

Software Requirements Specification

ScribbleInspect - Automated Handwritten-Assignment Grading System

Prepared by

Ann Mary Saji
Muhammed Farhan M
Sree Lakshmi Chandran
Vishal Vidyanand

15-02-2023

Table of Contents

Table of Contents

Revision History

1. Introduction

- 1.1 Purpose
- 1.2 Product Scope
- 1.3 Intended Audience and Document Overview
- 1.4 Definitions, Acronyms and Abbreviations
- 1.5 Document Conventions
- 1.6 References and Acknowledgments

2. Overall Description

- 2.1 Product Perspective
- 2.2 Product Functions
- 2.3 User Classes and Characteristics
- 2.4 Operating Environment
- 2.5 Design and Implementation Constraints
- 2.6 Assumptions and Dependencies

3. External Interface Requirements

- 3.1 User Interfaces
- 3.2 Hardware Interfaces
- 3.3 Software Interfaces
- 3.4 Communications Interfaces

4. System Features

- 4.1 Optical Character Recognition (OCR)

4.2 Automated assessment

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.2 Safety Requirements

5.3 Security Requirements

5.4 Software Quality Attributes

5.5 Business Rules

Revision History

Name	Date	Reason For Changes	Version
Ann Mary Saji	01-04-23	Minor changes in functional requirements specification and design constraints.	1.0

1. Introduction

1.1 Purpose

The aim of this project is to develop an advanced method for automatically evaluating descriptive answers using machine learning and natural language processing. This technological approach represents a significant leap forward in assisting the education sector to perform its duties more efficiently and effectively, by reducing the need for manual labor in tasks such as comparing answers with a correct solution.

1.2 Product Scope

Evaluating subjective answers requires the checker to actively score every word, and the evaluator's mental health, fatigue, and objectivity can significantly impact the overall result. Therefore, delegating this tedious and critical task to a system can be a more time and resource-efficient approach to ensure consistency and accuracy in the evaluation process. Given that handwritten answers are still a common method for submitting assignments, there is significant potential for the use of optical character recognition in this context. Unlike most existing systems, we have integrated both of these capabilities into our platform, providing a more comprehensive and sophisticated solution for our users.

1.3 Intended Audience and Document Overview

The readers for this project consist of two groups: the panel of professors who will evaluate the technical and academic merit of the work, and the team members, which includes developers and documentation writers who will review the work for accuracy and completeness. This document provides a comprehensive overview of our software product, including its parameters and goals. It outlines the project's target audience and details the user interface, as well as the hardware and software requirements for optimal performance.

1.4 Definitions, Acronyms and Abbreviations

1. API - Application Programming Interface
2. DB - Database
3. OCR - Optical Character Recognition
4. IDE - Integrated Development Environment

1.5 Document Conventions

- Use Times New Roman font for the entire report – Chapter/Section Title – Times New Roman 23, Bold; Heading 2 – Times New Roman 17, Bold; Heading 3 – Times New Roman 11, Normal; Body- Times New Roman 11, Normal.
- Line Spacing – Between Heading 2 – 3 lines, between lines in paragraph 1.5 lines.
- Alignments – Chapter/Section Title – Center, Heading 2 & 3 should be Left Aligned. Ensure that all body text is paragraph justified.

1.6 References and Acknowledgments

[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>

2. Overall Description

2.1 Product Perspective

Our product aims to redefine the process of grading written assignments, replacing the traditional methods with a modern and efficient approach. The conventional method includes manual assessment and computations that consume more time and effort.

2.2 Product Functions

- Login

Allows students and teachers to access the system by providing their username and password.

Password specifications:

- 1) Minimum length: 8 characters
- 2) Cannot be similar to any other attribute values of user (such as username or email)
- 3) Common passwords like "password" and "hello123" are recognized and rejected
- 4) Cannot contain only numbers.

- Upload pdf

Allows teachers to upload answer keys, and students to upload their assignment answers. This functionality enables the user to select a PDF file from their device and upload it to the system for submission.

- Review pdf

Allows students to view their uploaded PDF files.

- Remove pdf

Allows both teachers and students to delete their previously uploaded PDF files

- Compute grade

Allows teachers to start the automated grading process for student assignments. This option allows for grading of all assignments received for a task in a batch, rather than individually.

- View marks

Allows students to view their marks once the grading process for their assignments is completed.

- Generate report

Allows teachers to create a customized report for a specific student

- Request for review

Allows students to request for reevaluation of their assignments if they are unsatisfied with the marks they received.

- Add or delete task

Allows teachers to create and announce a new assignment to a class. This feature enables the teacher to specify the details of the assignment, such as the deadline, difficulty level, and evaluation criteria.

