

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

Jnana Sangam, Belagavi-590018



A Project Report

ON

**“AUTOMATED EVALUATION OF INTERNAL PAPER”**

BY

**ANKITHA  
ASHWINI V SHETTY  
KASHYAP BHANU DAS  
KEERTHI**

**4MT15CS012  
4MT15CS019  
4MT15CS047  
4MT15CS049**

**Project Guide**

**Mr. Manjunatha A. S.,  
Sr. Assistant Professor,  
Department of CS & E,  
MITE, Moodbidri.**

**February – May 2019**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**  
(Accredited by NBA)

**MANGALORE INSTITUTE OF TECHNOLOGY & ENGINEERING**  
(An ISO 9001:2015 Certified Institution)  
**BADAGA MIJAR, MOODABIDRI**  
**DK DIST-574225**

# **ABSTRACT**

Today, the education system is totally changed by the technology. It has become more interesting and informative by projector teaching, online tutorial, online teaching video and animation etc. There is an intense use of technology to teach the student. But evaluation process is still done in the traditional way. The correction of examination paper mainly depends on teachers' manual correction. If machine learning can replace the original manual processing, the inconvenience and the rate of error caused by manual operation can be effectively reduced. The proposed project aims to solve the problem of consuming too much time and energy in correcting exam papers manually. Evaluating the answer paper in online mode is good in comparison to offline mode which is faster.

# ACKNOWLEDGEMENTS

The satisfaction and the successful completion of this project would be incomplete without the mention of the people who made it possible, whose constant guidance encouragement crowned our efforts with success.

This project is made under the guidance of **Mr. Manjunatha A. S.**, Sr. Assistant Professor, in the Department of Computer Science and Engineering. We would like to express sincere gratitude to our guide for all the helping hand and guidance in this project.

We would like to thank our project coordinators **Mr. Manjunatha A. S.**, Sr. Assistant Professor and **Mr. Guruprasad**, Assistant Professor in the Department of Computer Science and Engineering, for their cordial support, valuable information and guidance, which helped us in completing this project through the various stages.

We would like to express appreciation to **Dr. Venkatramana Bhat P.**, Professor and Head of the department, Computer Science and Engineering, for his support and guidance.

We would like to thank our Principal **Dr. G.L. Easwara Prasad**, for encouraging us and giving us an opportunity to accomplish the project.

We also thank our management who helped us directly and indirectly in the completion of this project.

Our special thanks to faculty members and others for their constant help and support.

Above all, we extend our sincere gratitude to our parents and friends for their constant encouragement with moral support.

<b>ANKITHA</b>	<b>4MT15CS012</b>
<b>ASHWINI V SHETTY</b>	<b>4MT15CS019</b>
<b>KASHYAP BHANU DAS</b>	<b>4MT15CS047</b>
<b>KEERTHI</b>	<b>4MT15CS049</b>

# TABLE OF CONTENTS

<b>Contents</b>	<b>Page No.</b>
<b>ABSTRACT</b>	i
<b>ACKNOWLEDGEMENT</b>	ii
<b>TABLE OF CONTENTS</b>	iii
<b>LIST OF TABLES</b>	vi
<b>LIST OF FIGURES</b>	vii

<b>Chapter No.</b>	<b>Title</b>	<b>Page No.</b>
<b>1.</b>	<b>INTRODUCTION</b>	<b>1</b>
1.1	Introduction	1
1.2	Problem Statement	2
1.3	Objectives (Purpose of the project)	2
1.4	Scope of the project	2
1.5	Organization of the report	3
<b>2</b>	<b>LITERATURE SURVEY</b>	<b>4</b>
2.1	Existing System	4
	2.1.1 Handwriting Recognition	4
	2.1.2 Automatic Evaluation	5
2.2	Proposed System	5
	2.2.1 Image acquisition process	6
	2.2.2 Handwriting Recognition System	6
	2.2.3 Answer Sheet Evaluation System	7
	2.2.4 Database	8
	2.2.5 Final Report	8

<b>3</b>	<b>SYSTEM REQUIREMENTS SPECIFICATION</b>	<b>9</b>
3.1	Overall description	9
	3.1.1 Product Perspective	9
	3.1.2 Product Functions	9
	3.1.3 Assumptions and Dependencies	10
3.2	Specific Requirements	10
	3.2.1 Hardware Requirements	10
	3.2.2 Software Requirements	10
	3.2.3 Functional Requirements	10
	3.2.4 Non-functional Requirements	11
<b>4</b>	<b>GANTT CHART</b>	<b>12</b>
<b>5</b>	<b>SYSTEM DESIGN</b>	<b>14</b>
5.1	Architectural Diagram	12
5.2	Use Case Diagram and Description	14
5.3	Sequence Diagram	16
5.4	Activity Diagram	17
5.5	Data flow Diagram	18
<b>6</b>	<b>IMPLEMENTATION</b>	<b>19</b>
6.1	Algorithm	19
6.2	Code Snippets	20
	6.2.1 Connecting to a database	20
	6.2.2 Upload questions	20
	6.2.3 Upload answer scripts	21
	6.2.4 Convert image into text	22
	6.2.5 Preprocess answer script	23
	6.2.6 Evaluate answers of category theory	24
	6.2.7 Evaluate answers of category c program	25
	6.2.8 Display results	26

<b>7</b>	<b>TESTING</b>	<b>27</b>
7.1	Test Levels	27
	7.1.1 Unit Testing	27
	7.1.2 Integration Testing	28
	7.1.3 System Testing	28
	7.1.4 Acceptance Testing	29
7.2	Test Cases	29
<b>8</b>	<b>RESULTS AND SNAPSHOTS</b>	<b>33</b>
8.1	Snapshots	33
<b>9</b>	<b>CONCLUSION AND FUTURE WORK</b>	<b>40</b>
9.1	Conclusion	40
9.2	Future Work	40
	<b>REFERENCES</b>	

# LIST OF FIGURES

<b>Fig 2.1 System Design</b>	<b>6</b>
<b>Fig 2.2 Answer Sheet Evaluation System</b>	<b>8</b>
<b>Fig 4.1 Gantt Chart</b>	<b>13</b>
<b>Fig 5.1 Architectural Design Diagram</b>	<b>14</b>
<b>Fig 5.2(a) Use case diagram for question paper</b>	<b>15</b>
<b>Fig 5.2(b) Use case diagram for evaluation</b>	<b>15</b>
<b>Fig 5.3 Sequence diagram</b>	<b>16</b>
<b>Fig 5.4 Activity diagram</b>	<b>17</b>
<b>Fig 5.5 Data Flow diagram</b>	<b>18</b>
<b>Fig 8.1 Homepage</b>	<b>33</b>
<b>Fig 8.2 Webpage to add question and key answers</b>	<b>34</b>
<b>Fig 8.3 Webpage to add student answer sheets</b>	<b>35</b>
<b>Fig 8.4 Add a question and key answer belonging to “theory” category</b>	<b>36</b>
<b>Fig 8.5 Add a question and 3 different testcase belonging to “program” category</b>	<b>37</b>
<b>Fig 8.6 Uploading answer sheets of each student by specifying their USN</b>	<b>38</b>
<b>Fig 8.7 Students result page</b>	<b>39</b>

## **LIST OF TABLES**

<b>Table 4.1: Gantt Chart of planning and scheduling of project</b>	<b>12</b>
<b>Table 7.1: Test Case</b>	<b>32</b>



# **Chapter - 1**

## **INTRODUCTION**

# Chapter 1

## INTRODUCTION

This chapter gives the introduction about the Automated Evaluation of Internal Paper.

### 1.1 Introduction

In the current evaluation system, descriptive answers given by the students are evaluated manually by the faculty. This is an error-prone process, as different professors are likely to award different marks to the same answer. The current evaluation system is tedious as faculty has to put lot of manual effort to read through and evaluate the scripts of all students. When human being evaluates anything, the quality of evaluation may vary along with the emotions of the person. Performing evaluation through computers using intelligent techniques ensures uniformity in marking as the same inference mechanism is used for all the students.

