**VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM 590014**

****

Software Engineering

**“Automatic Answer Script Correction - Text”**

By

Deepthi Bhat (1BM16CS003)

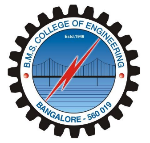
Aditi Awasthi (1BM16CS008)

Medhini Oak (1BM16CS047)

Software Engineering

Alternative Assessment Test

carried out at

****

Department of Computer Science and Engineering

BMS College of Engineering

(Autonomous college under VTU)

P.O. Box No.: 1908, Bull Temple Road, Bangalore-560 019

Aug-Dec 2018

TITLE OF THE PROJECT

Automatic Answer Script Correction

ABSTRACT

The current physical system of written answer script correction involving each answer being read and analysed by an examiner and, in most cases, by another examiner for further verification is time-consuming and the cost required to maintain such a huge database manually is very high. This system has been developed to the digitalize the process of answer script correction using technologies like image processing and natural language processing thereby making the process more time and cost efficient. As the end result, this system is able to provide nearly accurate grades for a given answer script and maintain a database for it.

INTRODUCTION

Automatic Answer Script Correction is a software that is designed for the use of examiners to automatically grade answer scripts of students based on a given answer scheme provided by the examiner himself/herself.

In this system, the examiner will be able to upload multiple questions and the respective answers to the database. He can upload the photos of the answer scripts of each student to the system. Then the system will analyse the answer scripts of each student using image processing and compare the answers written by the student against the scheme given by the examiner and allot marks for the student.Using this system can reduce a lot of time and effort for teachers. They can avoid reading each answer script and comparing it with the scheme given for allotting grades.

SOFTWARE REQUIREMENT SPECIFICATION

FUNCTIONAL REQUIREMENTS

1. The software should be able to accept and store the questions and corresponding answers given by the examiner in the database for reference.
2. It should be able to take the answer scripts of the students in picture format and convert the written answers in the picture to text.
3. It should compare the answers in the scheme with the answers written by the student and be able to allot marks based on an effective comparison.
4. It should give the teacher a display of how much the answer has matched the scheme (in terms of keywords) and allow the teacher to make any other valid matchings.
5. The system should give a list of the total marks of all the students to the teacher after correcting all the answer scripts.

NON-FUNCTIONAL REQUIREMENTS

1. Accuracy:

* The answer script should be processed into text format as it is written.No words, symbols or spaces should be missing.
* Each answer should be graded against the given scheme after performing maximum matching of answers.

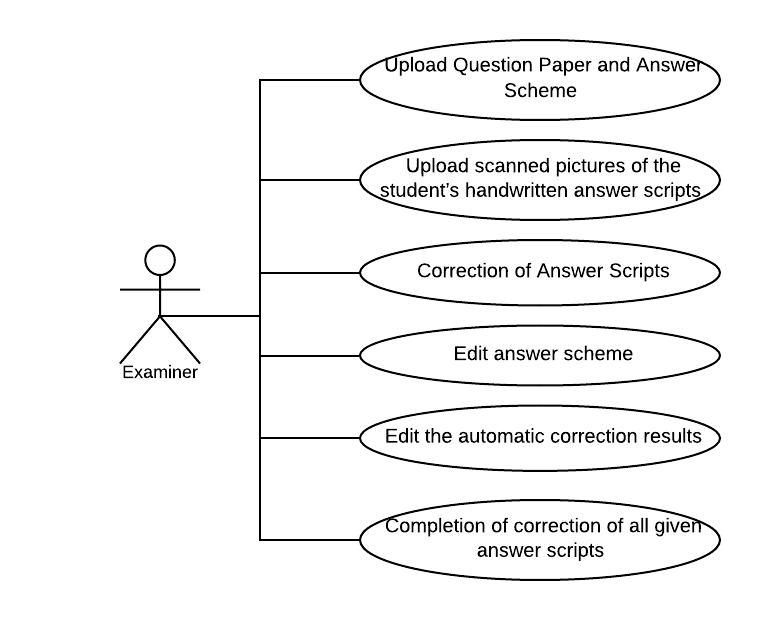
1. Reliability:

* Every student should get appropriate total marks according to the answer he/she has written.
* It should not allot the marks of one student to another student.

1. Performance:

* The system should match the answers to the maximum possible extent so as to reduce the work of the examiner.
* The marks calculated should be properly distributed to all the keywords in the answer.

