# Project Report

# for

# Automated Assignment Assessment and Evaluation System

**Prepared by - Harsh Bansal**

**B.Tech. Computer Science and Engineering, 2nd Year**

**Amity University Punjab**

**18-02-2023**

# Table of Contents

# 1. Introduction

## 1.1 Overview of the Project

The Automated Assignment Assessment and Evaluation System (AAAES) is a versatile tool that simplifies the often-tedious process of grading student assignments. By leveraging the latest advances in deep learning and natural language processing, the system can accurately and objectively evaluate student work, saving educators valuable time and effort. The system is designed to handle a wide range of Assignment types, from coding projects to written essays. By streamlining the grading process, it empowers teachers to spend more time providing feedback and mentoring their students, rather than spending countless hours on checking codes or grading papers.

## 1.2 Objectives of the Project

- Providing accurate evaluations of student assignments by utilizing Deep Learning and NLP Techniques.
- Providing accurate evaluations of student assignments by utilizing Deep Learning and NLP Techniques.
- Supporting evaluation for multiple types of Assignment types, which include Coding-based Assignments as well as Writing-based.
- Reducing the time spent on grading, thus allowing teachers to focus more on providing a more engaging and effective learning experience to the students.

## 1.3 The Need for the Project

The necessity for an Automated Assignment Assessment and Evaluation System is brought on by several significant difficulties instructors had while using traditional assignment evaluation techniques, which were mostly focused on manually reviewing the assignments. While examining code files, where the same result is reached by several methods for various students, is a laborious and time-consuming operation, it may nevertheless seem doable for written tasks. Assignment evaluation takes up time that a professor could be spending on improving the learning experience

for his or her students, and there is doubt about the reliability of the checks made on these assignments.

## 1.4 Discussion about Existing Systems and related Research Work

- There are multiple Code Quality Systems available on the web, which use a variety of technologies to find vulnerabilities in code like –
    - Snyk (DeepCode)
    - Clever-Commit, Commit Assistant
    - CodeGuru

  All these systems are at the corporate level, but they are unable to give the supplied code a score or other directly comprehensible evaluation statistic. Additionally, they come at a premium cost due to their extensive use. Our goal requires a more effective, less expensive, and less complicated approach that gives us a standard by which to evaluate each submission.
- The evaluation of texts produced by Deep Learning-based Text Generation Models has been the subject of numerous research studies. Rouge and BLEU Scores are two often used measures. Although several publications explore using ML techniques to evaluate essay quality, no reliable system has yet been put into place for assignments in general.

Overall, while there have been advancements in evaluations systems in specific areas, there are not many good implementations of these advancements in assignment assessment and evaluation.

# 2. Feasibility Study

This section analyses the project's technical, financial, and behavioral viability of this project and indicates if the project can be developed or not. Below are the three feasibility standards that have been covered:

## 2.1 Technical Feasibility

The AAAES is a web-based application. The following is an overview of the technologies being used for the Front-end and the Backend of the project:

### 2.1.1 Frontend

The user-interface will be built using:

- **ReactJS** library that primarily uses HTML, CSS, and JavaScript.
- **Custom CSS** Frameworks like Bootstrap, Material-UI might also be used.

### 2.1.2 Backend

The backend / processing-end will be built using:

- **Flask** web-framework built over Python to create API endpoints and connect the frontend to the Evaluation models
- **NLTK** and Related Python Packages from PyPi.
- **MongoDB's** Python library PyMongo to store the assignment records and evaluation results.

All the frameworks, libraries and packages used in the project are open-source, community-run projects i.e., they can be used without any cost attached, and required technical skills are manageable. To deploy the project initially, a free web-hosting server will be sufficient for a small-sized assignment assessment, but a proper implementation would require a high-performance server for multi-user setup.

## 2.2 Economic Feasibility

In this section, potential cost-effectiveness and profitability of this project has been assessed through three categories:

### 2.2.1 Development and Testing Costs

Since the AAAES will be created locally on the developer's computer, there won't be any development or testing expenses associated with the project. Additionally, Google Colab will be used to evaluate the backend procedures for calculating assignment evaluation metrics.

