Software Design Document for project Automated Evaluator of Handwritten Malayalam Answer Scripts

Devi Krishna M K
Maria Viji George
Navneeth Variar
Niranjan Neelakantan(Team Leader)
Supervised by: Prof. Princy Ann Thomas

December 16, 2022

1 Introduction

1.1 Purpose

The purpose of this document is to delineate the design of our project titled - Automated Evaluator of Handwritten Malayalam Answer scripts in different design viewpoints. It is intended for both the evaluators and management concerned with the evaluation of hand-written answer scripts in malayalam language as well as end-users of the system.

1.2 Scope

The "Automated evaluator of Handwritten Malayalam Answer scripts" will be a platform used for the automatic evaluation of hand-written answer scripts and generation of marks. The evaluator can download the answer paper template as a pdf and print it out. It is in this answer paper template that the candiates are asked to write their answers. The evaluator should scan and upload the answer scripts into the system for getting the answer scripts evaluated. The system will save the marks allocated to individual questions based on the answers given. The total marks are also calculated and saved in the system. The system will be designed to :

- save the time of evaluators by automating the evaluation of malayalam handwritten answer scripts.
- compare the written answers to the answer key and allocate marks to individual questions.
- calculate the total marks and save the total marks.
- provide platform independency and maintainability.

The product saves the time of the user by automatically evaluating malayalam hand-written answer scripts and granting marks to the correct answers thereby eliminating manual evaluation.

1.3 Overview

Section 2 provides the overview of the system in a nutshell.Section 3 gives the detailed architecture of the system.The detailed design of the software interface and the hardware interface is given in this section.Section 4 gives the details of the data design.Section 5 gives the functional description of each component.Section 6 gives the user interface design in detail.The UI designs are also given in this section.

1.4 Reference Material

- IEEE SDD Standard document

1.5 Definitions and Acronyms

Term	Definition
Software Design Document (SDD)	Used as the primary medium for communicating software design information.
DFD	Data Flow Diagram

2 System Overview

This product is a website for the automatic evaluation of malayalam hand-written answer scripts.This is an alternative for the existing manual system of evaluating malayalam handwritten scripts.This software allows the teachers of the pre-primary schools to conduct spelling test to their students in malayalam language. The software will help the teachers to evaluate the answer scripts in a more simplified way. The website will have a user-friendly interface. Buttons and dialog-boxes for insertion of scanned versions of malayalam handwritten scripts will be available.The user can select Upload answer script button from the menu to upload an answer script. To view the score generated,user can select View score button from the menu. Buttons will be available to download the score-card generated in pdf format. There is also an option to download the answer template which can be downloaded and printed by the teachers. The students are supposed to fill the answers in the this template.

3 System Architecture

3.1 Architectural Design

First, the machine learning model is created and trained with as much as data possible(Image vectors of Malayalam characters). After, optimising the model, it is stored into a compressed format (like pickle file) for prediction purpose in future. Then we create a REST API using Flask microframework. It acts as a server and helps to create the backend composed of the saved trained machine learning model and the webpage's front end.

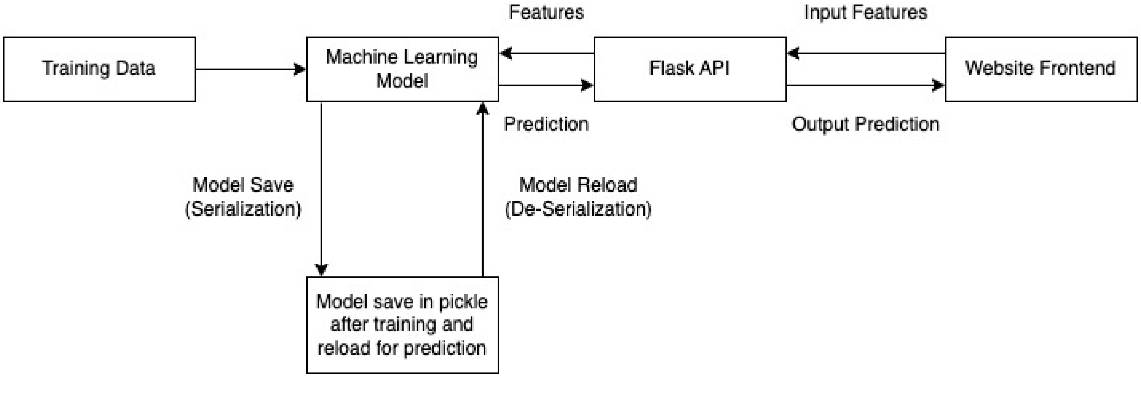

Features
Input Features
Training Data
Machine Learning Model
Flask API
Website Frontend
Prediction
Output Prediction
Model Save (Serialization)
Model Reload (De-Serialization)
Model save in pickle after training and reload for prediction

