with for making submissions, downloading the feedback, and checking the history of previous submissions result.
2. Database- for storing documents.
3. Similarity analysis- using our trained LSTM and CNN model for checking the similarity between documents.
4. Sentiment analysis- for determining the tone of the essay.
5. Preprocessing: removal of stop words, normalization, lemmatization.

****

**Figure 3.1: Architecture of the Proposed System**

In this architecture, a student can interact with the system using the user interface to upload documents for checking for plagiarism and grading their essays.

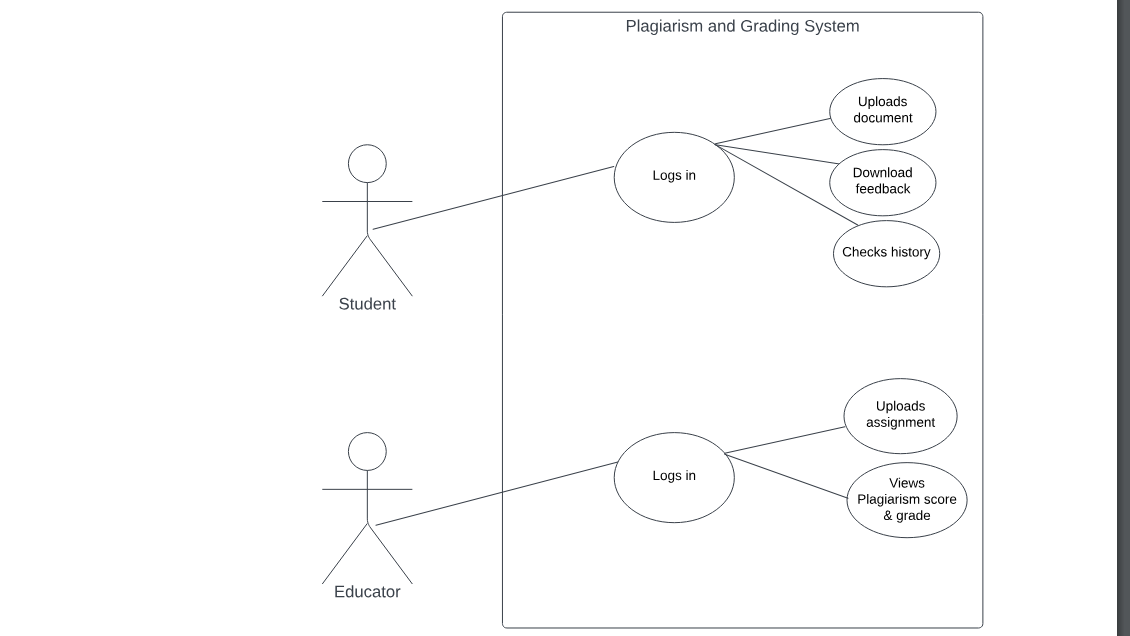
**3.6 LOGICAL DESIGN**

For the proposed system, the goal of logical design is to provide an abstract representation of the software system and to define the system's organization, functionality, and behaviour.

**3.6.1 Use Case Diagram**

A use case diagram shows how various types of users interacts with the system with various use cases. It is represented by using an actor and circle shapes.

The use-case diagram shows that the student can login to the system, upload documents, download feedback, and check history of past submissions. The educator can also login, upload assignments and view students’ plagiarism score and grade.