### 2.2.2 Operational Costs

As AAAES is a web-based application, it would require a webserver for hosting, and a domain name for making it accessible to people. Hence the system will have an associated hosting cost. Due to a Deep Learning based back-end, the server requirements are expected to be higher than usual but will be initially managed within Free-tier limits.

### 2.2.3 Revenue Generation

The system will be released as a Utility Application for Universities and Schools. The system will be licensed to universities under enterprise licensing according to the number of professors utilizing the software and the number of assignments being processed in a certain timeframe. The cost of using the software will be a recurring charge through a subscription-based model.

## 2.3 Behavioral Feasibility

### 2.3.1 Purpose-driven Project

The project intends to address the significant issue of the time-consuming task of assignment grading, which affects practically every educational institution. The project's goal will encourage its use in virtually every educational setting.

### 2.3.2 Tendency to save Time.

The project attempts to conserve time, which is the most valuable resource we have. This initiative has the potential to revolutionize this industry by saving educators a ton of time as workloads rise in the future.

# 3. Software Requirements Specification

This section specifies the requirements (both functional and non-functional) of the Automated Assignment Assessment and Evaluation System (AAAES).

## 3.1 Functional Requirements

### 3.1.1 Assignment Creation

**Input** – Details of the Assignment - Name, Description, Last Date, Maximum Marks.
**Output** – Assignment created with the given details and unique Assignment ID is given to the creator.

### 3.1.2 Assignment Submission

**Input** – Assignment ID and Assignment File in a supported format (e.g. .pdf, .zip, .txt, .docx, .py, .java, etc.).
**Output** – Confirmation message for successful submission and Submission ID is returned if Assignment ID is correct and submission is before the due date. The submission is stored in the database.

### 3.1.3 Assignment Assessment and Evaluation

**Input** – Submission ID obtained at Step 3.1.2
**Output** – The system uses Deep Learning and Natural Language Processing algorithms to evaluate the submission and assign marks based on the Grading Criteria. The evaluation results are stored in the database against the Submission ID.

### 3.1.4 Result Management

**Input** – Assignment ID or Submission ID
**Output** – The system generates an Excel sheet containing the scores of all submissions if Assignment ID is given, or the evaluation results of a particular submission if Submission ID is given. The results can be filtered by student name, date of submission, and other criteria.

## 3.2 Non-Functional Requirements

### 3.2.1 Accuracy and Performance

The system should ensure consistent and accurate evaluation of assignments, with grades matching those obtained through human inspection. The system should be able to grade one assignment in approximately 10 seconds to maintain an acceptable level of performance.

### 3.2.2 Scalability

The system should be designed to handle increasing demands in the future, ensuring consistent assessment times and reliable service for all users.

### 3.2.3 Usability

The system should be user-friendly, with clear and comprehensive documentation available for educators and students. Reports generated by the system should be easy to read and understand, and available in PDF format.

### 3.2.4 Flexibility

The system should be capable of assessing a wide range of assignment types, including short essays, term papers, and code-based assignments.

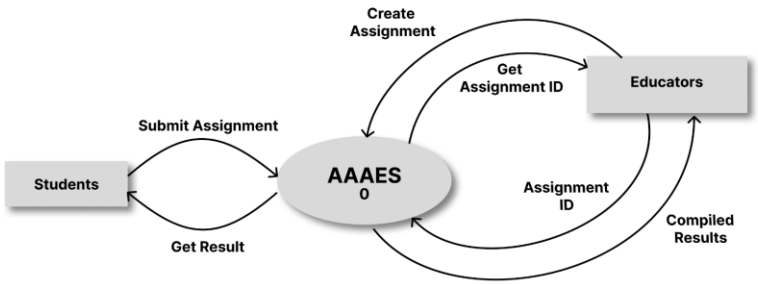
### 3.2.5 Security

The system must implement robust security measures, such as password protection, to prevent unauthorized access to student information and assignments. The system must comply with relevant data protection regulations to ensure student privacy and data security.