2.3 User Classes and Characteristics

2.3.1 Teacher

Teacher is the primary user. Teacher has access to the functionalities listed below

- Login
- Upload pdf
- Review pdf
- Remove pdf
- Compute grade
- View marks
- Generate report
- Add or delete task

2.3.2 Student

These are the functionalities available to the student user

- Login
- Upload pdf
- Review pdf
- Remove pdf
- View marks
- Request for review

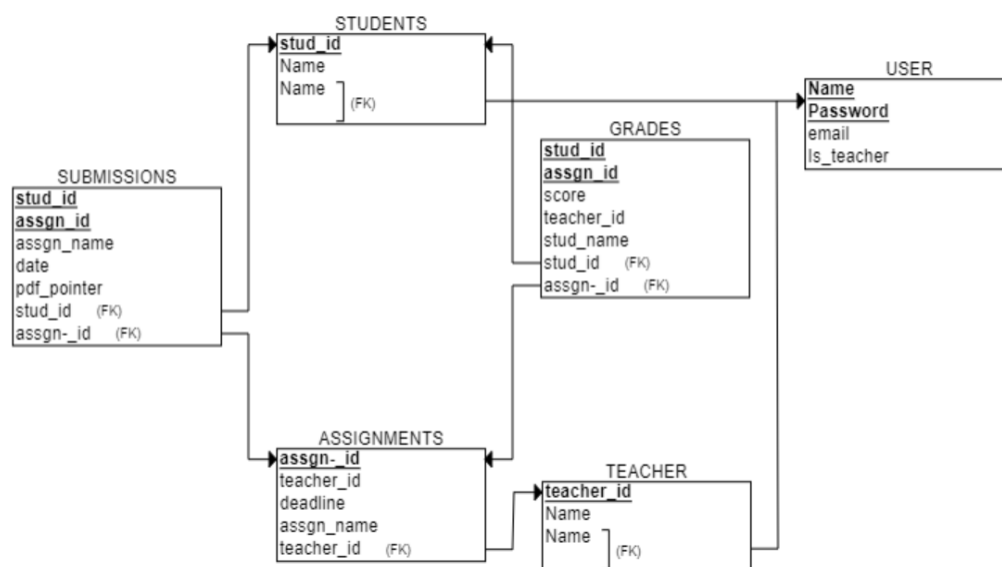
2.4 Operating Environment

To implement the system as a web application, the general operating environment requirements can be summarized as follows:

Web application can be run on any compatible web browser. A database server is needed to store and manage the website's data. The web and database servers can be run on a compatible operating system, such as Linux, Windows Server, or macOS Server. The servers should have sufficient processing power, memory, and storage to handle the expected traffic volume and data storage requirements.

2.5 Design and Implementation Constraints

- Hardware constraints
- Tool specification
 - Google Vision API
- IDE
 - Visual Studio Code
- Framework
- Design convention
- Programming standards
 - Naming convention
 - Project specific convention of module and subroutine headers
 - File organization
 - Specification for error handling
 - Revision and version control
 - Git is used
- Language requirements
 - Python
 - React
 - MongoDB
- Database Schema



2.6 Assumptions and Dependencies

Assumptions:

- Users have basic computer skills and are familiar with browsing the web.
- The website will be primarily accessed on desktop with modern web browsers
- The website will be hosted on a reliable web hosting service with adequate resources to handle the expected traffic.
- The website will be developed using commonly accepted web standards and best practices.
- Users will have a reliable internet connection with sufficient bandwidth to access the website.
- The system should have more capacity and provide fast access to the database.
- The system meets all the hardware requirements.

Dependencies:

- The website's functionality will depend on the features and capabilities of the web browser used by the user.
- The website's performance will depend on the speed and reliability of the internet connection used by the user.
- The website's design and layout may be affected by the screen size and resolution of the device used by the user.
- The website's data storage and retrieval may depend on external data sources or APIs.
- The website's security will depend on the web hosting service's security measures, as well as any security measures implemented in the website's code.

3. External Interface Requirements

3.1 User Interfaces

- Login and authentication: The user interface for logging in and authenticating users, which includes the use of usernames, passwords, and two-factor authentication.
- Scanned images display: The user interface for displaying the uploaded student documents, which can allow for zooming, panning, and scrolling.
- Grading tools: The user interface for tools that allow the evaluator to assign scores, marks, or grades to the student answers based on the grading criteria.
- Rubric display: The user interface for displaying the grading rubric or scoring guide that outlines the criteria and expectations for evaluating the student answers.
- Reports and analytics: The user interface for generating reports and analytics on the performance of individual students, groups of students, or the entire class, such as grade distribution, average scores, or knowledge gaps.
- Export and import: The user interface for exporting or importing data, such as the scanned images of student answers, grading criteria, or reports.
- Help and support: The user interface for accessing help and support resources, such as user manuals, FAQs, video tutorials, or customer support.

The user interfaces in the software should be designed to be intuitive, user-friendly, and efficient, allowing for easy and accurate evaluation of student answers.