USE CASES



1. Upload Question Paper and Answer Scheme

The examiner types the question and answer in the system window and submits it to add it into the database. He can upload multiple questions and corresponding answers at the same time.The answer to the question is raked and the keyword-weight pairs are stored in the database.

1. Upload scanned pictures of the student’s handwritten answer scripts

The examiner selects a student and uploads the pictures of his answer scripts together. He has to do this consecutively for every student.

1. Correction of Answer Scripts

Once all the answer script pictures of a student are uploaded, the pictures are converted to text and and raked to find the keyword-weight pairs in the answer. These pairs are compared with the pairs in the scheme and the matching pairs are graded according to their weights.

1. Edit answer scheme

The examiner will be shown the rake-results of the answer scheme. If the results do not entirely match the requirement, then he can edit the weight system. The changed keyword-weight pairs will be stored in the database.

1. Edit the automatic correction results

The examiner can see the list of matched keywords between the answer and the scheme. He can also see a list of unmatched keywords from the scheme and the answer. He is allowed to match those extra keywords to the un-matched keywords which can be considered be synonymous.

1. Completion of correction of all given answer scripts

Once the answer scripts of all the students have been evaluated the examiner is displayed a list of all the students with their total grades. He can then send individually mail every student his/her grade and send a final copy as a sheet to the Department Office.

1. Adding of new students to the database

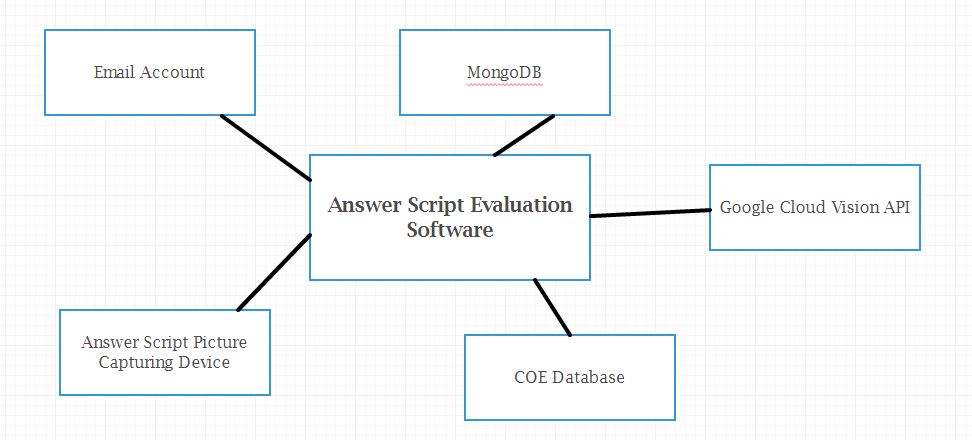
The examiner has the provision to add new students to the database. He should add the student’s USN number and email-id to the database.