Figure 1: Architectural Design

3.1.1 Software Interfaces

The software interface should follow the Model-View-Controller (MVC) model for rendering and modeling data objects.The interface must be able to connect to a database to store XML schema defined using XSD and data streams. Source and destination formats for data must include XML and may also include: Extensible Stylesheet Lanaguage Transformation (XSLT), JavaScript Object Notation (JSON), Portable Document Format(PDF), Comma Separated Value (CSV), and American Standard Code for Information Interchange (ASCII). The website requires the following software interfaces to run.

3.1.2 Hardware Interfaces

Since we are dealing with cross-platform as well as the cross-device types, the hardware components mainly consist of gadgets used by evaluators with limited computational capability.

3.2 Decomposition Description

The System is divided into 3 modules based on the functionality of the system, that is, it is divided into modules for each of the functionality the system would provide. The Modules are :-

3.2.1 Authentication

This Module mainly deals with user Authentication. The evaluator could create an account in the platform using their email id and a password.Then, the request from the user to gain access to the system would be received. This request would be processed and if verified, the user would be authorized into the system and can make use of its functionalities.

3.2.2 Handwritten Malayalam character recognition

This Module tackles the task of converting the uploaded Malayalam handwritten answer scripts in jpg format to the machine encoded textual format which is needed to carry out downstream tasks. The module also handles the necessary pre-processing needed to improve the quality of the inputs.

3.2.3 Similarity matching

This Module compares the answers in the answer scripts to that in the answer key and tries to recognize similarities based on the information known about them. The documents are represented as vectors of features, and compared by measuring the distance between these features.

3.2.4 Data Flow Diagram

The Dependencies between the modules and the flow of information is expressed in the DFD diagrams shown in figure 2 and figure 3.