3.2 Hardware Interfaces

- Computer system: This includes the hardware components used for running the handwriting recognition software and performing the evaluation. This can include processors, memory, storage, and other components like keyboard, mouse, touch screen and a display.
- Network connection: A network connection is required to upload and download files, access online resources, and communicate with other users or systems.
- Storage devices: A storage device, such as a hard disk or flash drive is required to store and backup the scanned images of student answers, grading reports, or other data.

The software should be designed to be compatible with a range of hardware configurations, accessible to users with different levels of technical proficiency and should provide the desired output for a user's preferred method of sharing information.

3.3 Software Interfaces

- Graphical User Interface (GUI): The graphical user interface of the software is the primary interface for users to interact with the software. It includes features like menus, icons, buttons, and other graphical elements that allow users to navigate and interact with the software.

- **Optical Character Recognition (OCR) Interface:** The OCR interface allows the software to read and interpret the handwriting on the scanned images of student answers. It includes features like image processing, character recognition, and machine learning algorithms.
- **Grading Engine Interface:** The grading engine interface is the core functionality of the software that is responsible for grading the student answers based on the grading criteria. This interface includes features like rule-based grading, machine learning algorithms, or a combination of both.
- **Database Interface:** The database interface allows the software to interact with a database or data storage system to store and retrieve data related to the student answers, grading criteria, and reports.
- **Communication Interfaces:** Communication interfaces enable the software to communicate with other users or systems, such as email or messaging services, which can also be used to share feedback, grades, or reports.
- **Reporting Interfaces:** The reporting interface enables the software to generate reports, which can be used to provide feedback to students, teachers, or administrators. This interface includes features like visualizations, charts and graphs to provide insights on student performance and progress.

The software interfaces in the software should be designed to be easy to use, reliable, and efficient, allowing for the accurate and effective evaluation of student answers.

3.4 Communications Interfaces

- **Email:** An email interface is used to send feedback, grades, or reports to teachers or students. This interface includes features like email templates, automated messages, and customizable email settings.
- **File sharing:** A file-sharing interface is used to share digital files, like graded answer sheets, grading criteria, and other relevant documents. This interface includes features like file uploading, sharing permissions, and version control.
- **API integration:** An API integration interface is used to connect the software to other systems or software applications using an API. This interface includes features like API documentation, authentication, and data exchange protocols.

The communication interfaces are essential for enabling communication between users, as well as allowing the software to integrate with other systems or applications. These interfaces should be designed to be user-friendly, secure, and efficient, and should provide the necessary functionalities to meet the needs of the users.

4. System Features

4.1 Optical Character Recognition (OCR)

4.1.1 Description

Optical character recognition enables the software to recognize the characters in the handwritten answers and convert them into digital text. It is the first and foremost step in implementing this system. This feature is triggered in response to the event fired by the teacher to compute marks.

4.1.2 Functional Requirements

- **Accurate recognition of text:** The OCR software should be able to accurately recognize the text in the scanned images, without introducing errors or omissions.
- **Input file format compatibility:** The system should be able to accept handwritten answer sheets in various formats, including scanned images or electronic documents.
- **Batch processing:** The OCR software should be able to process multiple pages of a document at once, such as by automatically detecting and separating pages.
- **Accuracy and error detection:** The OCR software should have built-in mechanisms to detect errors, such as through spell-checking, and should provide means to correct those errors.
- **Integration with other software:** The OCR software should be able to integrate with other software, such as document management systems, to provide seamless end-to-end solutions.

4.2 Automated assessment

4.2.1 Description

Automated grading enables the software to evaluate the answers based on predefined criteria and grading rubrics. The OCR recognized characters are compared to the model answer, character by character, to determine the degree of similarity.

4.2.2 Functional Requirements

- **Accurate recognition of student answers:** The system should be able to accurately recognize the student answers, which may be in various formats such as text, numbers, symbols, or even images.
- **Handling of multiple answer types:** The system should be able to handle a variety of answer types, including multiple-choice questions, short answers, long-form answers, and essays.
- **Real-time feedback to students:** The system should be able to provide instant feedback to students on their answers, such as correct answers, incorrect answers, or partial credit.
- **Statistical analysis:** The system should be able to generate reports and statistics on the performance of individual students, groups of students, or the entire class, which can be useful for identifying knowledge gaps and evaluating the effectiveness of teaching methods.

- Flexibility and customizability: The system should allow customization of grading criteria, weights, and parameters to fit the specific needs of the assessment and the educational context.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- Load time less than 5 seconds
- Response to user interactions in milliseconds
- Scalable according to increase in number of users
- Available 24/7, with minimum downtime

5.2 Safety Requirements

- Compliance with relevant safety regulations and standards
- Backup strategy to ensure prevention of data loss.

5.3 Security Requirements

- Compliance with relevant security regulations and standards
- The users require proper authentication in order to avoid cheating and fraudlery.

5.4 Software Quality Attributes

- Adaptability
- Availability
- Correctness
- Flexibility
- Maintainability
- Portability
- Reliability

5.5 Business Rules

Since the system involves the employment of free version of machine learning models, commercial viability of our system is restricted to some extent