DESIGN MODELS

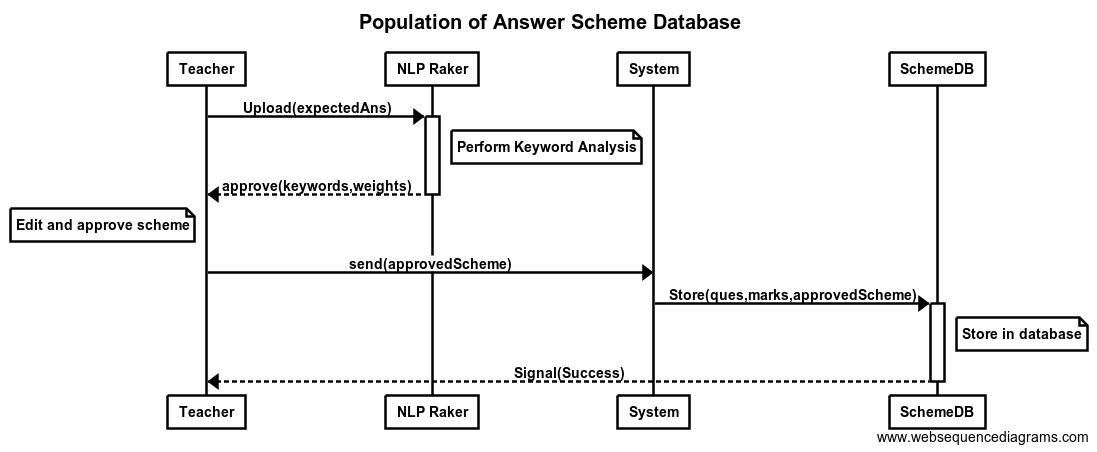
SOFTWARE MODULES

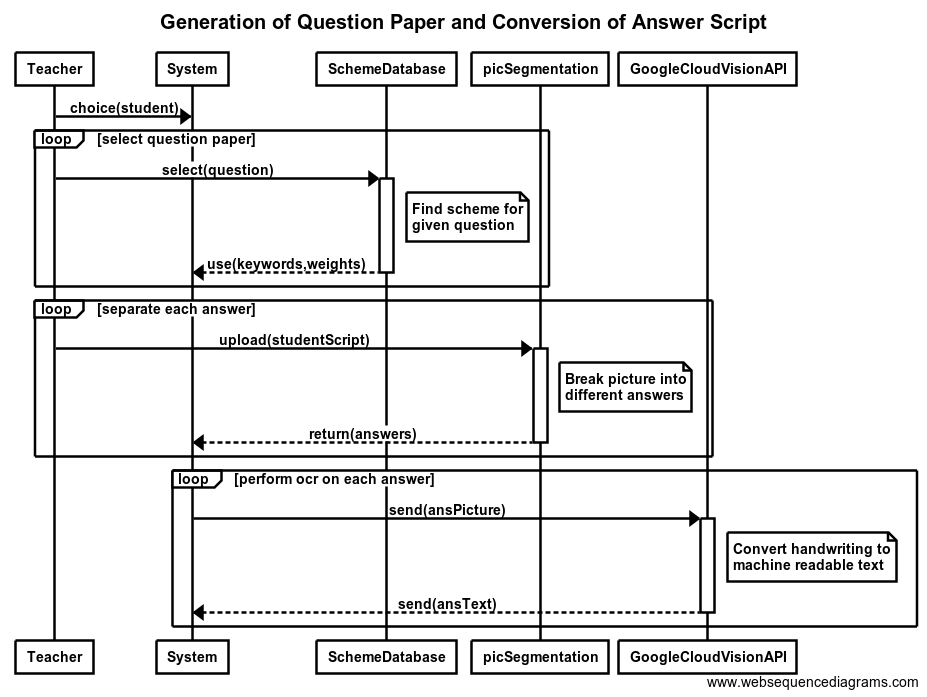
1. Upload of question scheme and answer script of student
2. Answer Image Segmentation
3. Handwriting Optical Character Recognition (OCR)
4. No -SQL Database (Mongo)
5. Rapid Automatic Keyword Extraction (RAKE)
6. Keyword matching and marks calculator
7. E-mailer system to notify student and COE office of the marks obtained

