

# Software Design Document

## ScribbleInspect - Automated Handwritten Assignment Grading System

**Prepared by**

**Group no: 7**

Ann Mary Saji

Muhammed Farhan M

Sree Lakshmi Chandran

Vishal Vidyanand

# Table of Contents

## Table of Contents

## Revision History

### 1. Introduction

- 1.1 Purpose of this document
- 1.2 Scope of the development project
- 1.3 Definitions, acronyms, and abbreviations
- 1.4 References
- 1.5 Overview of document

### 2. System architecture description

- 2.1 Overview of modules / components
- 2.2 Structure and relationships
- 2.3 User Interface
  - 2.3.1 Login Screen
  - 2.3.2 Dashboard
  - 2.3.3 Assignment List
  - 2.3.4 Assignment Details
  - 2.3.5 Feedback Screen
  - 2.3.6 Reports

### 3. Detailed description of components

- 3.1 Component template description
  - 3.1.1 Handwriting Recognition Module
  - 3.1.2 Image Preprocessing Module
  - 3.1.3 Answer Key Comparison Module
  - 3.1.4 Grading Module
  - 3.1.5 Feedback Module
  - 3.1.6 User Authentication and Administration Module

### 4. Reuse and relationships to other products

### 5. Design decisions and tradeoffs

## Revision History

Version	Name	Reason For Changes	Date

# **1. Introduction**

## **1.1 Purpose of this document**

An Automated Handwritten Assignment Grading System has the purpose of providing a detailed description of its design and architecture, including its different components, the data flow, algorithms, and interfaces that allow it to perform its assigned tasks. This system serves as a guide for the development team, facilitating communication and collaboration among its members and with stakeholders, ensuring that everyone has a common understanding of the system's design and functionality. It also enables quality assurance and testing, ensuring that the system's components and interfaces are tested thoroughly and that the system meets its design goals and objectives. Additionally, the system supports maintenance and future development, providing a basis for consistent updates and changes that align with its design and architecture.

## **1.2 Scope of the development project**

The purpose of this requirements specification is to:

- The development process is to define the system requirements. This includes identifying the functional and non-functional requirements of the system to ensure it supports subjective answer evaluation and handwritten submissions. The system should also be scalable, secure, and able to handle a large volume of assignments.
- The development of the evaluation criteria and algorithms is an essential aspect of the development process. This involves defining the evaluation criteria for subjective answers and developing algorithms that can automate the scoring process. The evaluation criteria should be comprehensive, and the algorithms should be accurate, consistent, and reliable.
- The development of a user-friendly interface, data management system, and testing plan is essential to ensure the system is easy to use and maintain. The interface should allow for the submission of assignments, viewing of results, and reporting. The data management system should handle large volumes of assignments and provide reporting capabilities. A testing plan should ensure accuracy and consistency, while the maintenance plan should provide ongoing functionality and support

## **1.3 Definitions, acronyms, and abbreviations**

- Definitions:

Automated Handwritten Assignment Grading System: a computer software or system that uses image processing and machine learning techniques to automatically grade handwritten assignments.

- Acronyms:
  - AHAGS: Automated Handwritten Assignment Grading System
  - HAGS: Handwritten Assignment Grading System
  - AIGS: Automated Image Grading System
- Abbreviations:
  - HW: Handwritten
  - AI: Artificial Intelligence
  - ML: Machine Learning
  - OCR: Optical Character Recognition

## 1.4 References

- [1] P. Dhande and R. Kharat, "Recognition of cursive English handwritten characters," 2017 International Conference on Trends in Electronics and Informatics (ICEI), Tirunelveli, India, 2017, pp. 199-203, doi: 10.1109/ICOEI.2017.8300915.  
<https://ieeexplore.ieee.org/document/8300915>
- [2] M. F. Bashir, H. Arshad, A. R. Javed, N. Kryvinska and S. S. Band, "Subjective Answers Evaluation Using Machine Learning and Natural Language Processing," in IEEE Access, vol. 9, pp. 158972-158983, 2021, doi: 10.1109/ACCESS.2021.3130902.  
<https://ieeexplore.ieee.org/document/9627669>

## 1.5 Overview of document

An Automated Handwritten Assignment Grading System is a software tool designed to help instructors grade handwritten assignments using computer vision and machine learning algorithms. The system takes scanned or photographed copies of handwritten assignments and analyzes them using image processing techniques to recognize characters, words, and sentences.