This system uses machine NLP and OCR mechanism to solve this problem. Natural language processing (NLP) is a subfield of computer science, information engineering, and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyse large amounts of natural language data.<sup>[7]</sup>

In off-line handwriting recognition, the completed writing is available as an image. This images of typed, handwritten or printed text, whether from a scanned document or a photo of a document is converted into editable text using OCR mechanism. This project presents the use of Google's open source Optical Character Recognition Software Tesseract. Tesseract is an optical character recognition engine for various operating systems. It is free software, released under the Apache License and development has been sponsored by Google since 2006. In 2006, Tesseract was considered one of the most accurate open-source OCR engines then available. Python-tesseract is an optical character recognition (OCR) tool for python. That is, it will recognize and "read" the text embedded in images. Python-tesseract is a wrapper for Google's Tesseract-OCR Engine. It is also useful as a stand-alone invocation script to tesseract, as it can read all image types supported by the Python Imaging Library, including jpeg, png, gif, bmp, tiff, and others, whereas tesseract-OCR by default only supports tiff and bmp. Additionally, Python-tesseract will write the recognized text to a file.<sup>[8]</sup>

The main aim of this project is to ensure user-friendly and more interactive software to the user. The online evaluation is a much faster and clear method to define all the relevant marking schemes. It brings much transparency to the present method of answer checking. The answers to all the questions after the extraction would be stored in a database. The database is designed as such that it is very easily accessible. Automating repetitive tasks has been the main aim of the industrial and technological revolution. The work of checking hundreds of answer sheets which more or less contains the same answer can be quite a boring task for the teachers. This system can be used instead in order to reduce their burden. It will save a lot of effort and time on teacher's part. The human efforts applied in this repetitive task can be saved and spent more in other academic endeavors. The obvious human mistakes can be reduced to obtain an unbiased result. The system calculates the score and provides results fairly quickly. This system can be widely used in academic institutions such as schools, colleges, coaching and institutes for checking answer sheets. It can also be implemented in different organizations which conduct competitive examinations.

## **1.2 Problem Statement**

In the era of digitalization and automation educational institutions apply traditional evaluation methods, using pen and paper. The evaluators are to read through the entire answers manually which requires a lot of man power and also consumes more time. Also, the evaluation pattern may differ from person to person. To overcome the above-mentioned problems the evaluation process has to be automated using the concepts of machine learning techniques. This model aims to decrease the evaluation time and applies common evaluation pattern. Hence, this model standardizes the process of evaluation of the answer papers.

## **1.3 Objectives**

The objective of this system is as follows:

- To evaluate the internal papers without consuming too much time since online mode is faster than offline mode.
- To avoid the inconvenience and the rate of error caused by manual operation.

## **1.4 Scope**

The scope of this project is to design a system for the automatic evaluation of internal

papers. This system give marks to the answers, calculates the total score and provides results fairly quickly.

## **1.5 Organization of the Report**

- Chapter 1 of this document consists of Introduction which gives a brief description of the project and the scope of it.
- Chapter 2 of this document describes the Literature Survey. It provides details about the existing system, the limitations that the existing system experiences and the proposed system for the project.
- Chapter 3 of this document describes the Software Requirements Specification. It includes overall description and specific requirements. The overall requirement is further classified as product perspective, product functions, user classes and characteristics, design and implementation constraints, assumptions and dependencies. Specific requirements are classified as hardware requirements, software requirements, functional requirements and non-functional requirements.
- Chapter 4 is the Gantt Chart which is a bar chart showing the project schedule.
- Chapter 5 is concerned with the System Design. It includes the architectural diagram, the class diagram, the use case diagram and description, sequence diagram, activity diagram and data flow diagram.
- Chapter 6 describes the Implementation. It includes the detail description about how the project is been implemented.
- Chapter 7 describes the Testing where the proposed system is tested in various levels like unit test, integration test and system test and how the program is executed with the set of test cases.
- Chapter 8 describes the Results and Snapshots of the project.
- Chapter 9 describes the Conclusion and Future work of the project.

# **Chapter - 2**

## **LITERATURE SURVEY**

## **Chapter 2**

### **LITERATURE SURVEY**

A literature survey or a literature review in a project report is that section which shows the various analysis and research made in the field of interest and the results already published, taking into account the various parameters and the extent of the project.

#### **2.1 Existing System**

##### **2.1.1 Handwriting Recognition**

In off-line handwriting recognition, the completed writing is available as an image. This images of typed, handwritten or printed text, whether from a scanned document or a photo of a document is converted into editable text using OCR mechanism. There exist many different OCR engines like Tesseract, GOCR, Transym used in market from earlier times each having their own strengths and weaknesses.

The paper entitled, "An Overview of the Tesseract OCR Engine" provided a good reference for the capabilities and applications of the Tesseract OCR. The authors presented overview of the subject and then the experiment they took out to OCR the car license plates with an added preprocessing. This article provided relevant, useful and up-to-date information regarding Tesseract OCR and it's use. This paper is a very good source by providing a good comparative background and introduction to Tesseract.<sup>[1]</sup>

The paper entitled, "Optical Character Recognition by Open Source OCR Tool Tesseract: A Case Study" provided a comparison study of Tesseract with Transym. Transym OCR is one of the proprietary Optical Character Recognition engines. As per their study, Tesseract provides better accuracy of 61% for color image and 70% for gray scale images as compared to Transym, which provides only 47% of accuracy in the set of data. Tesseract is faster than Transym because it takes average 1 second and 0.82 seconds for processing color and gray scale images respectively to process one image, while Transym takes average 6.75 seconds to process one image.<sup>[2]</sup>

The paper entitled "Tesseract Vs GOCR: A Comparative Study" focused on comparing two tools named Tesseract and GOCR on the basis of their Accuracy and Precision by undertaking different parameters: image type, resolution and font type. <sup>[3]</sup>

It is concluded from the study by considering different types of parameters that Tesseract has better accuracy and precision than GOCR in most of the cases.

### **2.1.2 Automatic Evaluation**

"A Dynamic Semantic Space Modelling Approach for Short Essay Grading" focused on the NLP techniques such as Tokenization, Part of Speech (POS) tagging, Stemming, Stop Words Removal, Spelling mistakes and upper and lower case of words in the sentences.

Evaluation of objective test demands one specific answer from multiple answers. <sup>[4]</sup>

The paper entitled "Automatic Objective Answer-Sheet Evaluation-A OCR Based Approach" is a discussion on a solution to evaluate the objective answer-sheet automatically by computer. <sup>[5]</sup>

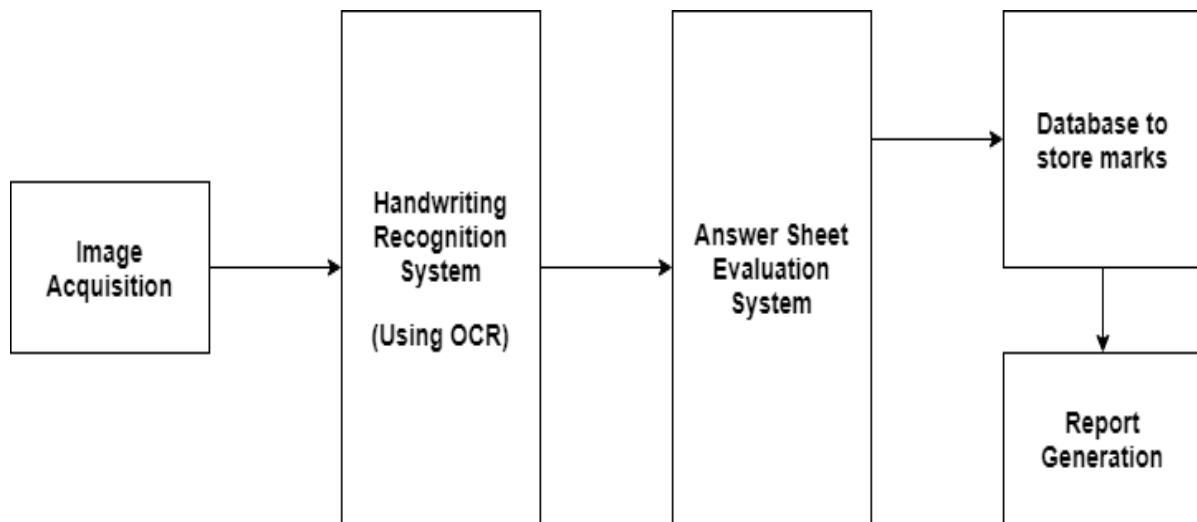
The paper entitled "An Automatic Answering System with Template Matching" presents an automatic Frequently Asked Questions (FAQ) answering system with the ability of detecting and answering questions asked in English or SMS language. System is capable of identifying spelling mistakes up to certain level (can identify where the vowels are used incorrectly). There are pre-stored question templates and answers for those template questions. The templates have to be uploaded manually to the system. When a user asks a question, the question is matched with one of the question templates. When a template is found for a given question the answer will be generated and shown to the user automatically. <sup>[6]</sup>

"Paper Rater", is a tool which checks the quality of a text body automatically. Once a document is submitted to the application it checks the content for plagiarism, correct grammar, correct spelling and vocabulary usage. Then it generates a feedback on the above mentioned features and provides a score to each component. <sup>[9]</sup>

## **2.2 Proposed System**

Examiners get bored by checking many answer sheets, hence the system reduces their workload by automating the manual checking process accurately. The system calculates the score and provides results instantly. It removes human errors that commonly occur during manual checking. The system provides an unbiased result. Thus, the system excludes human efforts and saves time and resources. This system can be used in schools, colleges for checking answer sheets. In this section, the proposed system design is described. It includes Image Acquisition phase, Handwriting Recognition phase using OCR, Answer Sheet Evaluation

phase, Database to store marks and Report Generation. Fig. 2.1 depicts the System Design of the proposed project.