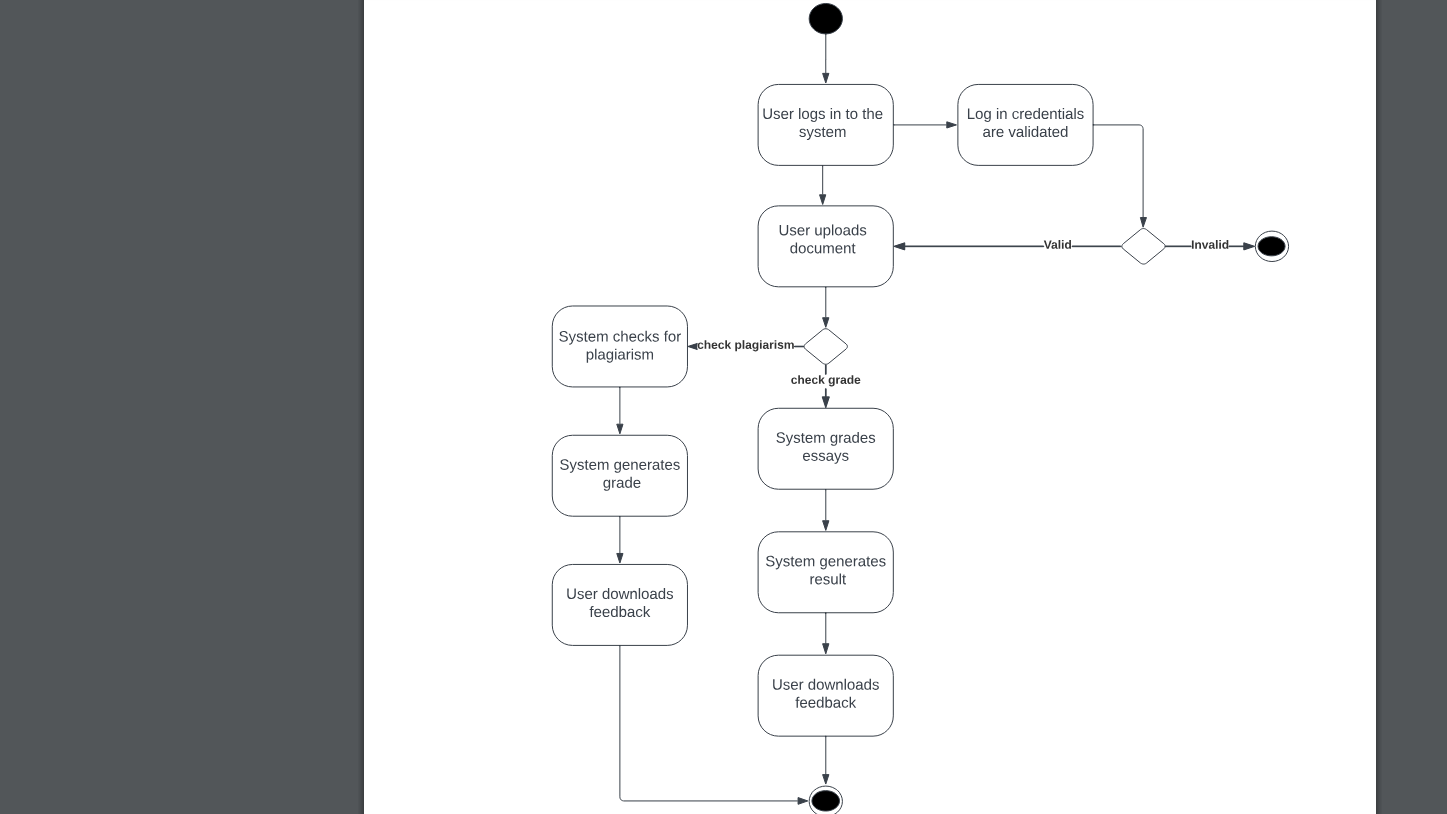


**Figure 3.2: Use case diagram of the Proposed System**

**3.6.2 Activity Diagram**

An activity diagram represents a series of actions or flow of control in a system. It is similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modelling. They also describe the steps in a use case diagram.