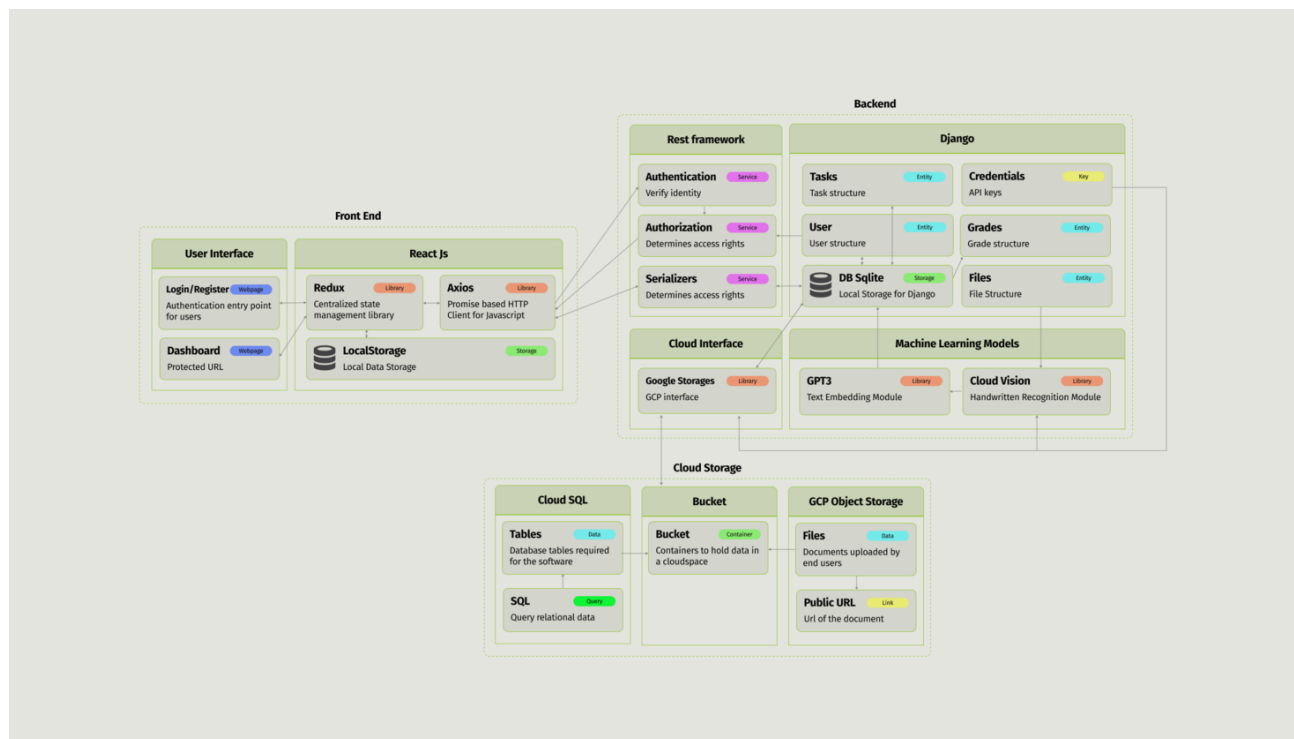
The system then compares the recognized text with a predefined answer key to determine the accuracy of the student's responses. The grading process is typically automated, which helps to reduce the time and effort required for manual grading, while also providing consistent and objective feedback to students.

Some of the key features of an Automated Handwritten Assignment Grading System may include the ability to handle different handwriting styles and languages, as well as the ability to provide detailed feedback on common errors and areas for improvement.

## 2. System architecture description

### 2.1 Overview of modules / components

An Automated Handwritten Assignment Grading System is typically composed of several modules or components. The Image Preprocessing Module is responsible for preparing the scanned or photographed images of handwritten assignments for analysis. The Handwriting Recognition Module uses machine learning algorithms to recognize characters, words, and sentences in the pre-processed images. The Answer Key Comparison Module compares the recognized text with a predefined answer key to determine the accuracy of the student's responses. The Grading Module assigns a grade to the student's assignment based on the accuracy of their responses, while the Feedback Module provides feedback to the student on their assignment, highlighting areas where they performed well and areas where they need to improve. The Reporting Module generates reports for instructors, summarizing the grades and feedback for each student's assignment, and providing insights into common errors or areas where students are struggling. Other features that may be included in an Automated Handwritten Assignment Grading System include user authentication, administration tools, and the ability to handle different handwriting styles and languages.