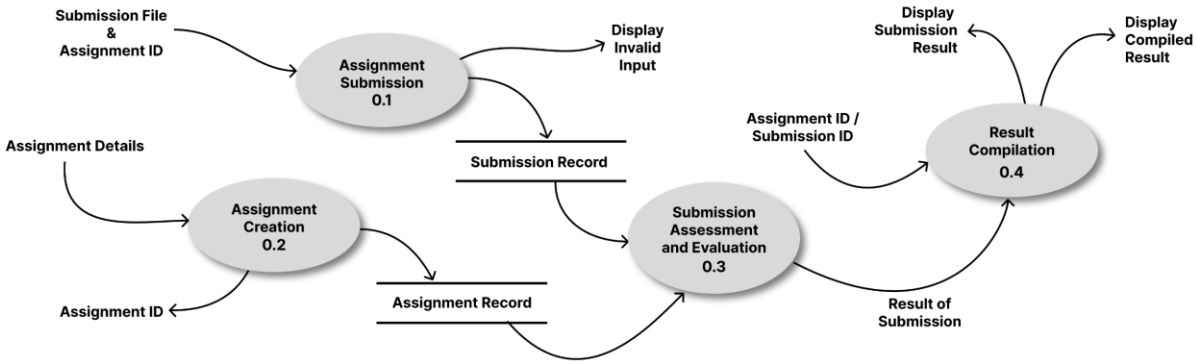
To summarize, this System Requirements Specification (SRS) document provides a comprehensive overview of the functional and non-functional requirements for the development of the system. It serves as a valuable reference for all stakeholders involved in the project and ensures that the final product meets the needs and expectations of the users.