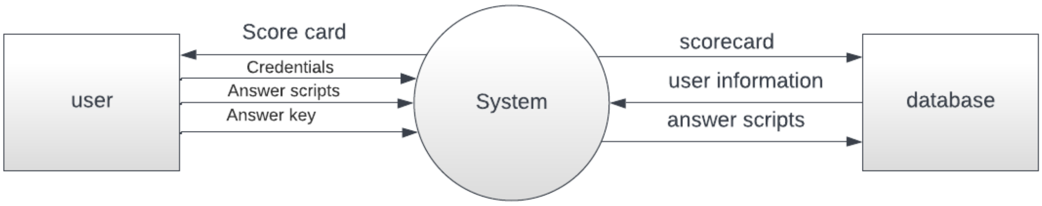


Figure 2: Data Flow Diagram Lvl 0

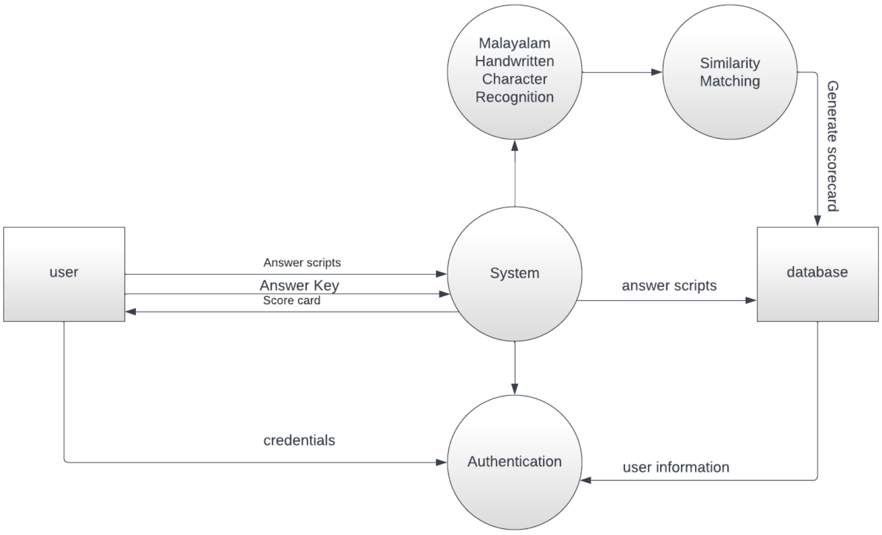


Figure 3: Data Flow Diagram Lvl 1

3.3 Design Rationale

This simple architecture shown above consisting of the pre-trained model, flask API and a front-end is one of the most basic working architectures used in low-scale machine learning implementations. Compared to Django, Flask is open to change, and it adds a layer of flexibility to the web app development process. Furthermore, the framework is straightforward to use and understand, even for a newbie.

4 Data Design

4.1 Data Description

The authentication details of the evaluator are the first details stored in the database. Scanned copies of the answers are uploaded from the local storage of the user via the frontend web page. These images are passed on to the database as well as input to the pre-trained model. The correct answer provided by the teacher is compared with the output given by the model. The teacher's answer is already stored in the database. The output is the corresponding input file's name and the marks obtained in percentage. It is termed as the score card and is stored in the database itself.

4.2 Data Dictionary

- User : The account details are first data given by the user. Then she/he creates the answer template which consists of the correct answer to be compared with. Along with this, the scanned copies of the papers to be evaluated are also given by the user, which is given to the system.
- System: The system accepts the scanned copies as the input and recognizes the handwritten characters and converts to digital characters, which is the output. Also, it accepts account details as well as answer template from user and gives it to database.
- Database: Accepts the details of user via the system. Stores the scanned images, correct answer provided by evaluator, and finally the score card.

5 Component Design

6 Human Interface Design

6.1 Overview of User Interface

The website will have a user-friendly interface. Buttons and dialog-boxes for insertion of scanned versions of malayalam handwritten scripts will be available.The user can select "Upload answer script" button from the menu to upload an answer script. To view the score generated,user can select "View score" button from the menu. Buttons will be available to download the score-card generated in pdf format. There is also an option to download the answer template which can be downloaded and printed by the teachers. The students are supposed to fill the answers in the this template.

6.2 Screen Images

6.2.1 Login Screen User Interface

There will be a login screen user interface where the user can create an account where he/she can upload the answer scripts and marks evaluated will be displayed.