**Fig. 2.1 System Design**

### **2.2.1 Image acquisition process**

Image Acquisition Digitized/Digital Image is initially taken as input. The most common of these devices is the electronic tablet or digitizer. These devices use a pen that is digital in nature. Input images for handwritten characters can also be taken by using other methods such as scanners, photographs. It is the process of document into an electronic form. This has been done with the help of scanning process. With the help of this process digital image of the document is captured. Generally, the image is used in black and white form with any format such as JPEG, BMT, and BMP etc. this image is forward to the subsequent blocks for further processing. Image acquisition is the creation of digital images. Digitization produces the digital image which is fed to the pre-processing phase.

### **2.2.2 Handwriting Recognition System**

The technique by which a computer system can recognize characters and other symbols written by hand in natural handwriting is called handwriting recognition system. In off-line handwriting recognition, the completed writing is available as an image. This images of typed, handwritten or printed text, whether from a scanned document or a photo of a document is converted into editable text using OCR mechanism.



This project presents the use of Google's open source Optical Character Recognition Software Tesseract. Tesseract is an optical character recognition engine for various operating systems. It is free software, released under the Apache License and development has been sponsored by Google since 2006. In 2006, Tesseract was considered one of the most accurate open-source OCR engines then available. Python-tesseract is an optical character recognition (OCR) tool for python. That is, it will recognize and "read" the text embedded in images.

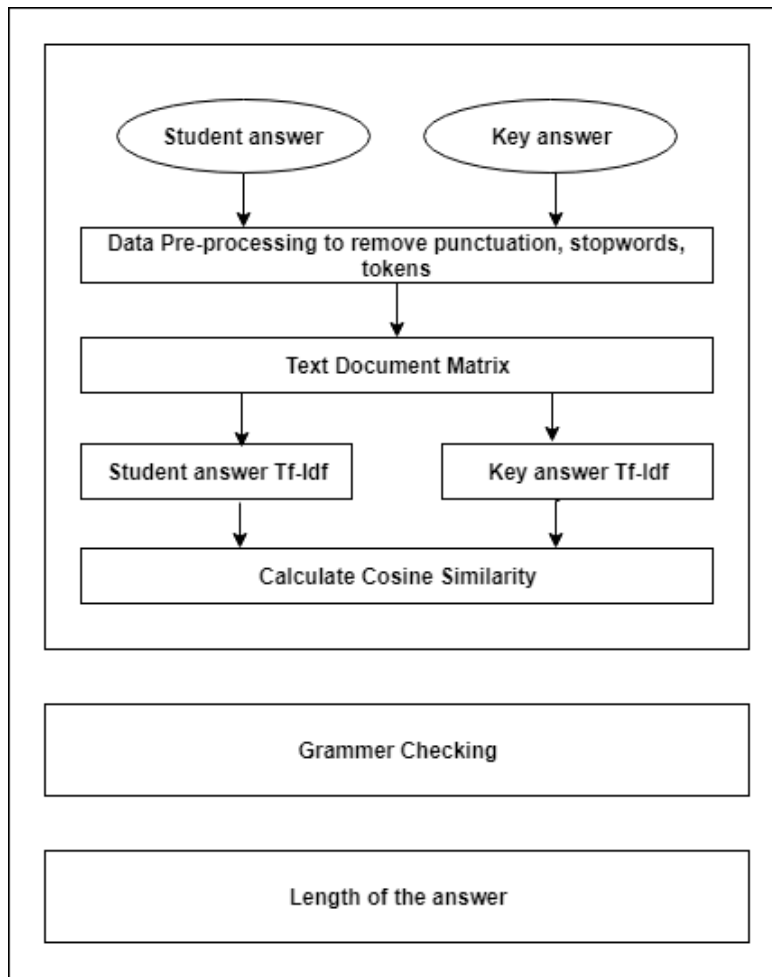
Python-tesseract is a wrapper for Google's Tesseract-OCR Engine. It is also useful as a stand-alone invocation script to tesseract, as it can read all image types supported by the Python Imaging Library, including jpeg, png, gif, bmp, tiff, and others, whereas tesseract-ocr by default only supports tiff and bmp. Additionally, Python-tesseract will write the recognized text to a file. <sup>[10]</sup>

### **2.2.3 Answer Sheet Evaluation System**

The process of converting data to something a computer can understand is referred to as preprocessing. One of the major forms of pre-processing is to filter out useless data. In natural language processing, useless words (data), are referred to as stop words. These steps are needed for transferring text from human language to machine-readable format for further processing. After a text is obtained, text normalization has to be carried out. Text normalization includes:

- Removing punctuations.
- Removing white spaces.
- Removing stop words, sparse terms, and particular words

A term document matrix is a way of representing the words in the text as a table (or matrix) of numbers. The rows of the matrix represent the text responses to be analyzed, and the columns of the matrix represent the words from the text that are to be used in the analysis. Transform the student answer and key answer into the Tf-Idf matrix. Now, compute similarity between student answer and key answer by using cosine similarity. The structure of the sentence is formed by using Grammar. The user may only write the keywords and nothing else. Hence checking whether the answer is grammatically correct or not, is important in subjective answer evaluation. Length of the answer is also an important factor as the student may write all keywords and grammatically correct short sentences. Such answer would get full marks for keywords and grammar but less marks for short length. Fig. 2.2 depicts the Answer Sheet Evaluation System of the proposed project.



**Fig. 2.2 Answer Sheet Evaluation System**

### 2.2.4 Database

The final score for the student's answer is obtained by adding marks from individual sections which are keywords, grammar and length of the answer and the calculated marks will be stored in the database.

### 2.2.5 Final Report

The answer sheet is evaluated for each student and the overall result is displayed on the user interface.

**Chapter - 3**

**SYSTEM REQUIREMENTS**

**SPECIFICATION**

## **Chapter 3**

# **SYSTEM REQUIREMENTS SPECIFICATION**

A software requirements specification (SRS) is a description of a software system to be developed. The software requirements specification lays out functional and non-functional requirements, and it may include a set of use cases that describe user interactions that the software must provide. Use cases are also known as functional requirements. In addition to use cases the SRS also contains non-functional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation. For the hardware requirements the SRS specifies the logical characteristics of each interface between the software product and the hardware components. It specifies the hardware requirements like memory restrictions, cache size, the processor, RAM size etc. those are required for the software to run. Software requirements specification is a rigorous assessment of requirements before the more specific system design stages, and its goal is to reduce later redesign. <sup>[11]</sup>

### **3.1 Overall Description**

The SRS is a document, which describes completely the external behaviour of the software. This section of the SRS describes the general factors that affect the product and its requirements. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionalities of it.

#### **3.1.1 Product Perspective**

The architecture of the product describes the passing of the control over the modules in the system. Initially the student answer script in image format is given as input. This image is further processed and converted into text file. This is evaluated based on the keywords and criteria provided before-hand and the results are displayed.

#### **3.1.2 Product Function**

The final desired product is a model that evaluates an internal paper. It takes the student's answer scripts in the image format as input, converts it to text file and compares it with the key answer provided by the faculty. Based on this match, it allocates respective marks to the student answer.

### **3.1.3 Assumptions and Dependencies**

#### **Assumptions:**

- The students handwriting should be clear and to be written in block letters.
- Students must write the programs with respect to the test cases provided to them.
- The Faculty's system is active.
- Faculty should provide appropriate keywords corresponding to the questions.
- There is a continuous internet connection.

#### **Dependencies:**

- Accuracy of the handwriting recognition depends on the clarity of image provided to the system.
- Accuracy of the evaluation depends on the valid keywords provided.

## **3.2 Specific Requirements**

This section includes the detailed description about the hardware requirements, software requirements. Functional requirements and non-functional requirements. The hardware and software requirements are:

### **3.2.1 Hardware Requirements**

- Processor: A computer with 333 MHz processor
- RAM: 512 MB RAM
- Disk: 500 MB of Hard disk space

### **3.2.2 Software Requirements**

- Operating System: Windows 98 or higher
- IDE: Jupyter Notebook
- HTML editor

### **3.2.3 Functional Requirements**

- The system must be able to interact with the user input like mouse, keyboard and menu.
- The system must allow the user to traverse from one page to another.
- The system must allow the faculty to upload multiple files at a time.

### **3.2.4 Non-functional Requirements**

- Reliability: It must make sure that the system is reliable in its operation and for allocating the marks.
- Safety requirement: Software shall not cause any harm to the human user.
- Security requirements: System shall not behave abruptly or corrupt the project file on receiving wrong input from the user.
- Performance requirement: The program reacts to the input quickly as specified.