# 4. Design

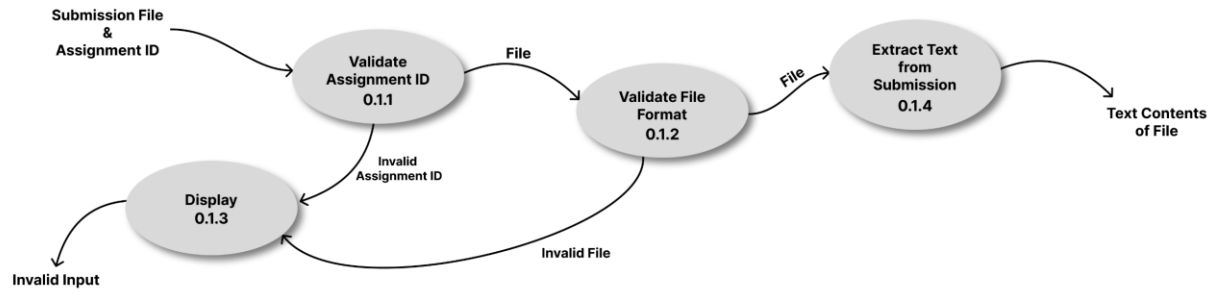
## 4.1 Data Flow Diagram



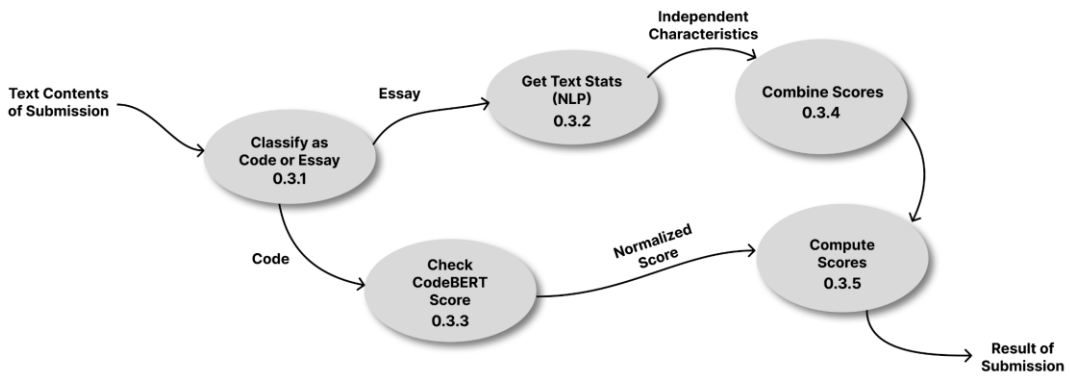

Level 0 - Data-Flow Diagram - AAAES




Level 1 - Data Flow Diagram - AAAES




Level 2 (A) - Assignment Submission - Data Flow Diagram - AAAES




Level 2 (B) - Submission Assessment and Evaluation - Data Flow Diagram - AAAES

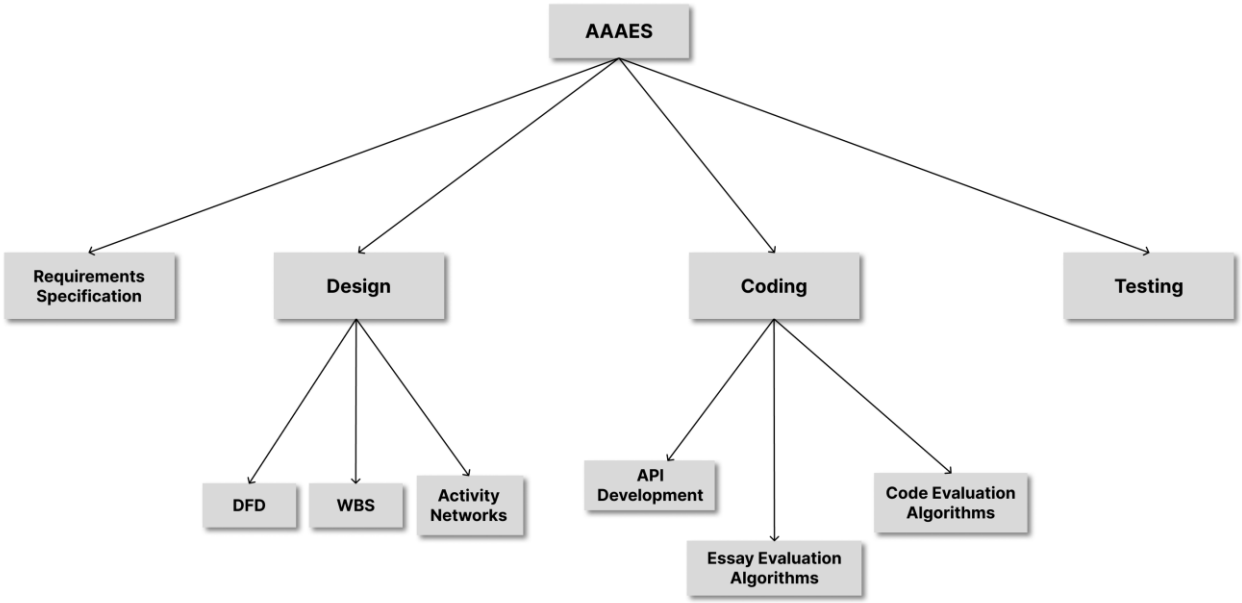
## 4.2 Data Dictionary

Here is the Data Dictionary for the elements of DFD shown in Section 4.1:

| **Data Item** | **Type** | **Description** |
|---|---|---|
| AAAES | abbreviation | Automated Assignment Assessment and Evaluation System |
| Assignment ID | integer | Unique ID for every assignment created |
| Submission ID | integer | Unique ID for every submission done by student |
| Assignment Details | input | name + last date + description + format |
| Assignment Record | data-store | Assignment Details + Assignment Creator |
| Submission File | file | [ pdf file, word file, .cpp file ] |
| Submission Record | data-store | Submission File + Submission ID + Assignment ID |
| Display | string | [“Invalid Assignment ID”, “Invalid Submission ID”, “Format not Supported” ] |
| File | file | {Submission File}* |
| Essay | string | /* contents of Submission File */ |
| Code | string | /* contents of Submission File in-case of Code */ |
| Independent Characteristics | float | Evaluation Metric for Grading of Essay |
| Normalized Score | float | Normalized CodeBERT Score |
| Result of Submission | float | Combined Score after Grading of [Essay, Code] Assignment |