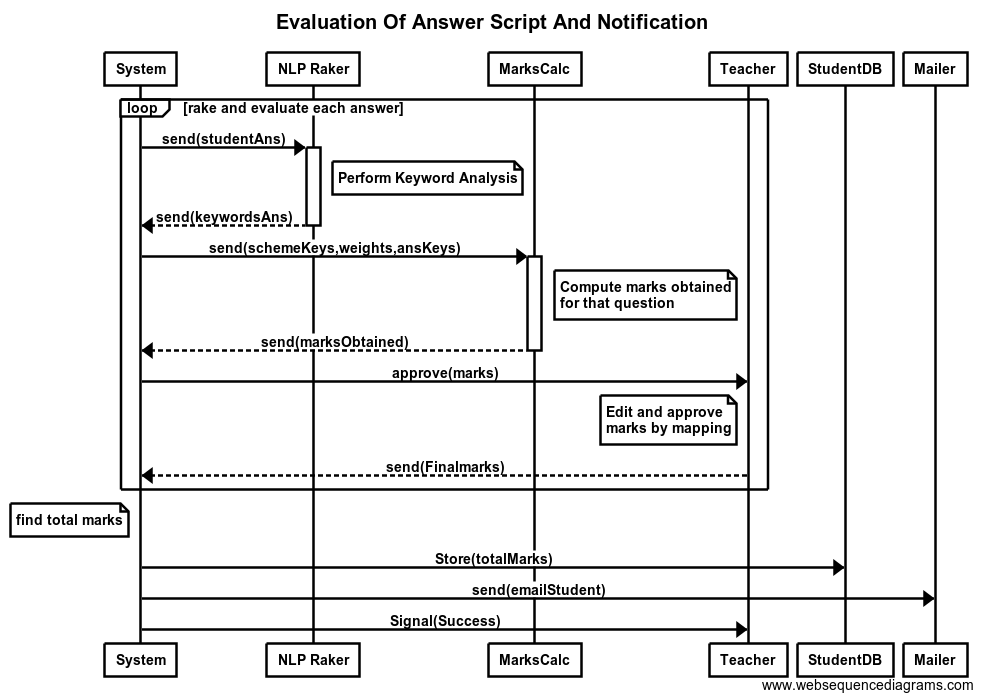
CONTEXT MODEL



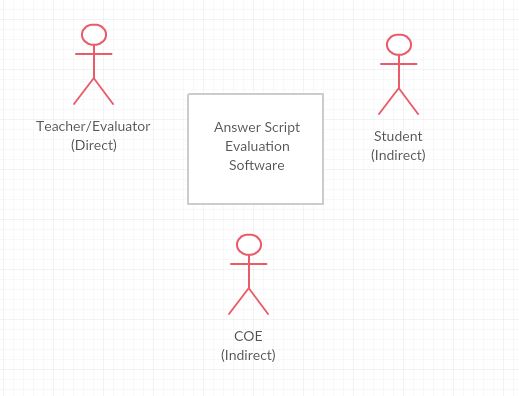
SEQUENCE DIAGRAMS



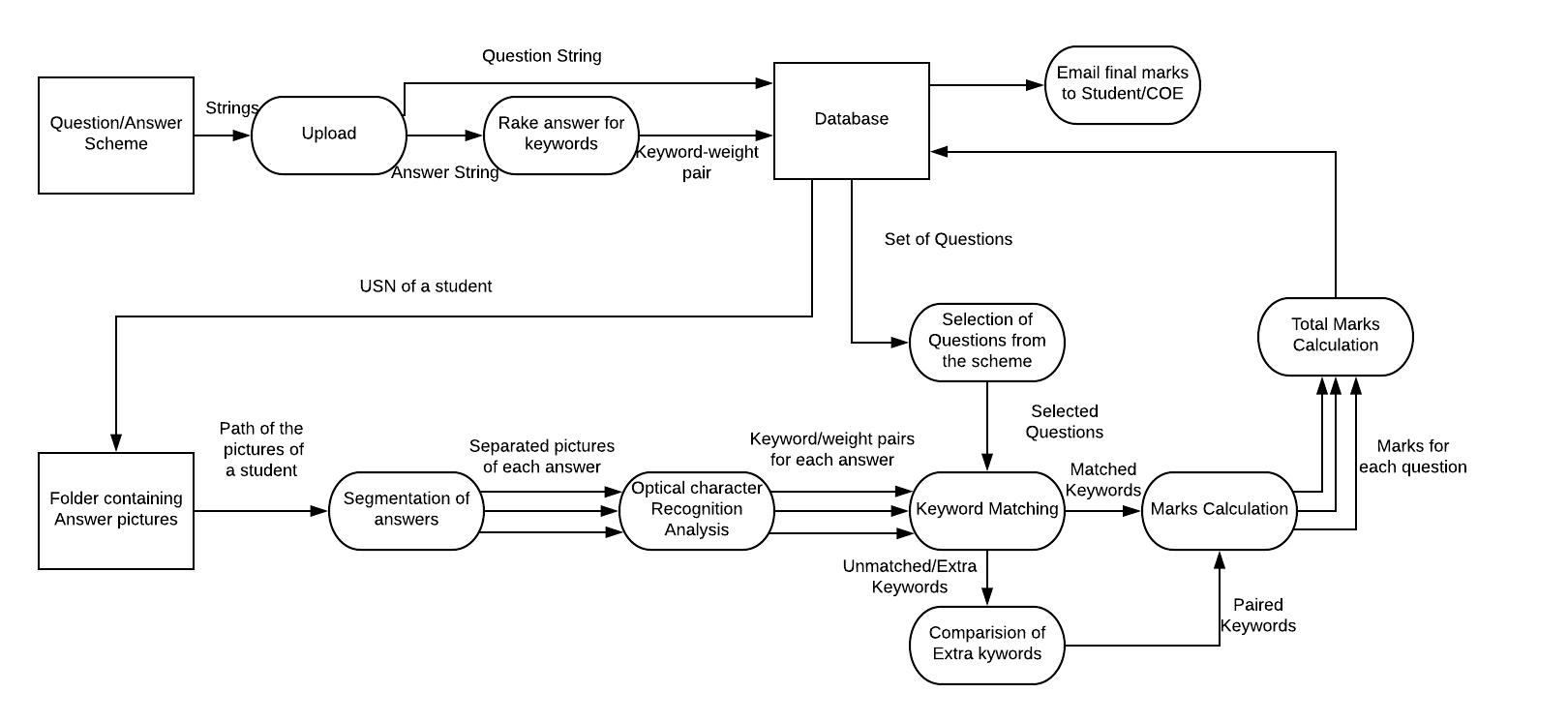




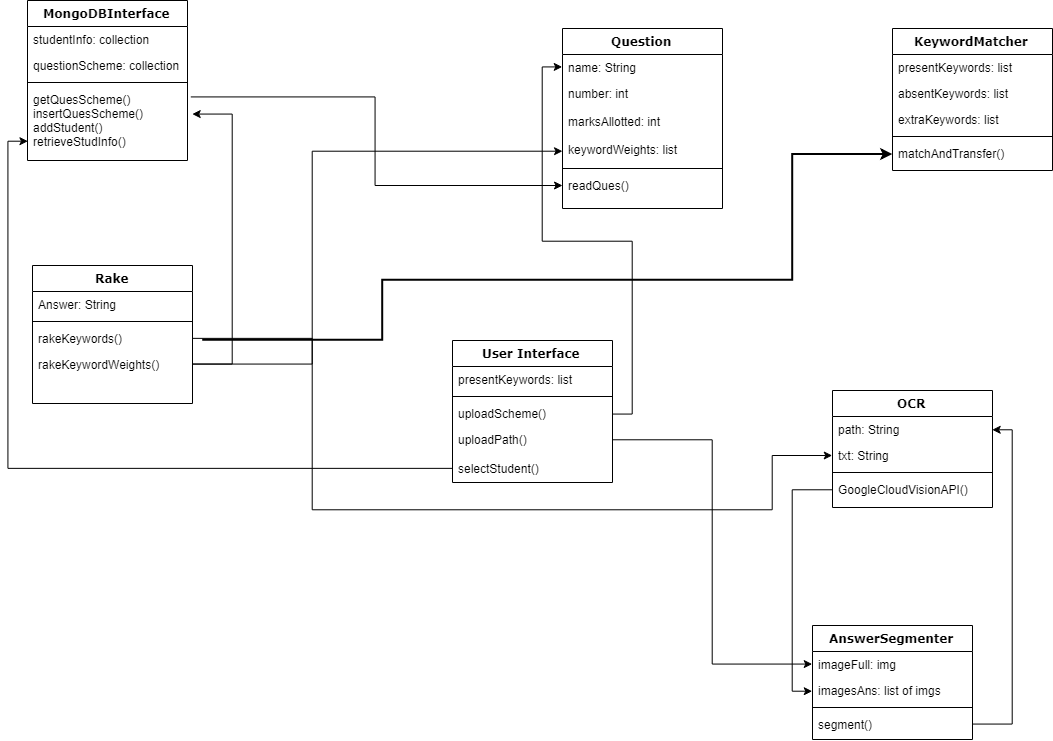
INTERACTORS



DATA FLOW MODEL



OBJECT ORIENTED DESIGN MODEL



DETAILED DESCRIPTION OF MODELS

A. Upload of question scheme and answer script of student

The teacher must enter the question along with the correct answer to the it and the number of marks allotted to the question.

Once the required scheme is uploaded, for a particular instance, the question paper is selected and the scanned pictures of the answer script of a particular student are uploaded into the application.

For creation of graphical user interface for all modules, 2 libraries - Tkinter, easygui have been used. These are the most commonly used GUI toolkits in python.

B. Answer Image Segmentation

Libraries used : opencv, numpy, PIL

**OpenCV** (*Open source computer vision*) is a library of programming functions mainly aimed at real-time computer vision. Hough Transform is a popular technique to detect any shape, if you can represent that shape in mathematical form. It makes use of the numpy library.