# **Chapter - 4**

## **GANTT CHART**

## CHAPTER 4

### GANTT CHART

A Gantt chart is a type of bar chart, developed by Henry Gantt that illustrates a project schedule. Gantt charts illustrate the start and finish of the terminal elements and summary elements of the project. Terminal elements and summary elements comprise the work breakdown structure of the project.

The following is the Gantt chart of the project “AUTOMATED EVALUATION OF INTERNAL PAPER”.

**Table 4.1: Gantt chart of planning and scheduling of project**

Number	Task	Start	End	Duration(days)
1	Synopsis	28-Aug-2018	17-Sep-2018	21
2	Presentation on idea	20-Sept-2018	5-Oct-2018	15
3	Software Requirement Specification	7-Oct-2018	15-Oct-2018	8
4	System Design	16- Oct -2018	28- Oct -2018	11
5	Implementation	5-Jan-2019	20-Feb-2019	50
6	Presentation on work progress	21-Feb-2019	25-Feb-2019	4
7	Testing	21-Feb-2019	28-Feb-2019	7
8	Result and Report	1-Mar-2019	14-Mar-2019	14



ACTIVITY/ MONTH	AUG	SEP	OCT	NOV	DEC	JAN	FEB	MAR
SYNOPSIS								
PRESENTATION ON IDEA								
SRS								
DESIGN								
IMPLEMENTATION								
TESTING								
REPORT								

Fig. 4.1: Gantt chart

# **Chapter - 5**

## **SYSTEM DESIGN**

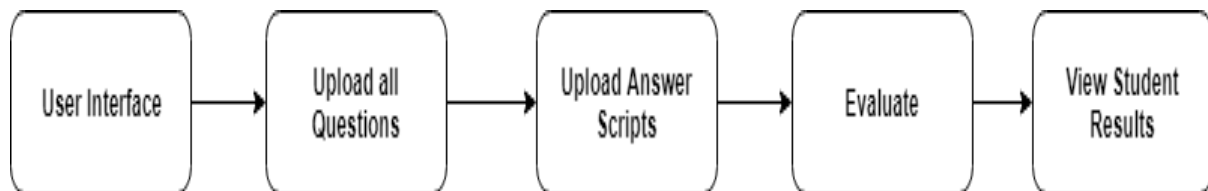
## Chapter 5

### SYSTEM DESIGN

System overview provides a top-level view of the entire software product. It highlights the major components without taking account the inner details of the implementation. It describes the functionality of the product and context and design of the software product. The application will be developed in a way which provides the user to interact with the system and simplifies the tasks by providing the smooth user interface and user experience with easily readable and understandable view.

#### 5.1 Architectural Diagram

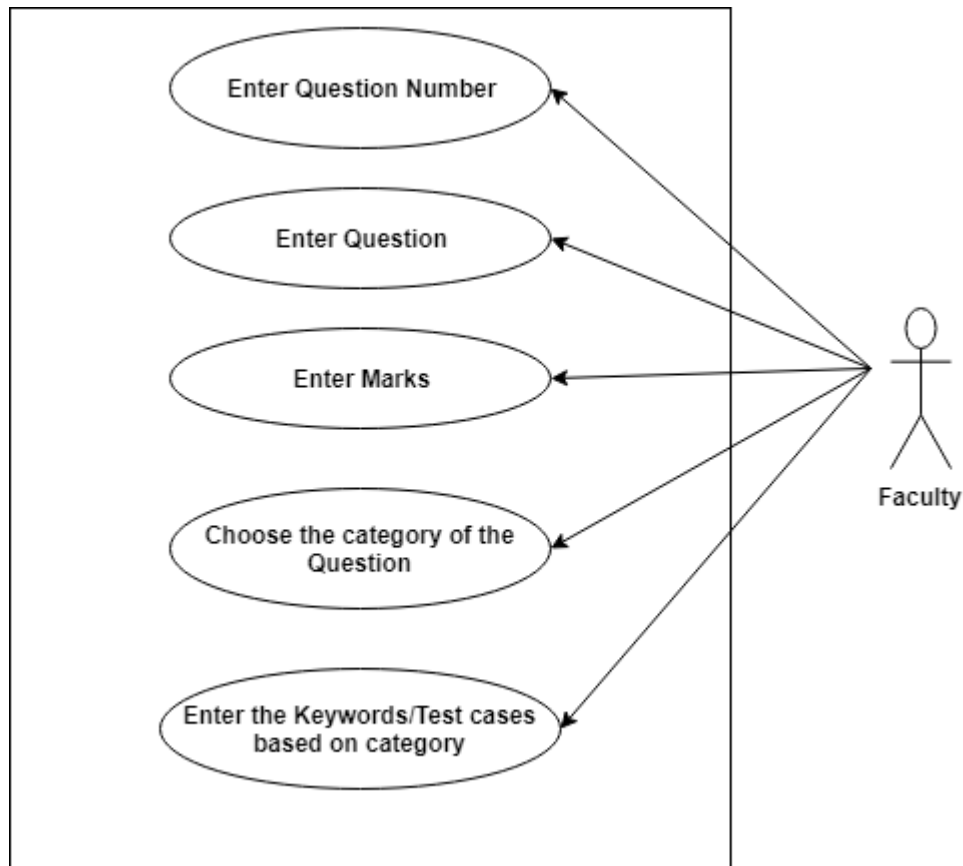
The architectural design gives the description about the overall system design. It is specified by identifying the components defining the control and data flow between them. The arrow indicates the connection and rectangular box represents the functional units. The Fig. 5.1 shows the architectural diagram of this project which shows the overall operation from uploading question papers to the evaluation of the student answer scripts.



**Fig. 5.1: Architectural Design Diagram**

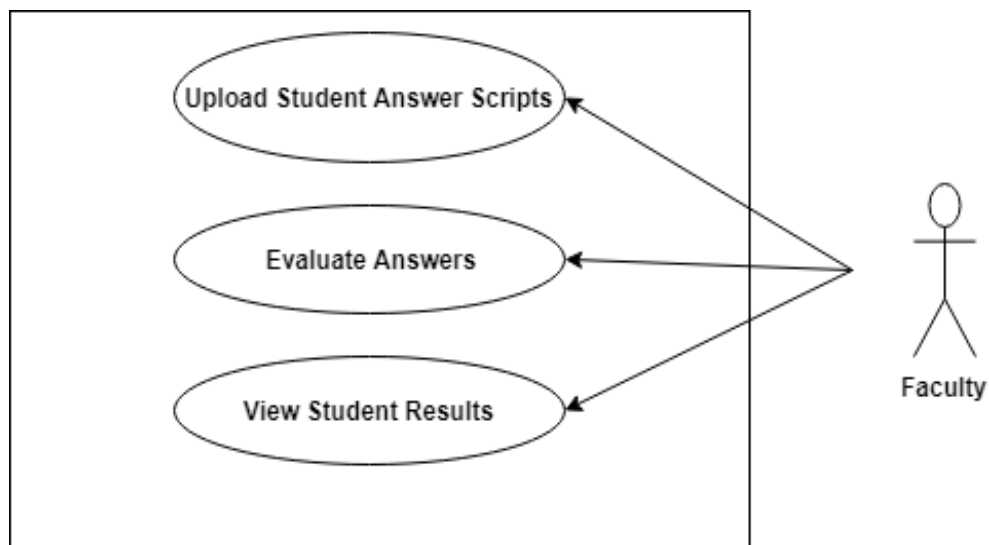
#### 5.2 Use case Diagram and Description

A use case is a coherent piece of functionality that a system provides interacting with actors. It describes a system which involves a set of use cases and a set of actors. The Fig. 5.2(a) shows the use case of the uploading question paper phase where the faculty has to enter the question related details.



**Fig. 5.2(a): Use case diagram for Question paper**

The Fig. 5.2(b) shows the use case of the evaluation process where the student answer scripts are evaluated based on the criteria provided and displays the result.



**Fig. 5.2(b): Use case diagram for Evaluation**

### 5.3 Sequence Diagram

A sequence diagram shows how a set of objects communicate with each other to complete a complex task. The Fig. 5.3 shows the sequence of operations between the different modules involved in the project.