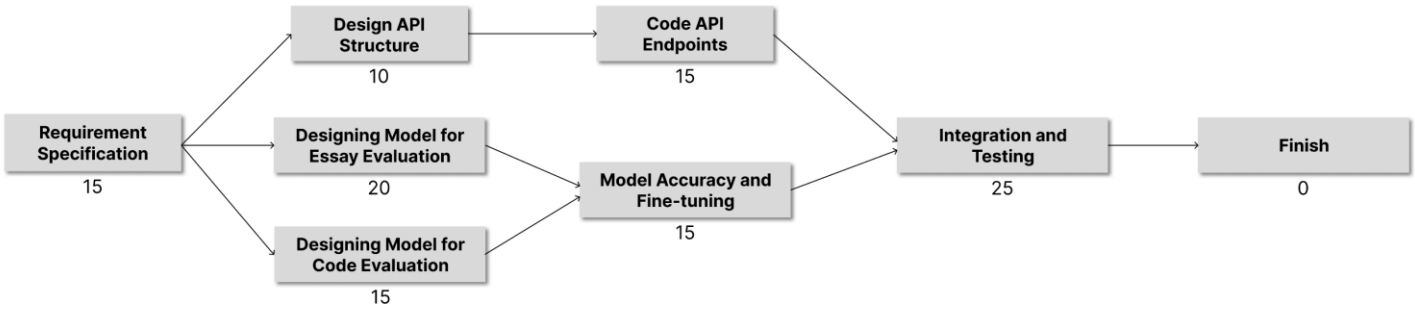
# 5. Project Scheduling

## 5.1 Work Breakdown Structure (WBS)




Work Breakdown Structure (WBS) for AAAES

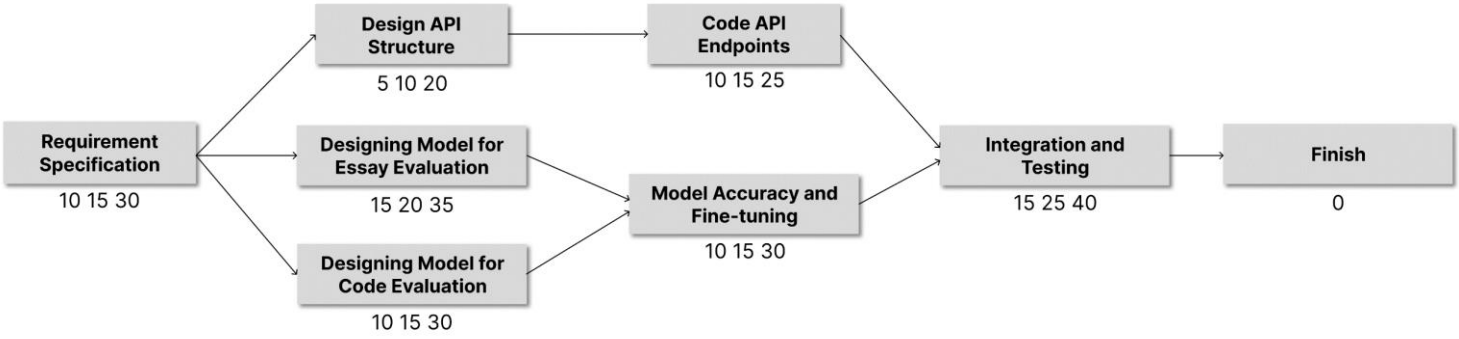
## 5.2 Activity Networks Diagram




Activity Network Diagram (ACN)