## 2.2 Structure and relationships

An Automated Handwritten Assignment Grading System is a computerized system that is designed to recognize, process, and grade handwritten assignments automatically. The system consists of several components that work together to achieve this goal. Here's an overview of the structure and relationships of the different components in an Automated Handwritten Assignment Grading System:

**Input Interface:** The input interface is responsible for capturing the handwritten assignments and converting them into a digital format that can be processed by the system. This component can include hardware such as scanners or cameras and software for image processing.

**Preprocessing:** The preprocessing component is responsible for preparing the digital images of the assignments for further processing. This may involve removing noise, skew correction, and normalization.

**Handwriting Recognition:** The handwriting recognition component is responsible for recognizing the handwritten text and converting it into machine-readable form. This component uses optical character recognition (OCR) algorithms and machine learning techniques to achieve accurate recognition.

**Grading Engine:** The grading engine component is responsible for evaluating the machine-readable text and assigning a grade to the assignment. This component uses a set of pre-defined rules and criteria to determine the grade.

**Database:** The database component stores the digital images of the assignments and the corresponding grades assigned by the grading engine. It may also include metadata such as student information, assignment details, and grading criteria.

**User Interface:** The user interface component provides an interface for teachers or instructors to view and manage the graded assignments. It may also include features such as assignment submission, feedback provision, and student progress tracking.

## 2.3 User Interface

### 2.3.1 Login Screen

The first screen a user sees is the login screen. Here, the user will need to enter their login credentials to access the system.

### 2.3.2 Dashboard

Once the user logs in, they will be taken to the dashboard. The dashboard is the main hub of the system and provides an overview of the user's grading activities. It displays information such as the

number of assignments to grade, the status of assignments, and any notifications or messages from the system.

### **2.3.3 Assignment List**

This screen displays a list of all the assignments that need to be graded. The list provides information such as the name of the assignment, the due date, and the number of students who have submitted the assignment.

### **2.3.4 Assignment Details**

When a user clicks on an assignment from the list, they will be taken to the assignment details screen. Here, the user can view the images of the handwritten assignments and the system-generated grades.

### **2.3.5 Feedback Screen**

If the user wants to provide feedback to the student, they can do so on this screen. The feedback can be in the form of text, audio, or video.

### **2.3.6 Reports**

This screen displays reports generated by the system. The reports provide an overview of the grading process, including the number of assignments graded, the average grade, and any patterns or trends observed.

## **3. Detailed description of components**

### **3.1 Component template description**

#### **3.1.1 Handwriting Recognition Module**

The below table outlines the functional diagram

Identification	Component Name: Handwriting Recognition Module Component ID: HRM
----------------	---



Type	The Handwriting Recognition Module is a software module of the Automated Handwritten Assignment Grading System.
Purpose	The purpose of the Handwriting Recognition Module is to recognize the handwritten text from scanned images of handwritten assignments
Function	The Handwriting Recognition Module functions by taking the input of scanned images of handwritten assignments and performing Optical Character Recognition (OCR) to recognize the text written on the paper. The module uses machine learning algorithms to recognize handwritten text
Subordinates	The Handwriting Recognition Module does not have any subordinates
Dependencies	The Handwriting Recognition Module is dependent on the Image Preprocessing Module to receive preprocessed images for OCR. It is also dependent on the machine learning algorithms to recognize the handwriting.
Interface	The Handwriting Recognition Module has interfaces with the Image Preprocessing Module to receive preprocessed images for OCR. It also interfaces with the Grading Module to provide the recognized text for grading
Resource	The Handwriting Recognition Module requires a computing system to run the machine learning algorithms for OCR. It requires access to the scanned images of handwritten assignments and the necessary libraries and tools for machine learning.
Processing	The Handwriting Recognition Module processes the scanned images of handwritten assignments and performs OCR to recognize the text. It uses machine learning algorithms to analyze the images and provide the recognized text.

Data	The Handwriting Recognition Module takes scanned images of handwritten assignments as input and provides the recognized text as output. The recognized text is then used by the Grading Module for grading the assignments. The Handwriting Recognition Module does not store any data beyond the recognized text.
------	--

### 3.1.2 Image Preprocessing Module

Identification	Image Preprocessing Module
Type	Software component
Purpose	Prepare handwritten assignments for recognition by removing noise, enhancing image quality and segmenting individual characters
Function	Process digital images of handwritten assignments using various techniques such as thresholding, filtering, and segmentation to prepare for handwriting recognition
Subordinates	None
Dependencies	Depends on the input image format and quality, and the algorithms used for preprocessing
Interface	Accepts digital images of handwritten assignments as input and outputs preprocessed images for recognition
Resources	Requires a computer or server with appropriate hardware and software resources
Processing	Uses various image processing techniques to enhance image quality and segment individual characters for recognition
Data	Receives digital images of handwritten assignments as input and produces preprocessed images for the handwriting recognition module