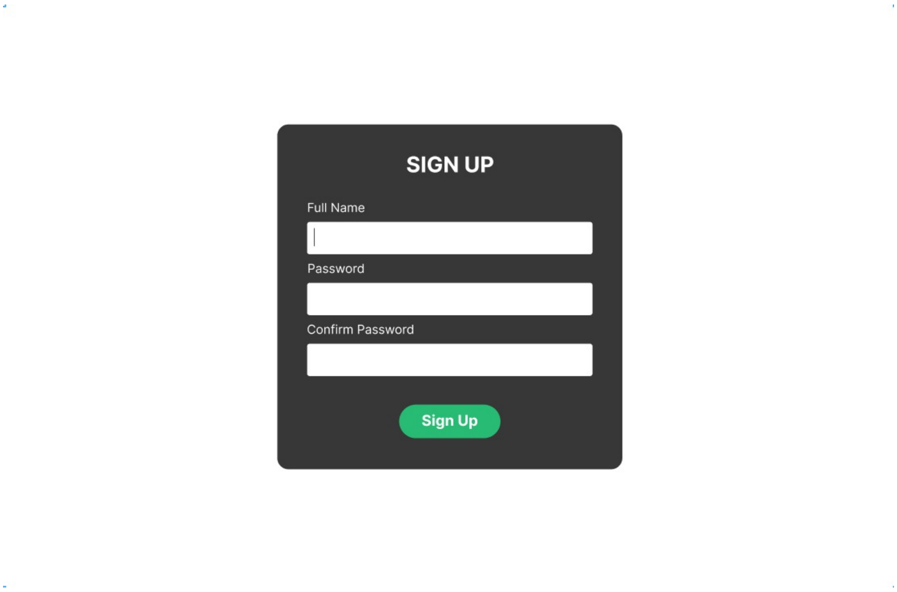
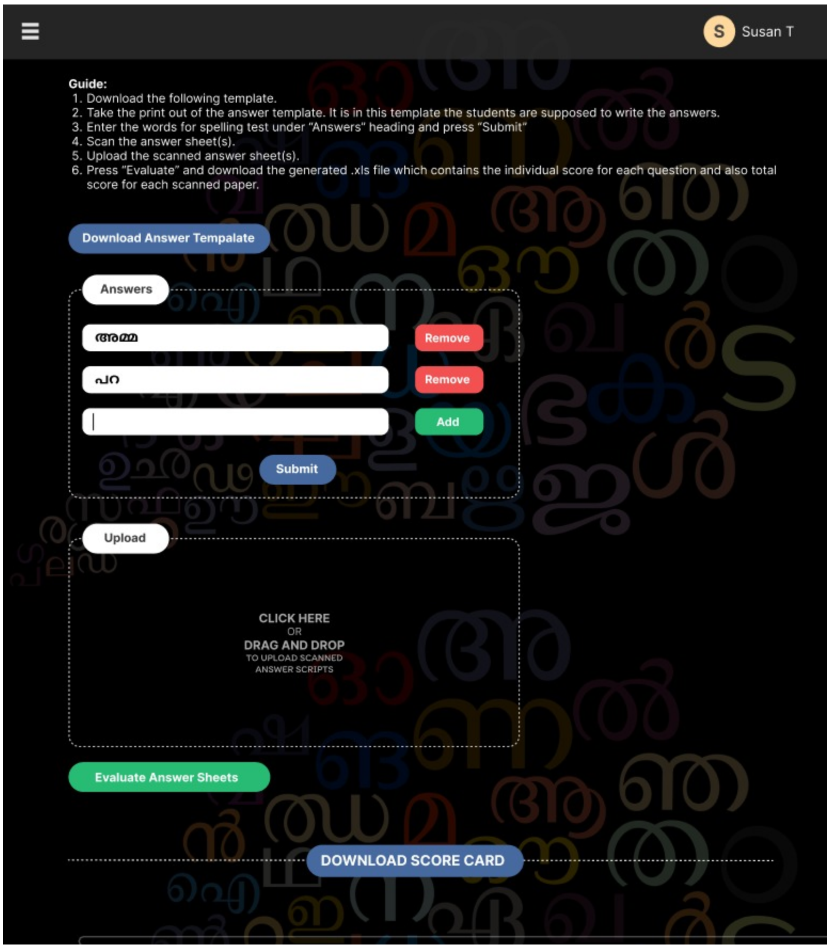


Figure 4: Login Screen User Interface

6.2.2 Output User Interface

There will be an output user interface where the user can set the answer key. There is also an option to upload the answer scripts. A button to download the answer template is also present which generates an answer template corresponding to the answer key. An option to download the scorecard after evaluation is also present.



Guide:

1. Download the following template.
2. Take the print out of the answer template. It is in this template the students are supposed to write the answers.
3. Enter the words for spelling test under "Answers" heading and press "Submit"
4. Scan the answer sheet(s).
5. Upload the scanned answer sheet(s).
6. Press "Evaluate" and download the generated .xls file which contains the individual score for each question and also total score for each scanned paper.

അമ്മ

പറ

Figure 5: Output User Interface

6.3 Screen Objects and Actions

- Login Screen User Interface - The Evaluator can create an account by entering the full name, and creating a password.
- Output User Interface - The evaluator can create the answer template by adding answers to each question. Also, here the user can upload scanned copies of handwritten answer scripts as well as download the final score card after evaluation

7 References:

[1] Abdul Rehman Javed Natalia Kryvinska Shahab S. Band Muhammad Farrukh Bashir, Hamza Arshad. Subjective answers evaluation using machine learning and natural language processing. *IEEE Access*, 9(25), November 2021.

[2] Chandran Saravanan Samatha P Salim, Ajay James. Proposed method to malayalam handwritten character recognition using residual network enhanced by multi-scaled features. 2019.

[3] Jawahar C.V Vijay Rowtula, Subba Reddy Oota. Towards automated evaluation of handwritten assessments, 2019.