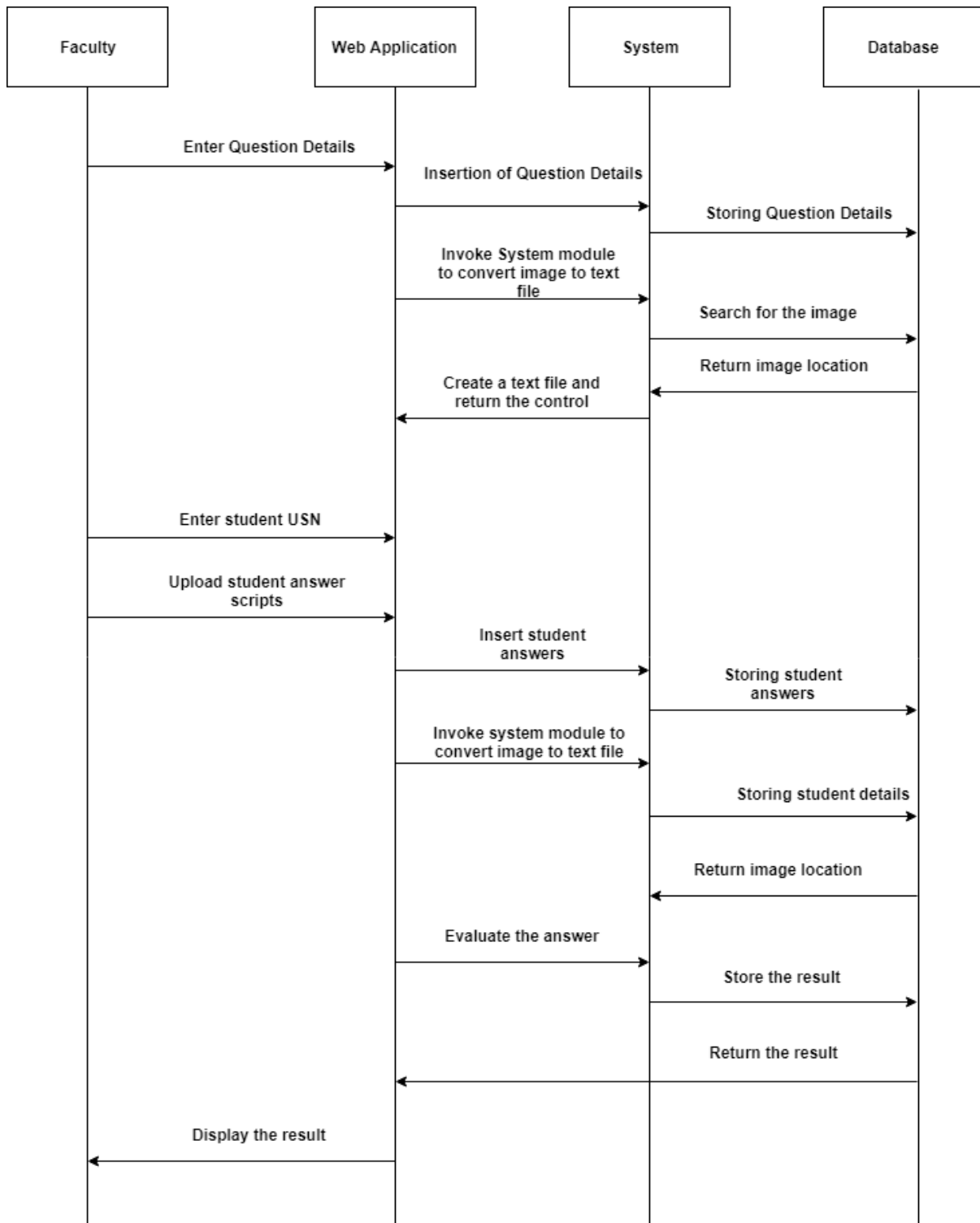
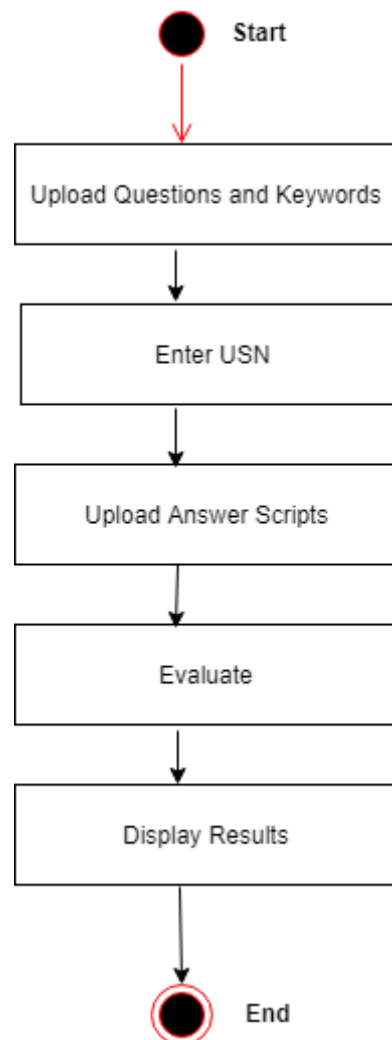


Fig. 5.3: Sequence Diagram

## 5.4 Activity Diagram

Activity Diagram shows the sequence of steps that make up complex process. It shows the flow of control, similar to sequence but focuses on operations rather than on objects. The components used in this are as follows:

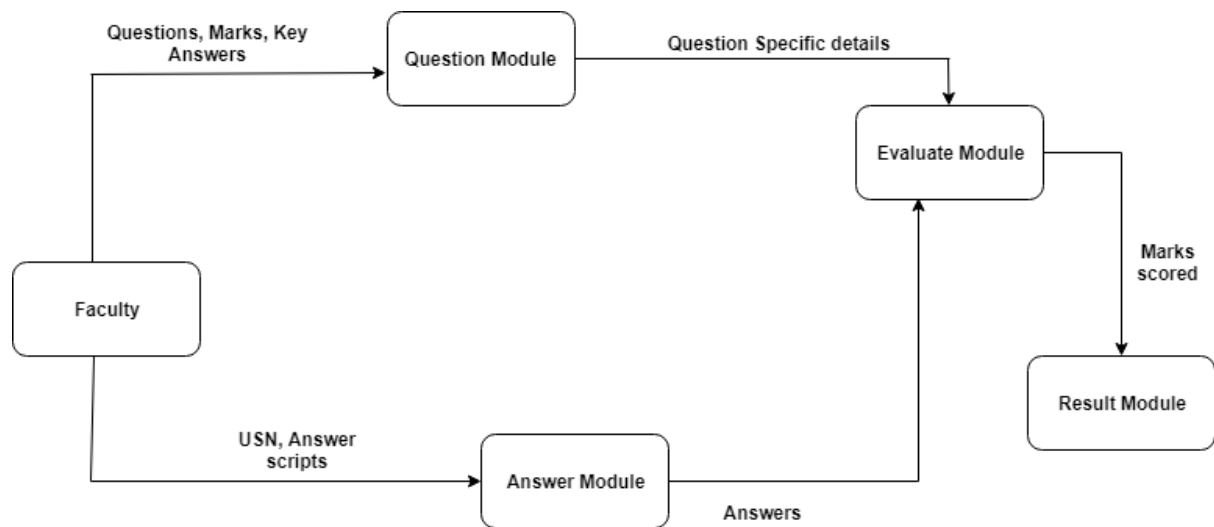
- Rounded Rectangle indicates the process
- Arrow indicates the transition line
- Rhombus indicates the decision.
- Bars represents the start or end concurrent activities
- Solid circle represents the initial state of work flow
- Encircled black circle represents the final state of work flow.



**Fig. 5.4 Activity Diagram**

## 5.5 Data Flow Diagram

A data flow diagram is a graphical representation of the “flow” of data through an information system, modelling its process aspects. It is often used as preliminary step to create an overview of the system, which can later be elaborated.



**Fig. 5.5 Data Flow Diagram**

The Fig. 5.5 shows the data flow between each component in the system. Faculty uploads the question papers and key answers to the Question module. Faculty also uploads the answer scripts to the answer module. These processed answer scripts are given to the evaluate module. Then the results are displayed in the result module.

# **Chapter - 6**

## **IMPLEMENTATION**



## Chapter 6

### IMPLEMENTATION

Implementation is the core step in software development life cycle. Implementation gives the detailed view of the project and describes the pseudo code and various important functions in the project.

#### 6.1 Algorithm

An algorithm is a step-by-step instruction to execute the program. The following Algorithm 6.1 shows the high-level algorithm of the project.