### 3.1.3 Answer Key Comparison Module

Identification	Answer Key Comparison Module
Type	Software component
Purpose	Compare recognized characters with the answer key to determine correctness and calculate a grade for the assignment
Function	Compares the characters recognized by the handwriting recognition module with the answer key to determine correctness, and calculates a grade based on the number of correct characters
Subordinates	None
Dependencies	Depends on the accuracy of the handwriting recognition module and the correctness of the answer key.
Interface	Accepts recognized characters and the answer key as input, and outputs a grade for the assignment
Resource	Requires a computer or server with appropriate hardware and software resources
Processing	Compares recognized characters with the answer key to determine correctness, and calculates a grade based on the number of correct characters
Data	Receives recognized characters and the answer key as input, and produces a grade for the assignment.

### 3.1.4 Grading Module

Identification	Grading module, a software component
Type	Functional module in an educational software system
Purpose	Automate grading process, provide consistent feedback, help manage workload
Function	Receives student submissions, applies grading criteria/rules, assigns grades
Subordinates	None specified
Dependencies	Educational software system
Interface	Interface with student submissions, grading criteria/rules, instructor workload management tools

Resource	Computing resources, grading criteria/rules
Processing	Automatic grading using predefined rules and criteria
Data	Student submissions, grades assigned

### 3.1.5 Feedback Module

Identification	Feedback Module, a software component
Type	Functional module in an educational software system
Purpose	Provide feedback to students on their performance and progress
Function	Generates feedback based on student work and progress, presents feedback to students and instructors
Subordinates	None specified
Dependencies	Educational software system, student data
Interface	Interface with student data, instructor tools for reviewing and delivering feedback
Resource	Computing resources, student data
Processing	Analyzes student data and generates feedback based on predefined criteria and rules
Data	Student data, feedback generated

### 3.1.5 Reporting Module

Identification	Reporting Module, a software component
Type	Functional module in an educational software system
Purpose	Generate reports on student performance and progress for instructors and administrators

Function	Collects and aggregates student data, generates reports, presents reports to instructors and administrators
Subordinates	None specified
Dependencies	Educational software system, student data
Interface	Interface with student data, instructor and administrator tools for accessing and reviewing reports
Resource	Computing resources, student data
Processing	Analyzes student data and generates reports based on predefined criteria and rules
Data	Student data, reports generated

### 3.1.6 User Authentication and Administration Module

Identification	User Authentication and Administration Module, a software component
Type	Functional module in an educational software system
Purpose	Authenticate users and manage user accounts
Function	Verifies user identity, manages user accounts and permissions, logs user activity
Subordinates	None specified
Dependencies	Educational software system, user data
Interface	Interface with user data, login and registration interfaces, administrative tools for managing user accounts and permissions
Resource	Computing resources, user data
Processing	Authenticates user credentials, manages user accounts and permissions based on predefined rules and criteria, logs user activity
Data	User data, logs of user activity

## **4. Reuse and relationships to other products**

The Automated Handwritten Assignment Grading System has potential for reuse and integration with other educational software products. One example is integrating the grading system with a Learning Management System (LMS) to automate the grading process for handwritten assignments, providing consistent feedback to students and reducing the workload for instructors. Additionally, the system could be integrated with a plagiarism detection system to automatically check for plagiarism, further enhancing grading accuracy. The Automated Handwritten Assignment Grading System may also be integrated with a Student Information System (SIS) to provide detailed reports on student performance and progress, helping instructors and administrators track outcomes and identify areas of improvement. Dependencies of the system may include OCR software for recognizing handwritten text, image processing software for enhancing image quality, and machine learning libraries for developing grading models.

## **5. Design decisions and tradeoffs**

These include factors such as accuracy, speed, cost, scalability, and more. For example, the choice of handwriting recognition technology can impact the system's accuracy and cost. OCR may be more cost-effective, but less accurate, while machine learning and deep learning may offer higher accuracy but require more computational resources. Similarly, image processing techniques can improve accuracy, but may increase computational costs and slow down the grading process. The balance between accuracy and speed is another important consideration, as a more accurate system may require more time and resources. Scalability is also crucial, as the system needs to be able to handle increasing numbers of assignments and students. Finally, cost is an important factor in the design of the system, as more accurate and faster systems may require more expensive hardware and resources. Balancing these factors is important to create a grading system that meets specific requirements and provides the greatest benefit to its users.