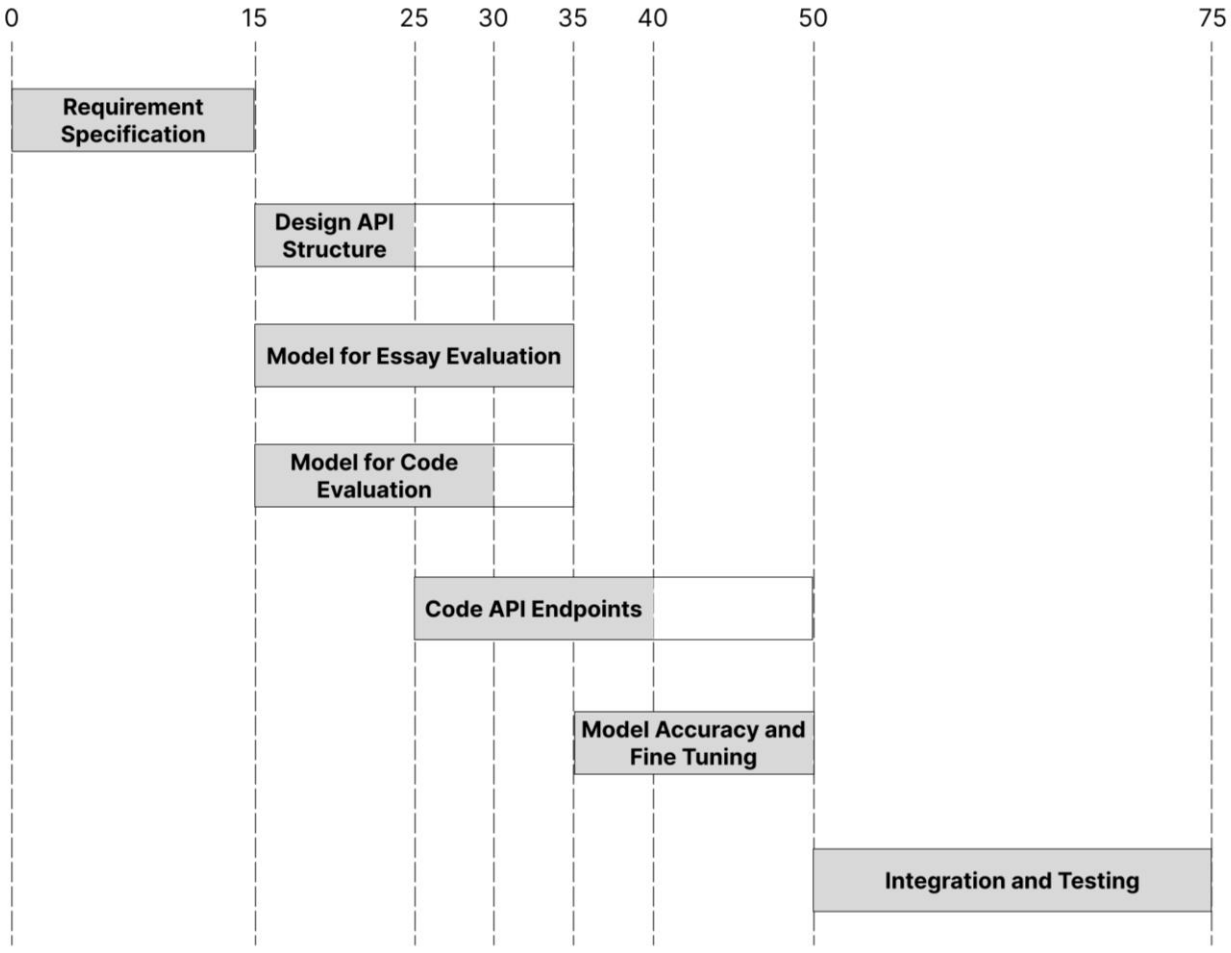
| TASK | ES | EF | LS | LF | ST |
|---|---|---|---|---|---|
| Requirement Specification | 0 | 15 | 0 | 15 | 0 |
| Design API Structure | 15 | 25 | 25 | 35 | 10 |
| Designing Model for Essay Evaluation | 15 | 35 | 15 | 35 | 0 |
| Designing Model for Code Evaluation | 15 | 30 | 20 | 35 | 5 |
| Code API Endpoints | 25 | 40 | 35 | 50 | 10 |
| Model Accuracy and Fine-tuning | 35 | 50 | 35 | 50 | 0 |
| Integration and Testing | 50 | 75 | 50 | 75 | 0 |
| Finish | 75 | 75 | 75 | 75 | 0 |

## 5.3 PERT Chart




PERT Chart for AAAES

## 5.4 Gantt Chart




Gantt Chart Representation of AAAES