**Algorithm 6.1: High-level algorithm of the project**

*Step 1: Start*

*Step 2: Upload a question to the system*

- 1. Input the question number*
- 2. Enter the question*
- 3. Enter the total marks*
- 4. Choose the question category*
  - a. Theory*
    - *Enter the important keywords*
    - *Upload the reference answer in image format*
  - b. C Program*
    - *Enter 3 test cases*

*Step3: Update question details into the database and convert reference answer from image to text*

*Step 4: Repeat Step 2 to 3 for all the questions*

*Step 5: Upload student answer script in image format*

*Step 6: Convert student answer from image to text*

*Step7: Evaluate student answer and Update the marks*

*Step8: Stop*

## 6.2 Code Snippets

### 6.2.1 Connecting to a database

The following code snippet is used to connect the user interface to the database.

```
<?php
error_reporting(1);
$con=mysqli_connect("localhost","root","")
or
die ('I cannot connect to the database because: ' .
mysql_error());
mysqli_select_db ($con,'aep');
?>
```

### 6.2.2 Upload questions

The following code snippet is used to upload the question number, question, maximum marks, type of the question and reference answer to the system.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>AEIP</title>
<form action="upload.php" method="post"
enctype="multipart/form-data" class="booking-form">
...
<label for="#">KEYWORDS</label>
<input type="text" name="keywp" id="keyw1" class="form-control
hide_input" placeholder="Important Keywords">
...
</html>
```

### 6.2.3 Upload answer scripts

The following code snippet is used to upload the student answer script into the system.

```
<!DOCTYPE html>

<form action="uploadanswer.php" method="post"
enctype="multipart/form-data" class="booking-form">

<div class="wrap">

  <label for="#">Answer for Q1</label>

  <input type="file" name="ans1p[]" id="text1" class="form-
inline,form-control" accept="image/*" multiple/>

</div>

...

<div class="wrap">

<label for="#">Answer for Q6</label>

<input type="file" name="ans6p[]" id="text1" class="form-
inline,form-control" accept="image/*" multiple/>

</div>

<input type="submit" name="answer1" value="Submit Answer"
class="btn btn-primary py-3 px-4 align-self-stretch">

</form>

</html>
```

### 6.2.4 Convert image into text

The following code is used to convert the image into text.

```
import sys
quesno=sys.argv[1]
import pymysql
db = pymysql.connect("localhost","root","","aep" )
cursor = db.cursor()
cursor.execute("SELECT * from question where qno=%s",quesno)
data = cursor.fetchall()
...
def ocr(file_to_ocr):
    im = Image.open(os.path.join(file_to_ocr))
    ...
    return txt
...
for i in range(index):
    image_path=path1+a[i]
    txt=ocr(image_path)
    file_to_open=directory+"\\ "+fname+".txt"
    if (os.path.isfile(file_to_open)):
        with open(file_to_open,'a') as f:
            f.write(str(txt))
    else:
        with open(file_to_open,'w') as f:
            f.write(str(txt))
```

### 6.2.5 Preprocess answer script

The following code is used to preprocess the student answer.

```
def PreprocessingTheSentence(stmt1):  
    stmt1=stmt1.lower()  
    import string  
  
    remove=dict.fromkeys(map(ord, '\n'+string.punctuation), " ")  
    PSA1=stmt1.translate(remove)  
    ...  
    word_tokens = word_tokenize(PSA1)  
    PSA2 = [w for w in word_tokens if not w in stop_words]  
    ...  
    for i in range(len(PSA2)):  
        a=list(nltk.pos_tag([PSA2[i]]))  
        if a[0][1]!='IN':  
            b.append(PSA2[i])  
    PSA3=" ".join(b)  
    return PSA3
```

### 6.2.6 Evaluate answers of category theory

The following code is used to evaluate the student answers of category theory.

```
import os

...

if(os.path.isdir(path1) ==True):

    ...

    f1=path1+"\\ "+str(list_ans[i])
    if(os.path.isfile(f1)):
        num_words=0
        with open(f1, 'r') as f:
            for line in f:
                words = line.split()
                num_words += len(words)
    f1=path1+"\\ "+str(list_ans[i])
    file_ans = open(f1, 'r')
    text=' '.join(file_ans.readlines())
    import re
    sentences=re.split(r' *[\.\?!][\(\)"\)\ \]]* ',text)
    ans_length=len(sentences)-1
    if ans_length<=0:
        ans_length=1
    T1=ansLen(num_words,ans_length)
    T2=grammarCheck(Sans,num_words,ans_length)
    T3=keyAnsMatch(Sans,Kans)
    T4=textBookMatch(Sans,Tans)
    marks_for_each_question(SUSN,T1,T2,T3,T4)
```

### 6.2.7 Evaluate answers of category c program

The following code is used to evaluate the student answers of category c program.

```
def evaluateCprogram(quesno2,SUSN,filename):
    ffl=open('a1.c','w')
        ffl.truncate(0)
    with open(f1) as myf:
        for num,line in enumerate(myf,1):
            ffl=open('a1.c','a+')
            ffl.write(line)
    ...
    with open('a1.c') as myFile:
        for num,line in enumerate(myFile,1):
            if lookup in line:
                line_numbers_list1.append(num)
                str1=line
                n=str1.find("&")+1
                i=find_variable(n,num)
                while str1[i]!=")":
                    if str1[i]=="," and str1[i+1]=="&":
                        n=i+2
                        i=find_variable(n,num)
    ...
    import subprocess
    subprocess.getoutput(["g++","z1.c"])
    ...
    marksss=math.ceil((count/num_lines)*5)
    listc[xyz-1]=marksss
```

## 6.2.8 Display results

The following code is used to view the student result.

```
<tbody>
  <?php
$sql = "SELECT * FROM answer";
$result = mysqli_query($con, $sql);
  if (mysqli_num_rows($result) > 0)
    {
      while($row = mysqli_fetch_assoc($result))
        {
          ?>
<tr>
  <td><?php echo $row['USN'];?></td>
  <td><?php echo $row['m1'];?></td>
  ...
  <td><?php echo $row['m6'];?></td>
  <td><?php echo $row['total'];?></td>
  ...
<td colspan="8" style="text-align:center;">NO RESULTS</td>
  </tr>
  <?php
  }?>