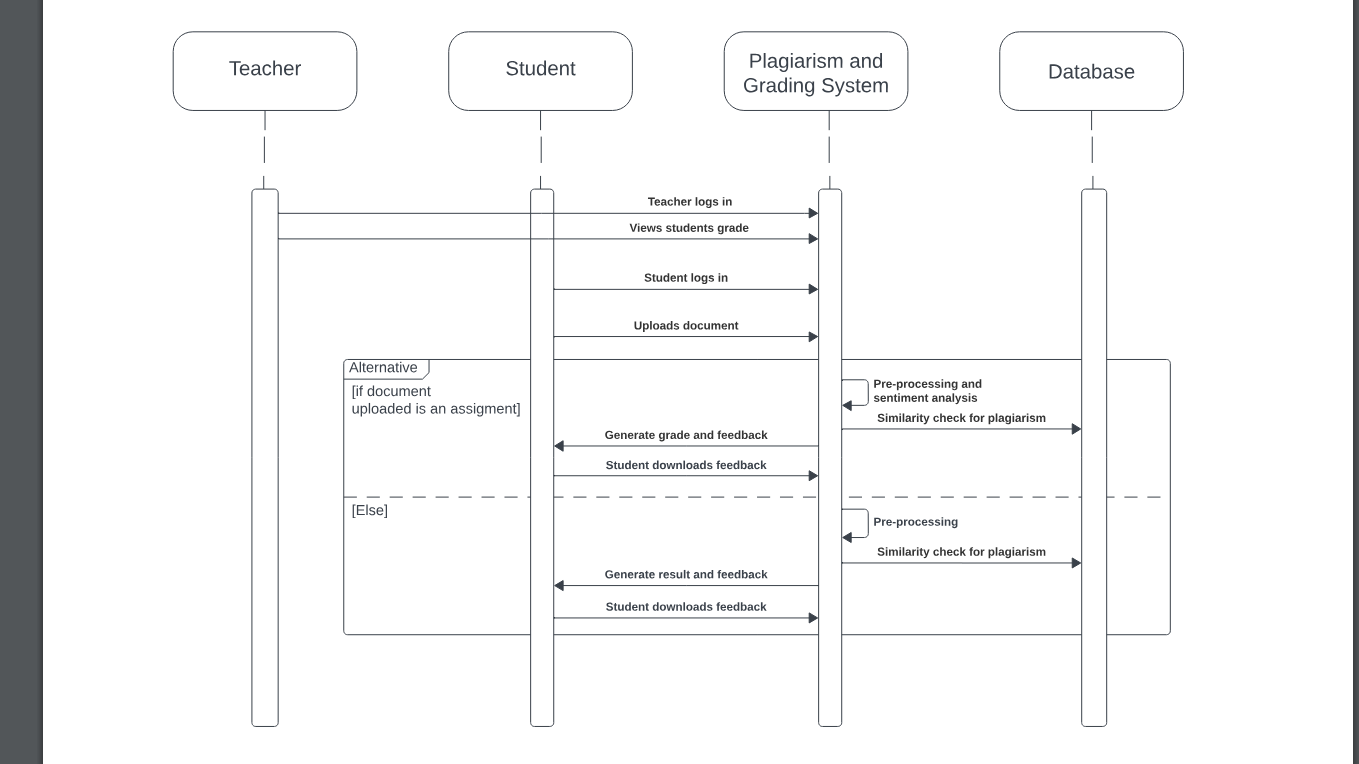
Figure 3.3 shows the activity diagram for the plagiarism and grading system.



**Figure 3.3: Activity diagram of the Proposed System**

**3.6.3 Sequence Diagram**

A sequence diagram is a type of interaction diagram in UML (Unified Modelling Language) that illustrates how objects interact in a particular scenario of a system or software application. It represents the sequence of messages exchanged between objects or components within a specific time frame to accomplish a task or use case.



**Figure 3.4: Activity diagram of the Proposed System**

**3.6 CONCEPTUAL DESIGN**

This section shows the Entity Relationship (ER) diagram of the proposed system as well as the database tables. The Entity Relationship diagram is shown in Figure 3.5.

**A diagram of a computer program

Description automatically generated with medium confidence**

**Figure 3.5: Entity Relationship diagram of the Proposed System**

The normalised table representation are as follows:

1. Table: Student

This table stores student information, including their name, email, and password.

The student can upload documents for checking for plagiarism and grading.

Normalized Representation: Student (id, name, email, password) with student\_id as the primary key.

1. Table: Educator

This table stores educator information, including their name, email, and password.

The educator can upload assignments.

Normalized Representation: Educator (id, name, email, password) with educator\_id as the primary key.

1. Table: Plagiarism

This table checks for plagiarism in documents and provides feedback.

Normalized Representation: Plagiarism (plagiarism\_id, document\_id, similarity, feedback) with id as the primary key and document\_id as the foreign key.

1. Table: Grading

This table grades assignments and provides score and feedback.

Normalized Representation: Grading (grading\_id, document\_id, score, feedback) with grading\_id as the primary key and document\_id as the foreign key.

1. Table: Document

This table represents the document uploaded by the student.

Normalized Representation: Plagiarism (student\_id, document\_id, document\_date, document\_author, document\_content) with document\_id as the primary key and student\_id as the foreign key.