The starting and ending coordinates of each line which separates two answers in the answer script are found. These coordinates are then used to crop out the different answers from the original image and store them as different images. The PIL (Python Imaging Library) has been used for this purpose.

C. Handwriting Optical Character Recognition (OCR)

Libraries used : google-api-python-client

Google Cloud AI provides modern machine learning services, with pre-trained models and a service to generate own tailored models. This project makes use of Google’s neural-net-based ML service, since it has better training performance and increased accuracy compared to other deep learning systems and its services are fast, scalable, and easy to use.

The Vision API can detect and extract text from images. These are the annotation features that support OCR:

* TEXT\_DETECTION
* DOCUMENT\_TEXT\_DETECTION
* HANDWRITING\_DETECTION (beta phase)

The project makes use of HANDWRITING\_DETECTION annotation feature. It employs component-based software development approach i.e. it uses commercial off-the-shelf third party services to achieve its objective. Developed for research purposes, this Handwriting Detection Google Cloud Vision API is still in beta phase and is offered to the public for research purposes. A Google Cloud account has been created and the Cloud Vision API enabled. After generation of a private key to make use of the API’s services, it can be used to convert the handwriting presented in the segmented answer scripts and store it in text files, which are RAKEd again.

D. No-SQL Database

Libraries used : pymongo

MongoDB is a DBMS that uses a document-oriented database model.Our project makes use of it because it involves processing jobs and data that doesn't fit well in a rigid relational model. Instead of using tables and rows as in relational databases, the MongoDB architecture is made up of collections and documents, which is very well-suited to the storage of a variable number of keywords along with the weight of each keyword in json format.

For the purpose of uniformity, student database is also maintained in MongoDB itself.

E. Rapid Automatic Keyword Extraction (RAKE)

Libraries used : rake-nltk

Rapid Automatic Keyword Extraction algorithm is a domain independent keyword extraction algorithm which tries to determine key phrases in a body of text by analyzing the frequency of word appearance and its co-occurrence with other words in the text.

This algorithm runs in two different places in the project. First, it ranks the keywords in the scheme uploaded by the teacher and assigns weights to every keyword present in the correct answer.

Second, once the answer written by the student is converted to a computer readable format through OCR, it identifies the keywords present in the answer, making it easier to compare the correct answer and the answer written in the answer script

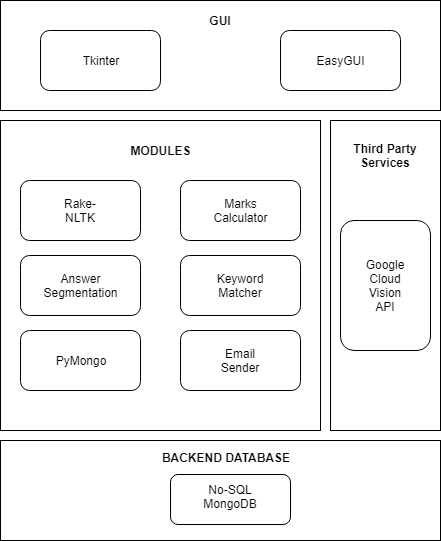
F. Keyword Matching And Marks Calculator

The examiner is offered a way to explicitly view the extracted keywords from the student’s answer and match them to the scheme’s keywords manually. This helps avoid errors which might occur due to different interpretation and wording used by the student as compared to the correct answer provided by the teacher.

The marks calculator compares the keywords which should be present in the answer and the keywords which are present in the given answer script. Marks allotted for that question are calculated using an algorithm which takes into account the weights of the keywords present in the answer and scales it according to the maximum marks which can be awarded for that question.

ARCHITECTURAL DESIGN

**LAYERED ARCHITECTURE**



Identifying Subsystems

Our system will make use of many off the shelf software packages and libraries.

* OpenCV (image processing)
* Rake (natural language processing)
* Google Cloud Vision API (used to locate nearby garages)

System Organization

**CLIENT-SERVER MODEL**

User interaction will take place on the client side of the system and make function calls to the various modules to provide their functionalities. The client (teacher) will avail the services provided by these modules to fulfill his/her purpose and populate the database accordingly.

Architectural Style

**EVENT-DRIVEN SYSTEM**

An architectural style provides a framework for a system which includes software components, its properties, and the relationships among them. Event-driven architecture pattern promotes the production, detection, consumption of, and reaction to events. One specific event is evaluating an answer script. It is the system’s focus, and thus is the reason why event-driven architecture is most suitable. When an answer script is uploaded, it causes software components to change and others to react. Therefore our system architecture is based on the event of the teacher wanting to evaluate an answer script, which in turn triggers other events.