</tbody>
</table>
```



# **Chapter - 7**

## **TESTING**

## **Chapter 7**

# **TESTING**

Testing is an activity to check whether the actual results match the expected results. Testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements. Testing is an important phase in the development life cycle of the product. During the testing, the program to be tested was executed with a set of test cases and the output of the program for the test cases was evaluated to determine whether the program is performing as expected. Errors were found and corrected by using the following testing steps and correction was recorded for future references. Thus, a series of testing was performed on the system before it was ready for implementation. An important point is that software testing should be distinguished from the separate discipline of Software Quality Assurance (SQA), which encompasses all business process areas, not just testing.

### **7.1 Testing Levels**

Testing is part of Verification and Validation. Testing plays a very critical role for quality assurance and for ensuring the reliability of the software.

The objective of testing can be stated in the following ways.

- A successful test is one that uncovers as-yet-undiscovered bugs.
- A better test case has high probability of finding un-noticed bugs.
- A pessimistic approach of running the software with the intent of finding errors.

Testing can be performed in various levels like unit test, integration test and system test.

#### **7.1.1 Unit Testing**

Unit testing tests the individual components to ensure that they operate correctly. Each component is tested independently, without other system component. This system was tested with the set of proper test data for each module and the results were checked with the expected output. Unit testing focuses on verification effort on the smallest unit of the software design module.

- After every module of the project is prepared debugging is carried out to ensure its proper functioning.
- A testing is carried out to check whether the database is connected to the user interface.
- Testing was done on question uploading unit to check whether the question specific

details like question number, question, maximum marks, question category, key answers and image of the reference answer are stored in the local storage.

- A test was carried out on handwriting recognition module to verify whether the images are getting converted into text.
- Testing was carried out on different units of this system like checking the grammar mistakes of student answers, finding the length of the answer, evaluating the similarity between student answer and reference answer, searching for important keywords in student answer.

### **7.1.2 Integration Testing**

Integration testing is another aspect of testing that is generally done in order to uncover errors associated with the flow of data across interfaces. The unit-tested modules are grouped together and tested in small segment, which makes it easier to isolate and correct errors. This approach is continued until we have integrated all modules to form the system as a whole.

- After the completion of each module it has been combined with the remaining module to ensure that the project is working properly as expected.
- An integration test was performed on evaluation module whether the student answer scripts are uploaded and evaluated, which consist of different units like image to text unit, a unit to preprocess the student answer, a program to check the grammar and find the length of the student answer, a unit to estimate the similarity between student answer and reference answer to check.

### **7.1.3 System Testing**

System testing tests a completely integrated system to verify that it meets its requirements. After the completion of all the module they are combined together to test whether the entire project is working properly.

- It deals with testing the whole project for its intended purpose. In other words, the whole system is tested here.
- System testing was carried out by uploading questions' details through user interface to the system and then student answer scripts were uploaded and evaluated. Then the system generated students' result which is accessed by the faculty through the user interface.

### 7.1.4 Acceptance Testing

Project is tested at different levels to ensure that it is working properly and was meeting the requirements which are specified in the requirement analysis.

- Acceptance testing is done once the project is done and checked for the acceptance by uploading student answer scripts to the system for automatic evaluation
- The results from the system was compared with the results from the traditional evaluation approach.
- Then the system was tested in terms of accuracy and throughput against traditional approach.

## 7.2 Test Cases

A test case is a software testing document, which consists of events, action, input, output, expected result and actual result. Technically a test case includes test description, procedure, expected result and remarks. Test cases should be based primarily on the software requirements and developed to verify correct functionality and to establish conditions that reveal potential errors.

Individual PASS/FAIL criteria are written for each test case. All the tests need to get a PASS result for proper working of an application.

### Test Case 1: Uploading questions into system

**Objective:** To insert question number, questions, maximum marks, question type, and keywords into the system.

**Steps:** The following steps have to be followed to carry out the test.

1. Enter the question number within the range 1 to 6.
2. Enter the question, maximum marks, type of the question.
3. If the question is of type Theory then upload the key answer in image format and keywords.
4. If the question is of type is Program then upload 3 test cases.
5. Click on submit button.

**Expected Results:** Question specific details are stored in the database.

**Result:** Successful.

## **Test Case 2: Uploading answer script into system**

**Objective:** To upload scanned copies of student answer scripts.

**Steps:** The following steps have to be followed to carry out the test.

1. Enter student's USN.
2. Select scanned answers for respective questions.
3. Click on Evaluate.

**Expected Results:** Student answer has to be uploaded to the database.

**Result:** Successful.

## **Test Case 3: Converting image to text**

**Objective:** To convert scanned copies of student answer scripts into text.

**Steps:** The following steps have to be followed to carry out the test.

1. Upload the scanned copy of the answer script.
2. Click on Evaluate.

**Expected Results:** The image has to be converted into a text file.

**Result:** Successful.

## **Test Case 4: Finding similarity between student answer and key answer**

**Objective:** To calculate the percentage of similarity between answer and keywords.

**Steps:** The following steps have to be followed to carry out the test.

1. Upload the question details to the system.
2. Upload student answer script into the system.
3. Click on Evaluate.

**Expected Results:** The similarity between answer and key answer is calculated out of 100.

**Result:** Successful.

## **Test Case 5: Display the student result**

**Objective:** To display student result.

**Steps:** The following steps have to be followed to carry out the test.

1. Fetch the data from the database.
2. Display the student mark details.

**Expected Results:** To display the student results.

**Result:** Successful.

### **Test Case 6: Clear answer script from the database**

**Objective:** To delete the student answer.

**Steps:** The following steps have to be followed to carry out the test.

1. Enter student USN.
2. Click on Clear all previous data.

**Expected Results:** To delete previous data.

**Result:** Successful.

### **Test Case 7: Delete questions from the system**

**Objective:** To delete questions from the database.

**Steps:** The following steps have to be followed to carry out the test.

1. Click on Instructor menu.
2. Press on Clear all previous data.

**Expected Results:** To delete the questions from the system.

**Result:** Successful.

The above-mentioned test Cases are summarized as shown in Table 7.1.

**Table 7.1 Test Cases**

<b>Test Numbers</b>	<b>Test Case ID</b>	<b>Test Case</b>	<b>Expected Results</b>	<b>Status</b>
1	UT_1	Uploading questions into system	Question specific details are stored in the database	Uploaded
2	UT_2	Uploading answer script into system	Student answer has to be uploaded to the database	Uploaded
3	UT_3	Converting image to text	The image has to be converted into a text file	Converted
4	UT_4	Finding similarity between student answer and key answer	The similarity between answer and key answer is calculated out of 100	Evaluated
5	UT_5	Display the student result	To display the student results	Displayed
6	UT_6	Clear answer script from the database	To delete previous data	Deleted
7	UT-7	Delete questions from the system	To delete the questions from the system	Deleted

# **Chapter - 8**

## **RESULTS AND SNAPSHOTS**



## Chapter 8

# RESULTS AND SNAPSHOTS

### 8.1 Snapshots

Fig. 8.1 shows the homepage of the project.

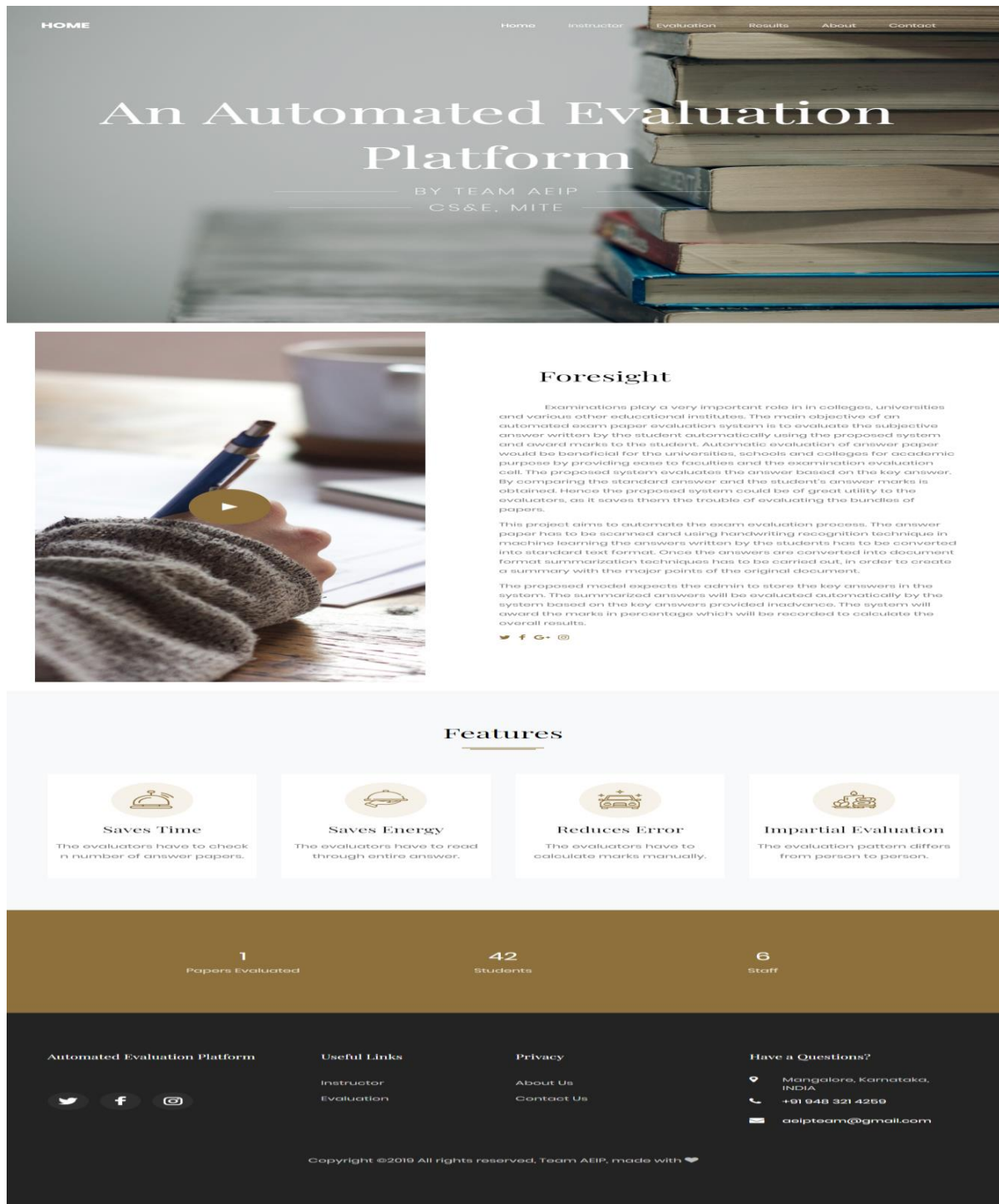


Fig. 8.1 Homepage

Fig. 8.2 shows a website page from where the question paper along with the key answers are uploaded by the faculty.

The screenshot shows the 'Instructor' page of the Automated Evaluation Platform (AEIP). The page has a header with the AEIP logo and navigation links: Home, Instructor, Evaluation, Results, About, and Contact. Below the header, the word 'Instructor' is prominently displayed. The main heading is 'Add the Questions with Key answers', followed by a 'Clear all previous data.' button. The form contains four input fields: 'QUESTION NUMBER' (with placeholder 'Enter the Qustion No.'), 'QUESTION' (with placeholder 'Enter the Question...'), 'MARKS' (with placeholder 'Enter the Marks...'), and 'CHOOSE QUESTION 1 CATEGORY' (with radio buttons for 'THEORY' and 'PROGRAM'). An 'Add the Question' button is at the bottom of the form. The footer includes social media icons, a 'Useful Links' section with 'Instructor' and 'Evaluation' links, a 'Privacy' section with 'About Us' and 'Contact Us' links, and contact information for Mangalore, Karnataka, India, including a phone number and email address. A copyright notice for 2019 is also present.

AEIP

Home Instructor Evaluation Results About Contact

HOME INSTRUCTOR

# Instructor

## Add the Questions with Key answers

Clear all previous data.

QUESTION NUMBER  
Enter the Qustion No.

QUESTION  
Enter the Question...

MARKS  
Enter the Marks...

CHOOSE QUESTION 1 CATEGORY

☐ THEORY

☐ PROGRAM

Add the Question

Automated Evaluation Platform

Useful Links

Instructor

Evaluation

Privacy

About Us

Contact Us

Have a Questions?

Mangalore, Karnataka, INDIA

+91 948 321 4259

aeipteam@gmail.com

Copyright ©2019 All rights reserved, Team AEIP, made with ♥

Fig. 8.2 Webpage to add questions and key answers

Fig. 8.3 shows a website page from where the student answer sheets are uploaded by the faculty.

AEIP

Home Instructor Evaluation Results About Contact

HOME EVALUATION

# Evaluation

## Add Answer Sheets

Clear all previous data.

STUDENT DATA

Enter Student USN

ANSWER FOR Q1

Choose Files No file chosen

ANSWER FOR Q2

Choose Files No file chosen

ANSWER FOR Q3

Choose Files No file chosen

ANSWER FOR Q4

Choose Files No file chosen

ANSWER FOR Q5

Choose Files No file chosen

ANSWER FOR Q6

Choose Files No file chosen

Submit Answer

Automated Evaluation Platform

Useful Links

Privacy

Have a Questions?

Mangalore, Karnataka, INDIA

+91 948 321 4259

aelpteam@gmail.com

Copyright ©2019 All rights reserved, Team AEIP, made with ♥

Fig. 8.3 Webpage to add student answer sheets

Fig. 8.4 depicts how the faculty can add questions belonging to the category “theory”.

The screenshot shows the 'Instructor' page of the Automated Evaluation Platform (AEIP). The page has a header with navigation links: Home, Instructor, Evaluation, Results, About, and Contact. Below the header, the title 'Instructor' is displayed. The main heading is 'Add the Questions with Key answers'. A button labeled 'Clear all previous data.' is present. The form contains the following fields:

- QUESTION NUMBER:** 1
- QUESTION:** Explain the concept of a File?
- MARKS:** 10
- CHOOSE QUESTION 1 CATEGORY:** ☒ THEORY
- KEYWORDS:** Collection of information recorded on secondary storage. File is referred by its name. Attributes of file name,type,location
- TEXT BOOK:**  scan0004.jpg
- PROGRAM:** (empty)

An 'Add the Question' button is located at the bottom of the form. The footer contains contact information for the Automated Evaluation Platform, including social media links, useful links (Instructor, Evaluation), privacy information, and contact details (Mangalore, Karnataka, INDIA; +91 948 321 4259; aeipteam@gmail.com). The footer also includes a copyright notice: Copyright ©2019 All rights reserved, Team AEIP, made with ♥.

**Fig. 8.4 Add a question and key answer belonging to “theory” category**

Fig. 8.5 depicts how the faculty can add questions belonging to the category “program” with three different testcases.

The screenshot shows the 'Instructor' dashboard of the Automated Evaluation Platform (AEIP). The main heading is 'Add the Questions with Key answers'. Below this is a 'Clear all previous data.' button. The form contains the following fields:

- QUESTION NUMBER:** 1
- QUESTION:** Write a C program to check palindrome.
- MARKS:** 10
- CHOOSE QUESTION 1 CATEGORY:**
  - ☐ THEORY
  - ☒ PROGRAM

Below the category selection, there is a text area for 'EXAMPLE TEST CASE (SAMPLE.TXT)' containing the following text:

