Capgemini

PROJECT TITLE: QUIZ EVALUATION SOFTWARE

High Level Design & Low Level Design

Employee ID	Name
46281983	Shaik Sameera
46281977	Sai Deepthi Vagu
46282011	Bhavani Nimmakayala
46282010	Venkata Radha Golla
46281056	Ramya Sri Mogilisetti

Document Control :

Project Revision History

Date	Version	Author	Brief Description of Changes	Approver Signature
21-11-2022	0.1	Group 6		
25-11-2022	0.2	Group 6		

1. Introduction

The aim of this document is to gather, analyze and give an in-depth insight into the complete Quiz Evaluation Software by defining the problem statement in detail. The intended audience includes all teachers and students in the potential system. These include teaching staff and Students. The detailed low-level design of the Quiz Evaluation Software is provided in this document.

1.1. Intended Audience

This project is useful for trainer and institutes.

1.2. Definition

This project will evaluate the quiz score and generate a CSV file for final report.

1.3. Project Purpose

Quiz-building software helps to create assessments and high-level checks of students' performance faster and easier. It plays a significant role in thigh-level flow/especially the n the e-learning domain because it offers a series of benefits to learners and trainers both.

1.3.1. In Scope

A quiz evaluation system can be used in private institutes as well as educational institutions. It is a user-friendly web base application that can be used anywhere and anytime. Every software may have some cases of bugs, errors, security-related problems, or system faults.

1.4. Functional Overview

This HLD Document is arranged in the following format :
-Section1 : Introduction
A brief explanation about the purpose, aim, scope and design format of the proposed system.
- Section 2 : General Description
This section is all about the assumptions and design aspect associated with the proposed system .The product perspective will gives an overall description of the system.
- Section 3 : Design Details
This section documents the detailed design of all modules associated with the development of the proposed system.