ESTIMATION AND SCHEDULE

ESTIMATION

| **Module Name** | **Estimated Lines of Code** |
| --- | --- |
| Upload of question scheme and answer script of student | 450 |
| Answer Image Segmentation | 1450 |
| Handwriting Optical Character Recognition (OCR) | 4530 |
| No -SQL Database (MONGO) | 1030 |
| Rapid Automatic Keyword Extraction (RAKE) | 7570 |
| Keyword matching and marks calculator | 560 |
| E-mailer system to notify student and COE office of the marks obtained | 2310 |
| **Total LOC** | **17900** |

Average Labour rate per month $10000

Effort 500 LOC per month

Cost of each Line of Code $20

Estimated Cost of the entire project $358,000

Estimated time required for the project 36 person-months

SCHEDULE

| **Module Name** | **Estimated Number of DAYS** |
| --- | --- |
| Upload of question scheme and answer script of student | 10 |
| Answer Image Segmentation | 10 |
| Handwriting Optical Character Recognition (OCR) | 25 |
| No -SQL Database (MONGO) | 15 |
| Rapid Automatic Keyword Extraction (RAKE) | 20 |
| Keyword matching and marks calculator | 5 |
| E-mailer system to notify student and COE office of the marks obtained | 5 |
| **Total Number of Days** | **90** |

TEST CASES

Testing is aimed at finding errors in the system under test and giving confidence in its correct behavior by executing the system with selected input situations.

Unit Testing

Unit testing will involve going through each module of this project, asserting them with different parameters.

Integration Testing

The goal of this method is to take all the individual unit test cases and combine them to form a complete or major part of the system. The individual components are proved through this method by proving through the environment itself. The main focus will be to not avoid any smaller cases that may arise through the individual components. Its simplicity and efficiency stems from its capability to cover a significant amount of individual unit cases.

Acceptance Testing

Acceptance test cases are events that the customer wishes to specify in order to ensure they handled properly. In order to carry out the integration testing mentioned above, acceptance test cases must be generated in order to understand what the user-like workloads will be. The following test cases show the required input, pass/fail criteria, and the various scenarios that can occur.

TEST CASE 1: Wrong Path

INPUT: User gives the wrong path of the pictures of the answer sheets.

OUTPUT: An Error Message should be shown asking for the path to have a folder with only the images of the answer script.

TEST CASE 2: Unexpected Termination

INPUT: The User terminates the program unexpectedly mid-evaluation of questions

EXPECTED OUTPUT: All the questions that have completed processing should be stored in the database and the data of any unfinished question should be removed.

TEST CASE 3: Unavailable Scheme

INPUT: The User tries to evaluate answer scripts before the scheme has been provided.

EXPECTED OUTPUT: An error message should be displayed asking the user to provide the scheme before evaluating the papers.

TEST CASE 4: Unclear Input

INPUT: The picture taken isn’t clear enough for the Google API to understand the words.

EXPECTED OUTPUT: A warning message saying that bad handwriting may produce unexpected results should be displayed and the User should decide whether he should proceed with execution.

TEST CASE 5: Correct Input

INPUT: The user provides with the scheme and then provides the path of the answer script correctly. The images are very clear and the handwriting has high readability.

EXPECTED OUTPUT: The questions are properly segmented, handwriting properly recognised, keywords properly compared and the student is awarded appropriate marks that he deserves which gets emailed to him/her.

TEST CASE 6: Unsegmented Answers

INPUT: The answers haven’t been segmented properly and all the answers have been taken as one.

EXPECTED OUTPUT: A warning message should be displayed saying that only one answer has been recognised and ask the user whether it should continue with execution.

CONCLUSION

This software automates the task of an examination answer script evaluator, making his/her job simpler. An important idea used here is “evaluator feedback” i.e. for each decision made by through machine learning or NLP, the developers have accounted for lack of precision and provided the evaluator an option to rectify oversights in such situations.

Overall, the technical challenges faced are solved with an efficient software program. Despite some shortcomings, mostly due to time constraint, the project helps understand the software engineering approach and its significance. Through the software engineering application, the answer script correction automation is implemented successfully.