```

Enter two character
Enter value
Enter the four numbers
1,2,3,4
Enter the four numbers
Enter the number of elements
0,1,2,3,4,5,6,7,8,9
The character is a
The character is
b
m=1.000000
  
```

Below the text area, there are three test case sections:

- TEST CASE 1:** Choose Files p1.jpg
- TEST CASE 2:** Choose Files P\_20190301\_140033.jpg
- TEST CASE 3:** Choose Files scan0002.jpg

At the bottom of the form is an 'Add the Question' button. The footer of the page contains contact information and social media links for AEIP.

**Fig. 8.5 Add a question and 3 different testcases belonging to “program” category**

Fig. 8.6 depicts how the faculty can upload answer sheets of each student by specifying student's USN.

AEIP Home Instructor Evaluation Results About Contact

HOME EVALUATION

# Evaluation

## Add Answer Sheets

Clear all previous data.

STUDENT DATA  
4MT15CS001

ANSWER FOR Q1  
Choose Files P\_20190301\_140102.jpg

ANSWER FOR Q2  
Choose Files t.jpg

ANSWER FOR Q3  
Choose Files p2.jpg

ANSWER FOR Q4  
Choose Files P\_20190301\_140033.jpg

ANSWER FOR Q5  
Choose Files New Doc 2019-...4516\_3.jpg

ANSWER FOR Q6  
Choose Files New Doc 2019-...4516\_1.jpg

Submit Answer

Automated Evaluation Platform

Useful Links  
Instructor  
Evaluation

Privacy  
About Us  
Contact Us

Have a Questions?  
Mangalore, Karnataka, INDIA  
+91 948 321 4259  
aeipteam@gmail.com

Copyright ©2019 All rights reserved, Team AEIP, made with ♥

**Fig. 8.6 Uploading answer sheets of each student by specifying their USN**

Fig. 8.7 shows the results page after evaluation of all students.

AEIP

Home Instructor Evaluation Results About Contact

HOME INSTRUCTOR

# Instructor

## Student Marks

USN	MARKS						TOTAL
	QUESTION 1	QUESTION 2	QUESTION 3	QUESTION 4	QUESTION 5	QUESTION 6	
4MT15CS003	6.2	0	0	0	0	0	6.2
4MT15CS002	5	0	0	0	0	0	5
4MT15CS001	9	0	0	0	0	0	9
4MT15CS004	0	10	0	0	0	0	10
4MT15CS005	0	10	0	0	0	0	10
4MT15CS006	0	0	0	0	0	0	0

Automated Evaluation Platform

Useful Links

Instructor

Evaluation

Privacy

About Us

Contact Us

Have a Questions?

📍 Mangalore, Karnataka, INDIA

📞 +91 948 321 4259

✉️ aeipteam@gmail.com

Copyright ©2019 All rights reserved, Team AEIP, made with ❤️

**Fig. 8.7 Students result page**

**Chapter - 9**  
**CONCLUSION AND**  
**FUTURE WORK**



## **Chapter 9**

# **CONCLUSION AND FUTURE WORK**

### **9.1 Conclusion**

This system attempts to evaluate the subjective answers. The model calculates the student's answer based on the key answer provided by the university. By judging against the model answer and the student's answer marks are allocated to the student. Thus, this system could be of great effectiveness to the educational institutes, as it saves time and the trouble of checking bundles of papers. This evaluation system will grade the answers depending upon the percentage of key answer matched.

### **9.2 Future Work**

Future work will involve developing an assessment system to address automatic evaluation of formulas, programs written in different programming languages and graphs of student answers with improved performance and throughput.

# REFERENCES

## Journals:

- [1] Ray Smith, Google Inc., “An Overview of the Tesseract OCR Engine”
- [2] Chirag Patel, Atul Patel and Dharmendra Patel, “Optical Character Recognition by Open Source OCR Tool Tesseract: A Case Study”, International Journal of Computer Applications (0975 – 8887) Volume 55– No.10, October 2012
- [3] Shivani Dhiman and A.J Singh, “Tesseract Vs Gocr A Comparative Study”, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-2, Issue-4, September 2013.
- [4] G. R. Perera, D. N. Perera and A. R. Weerasinghe, “A dynamic semantic space modelling approach for short essay grading,” 2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer), Colombo, 2015, pp. 43-49
- [5] M. Jangid and S. Srivastava, “Automatic Objective Answer-Sheet Evaluation-A OCR Based Approach1,” 2013 IEEE International Conference in MOOC, Innovation and Technology in Education (MITE), Jaipur, 2013, pp. 224-227
- [6] T. Gunawardena, M. Lokuhetti, N. Pathirana, R. Ragel and S. Deegalla, “An automatic answering system with template matching for natural language questions,” 2010 Fifth International Conference on Information and Automation for Sustainability, Colombo, 2010, pp. 353-358

## Websites:

- [7] [https://en.wikipedia.org/wiki/Natural\\_language\\_processing](https://en.wikipedia.org/wiki/Natural_language_processing)
- [8] <https://pypi.org/project/pytesseract/>
- [9] <https://www.paperrater.com>
- [10] <https://www.pyimagesearch.com/2017/07/10/using-tesseract-ocr-python>
- [11] [https://en.wikipedia.org/wiki/Software\\_requirements\\_specification](https://en.wikipedia.org/wiki/Software_requirements_specification)