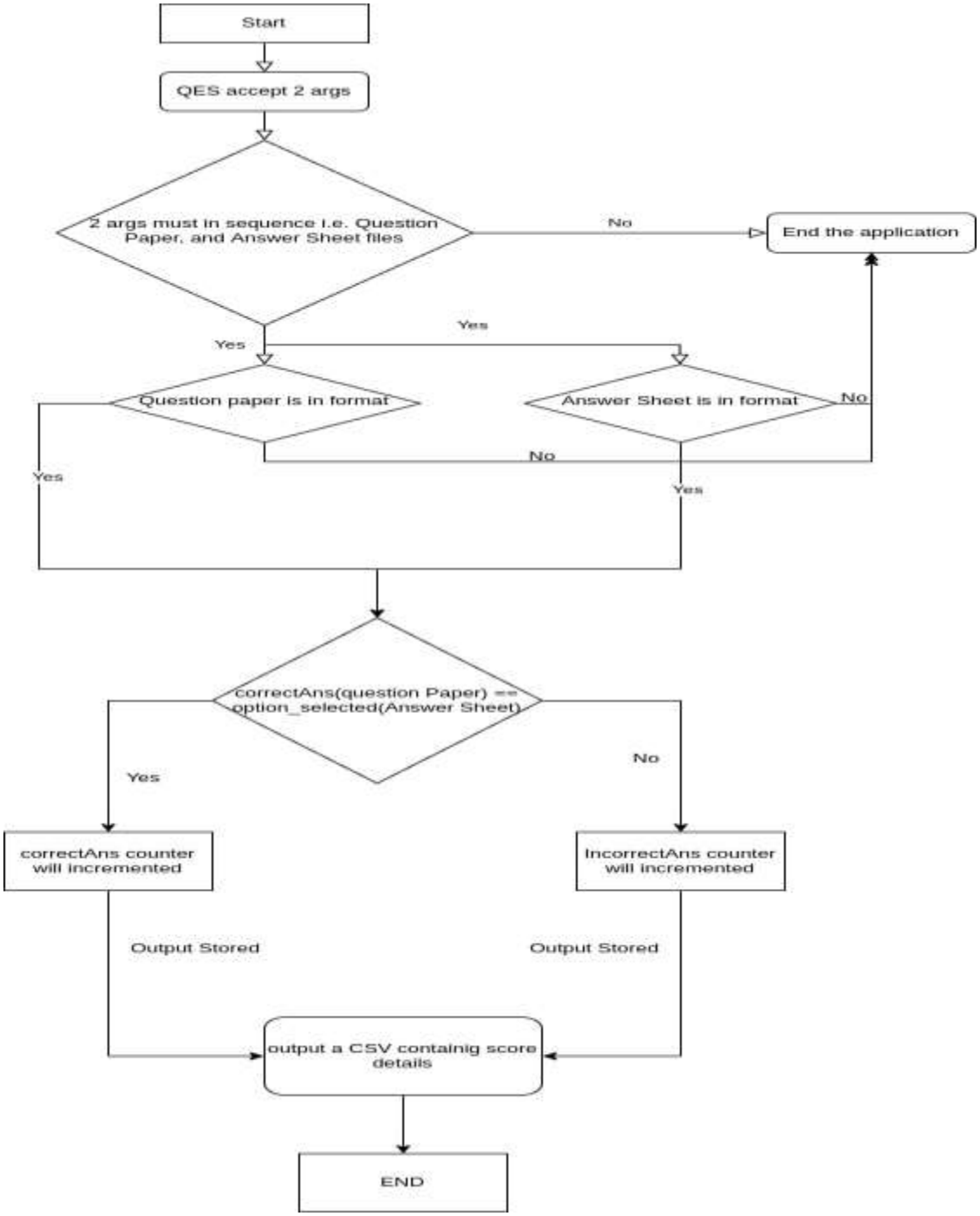
2. Design Overview

2.1. Design Objectives

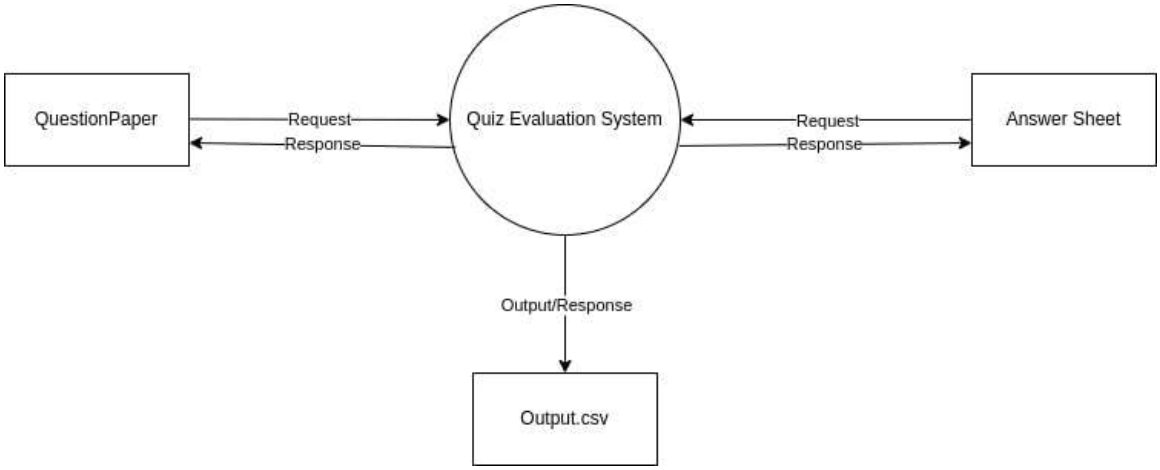
The main design features include four major parts: architecture, process relation, and automation. In order to make these designs easier to understand, the design has been illustrated in the attached diagrams (Class, Data flow & Flowchart diagrams).

3. System Architecture

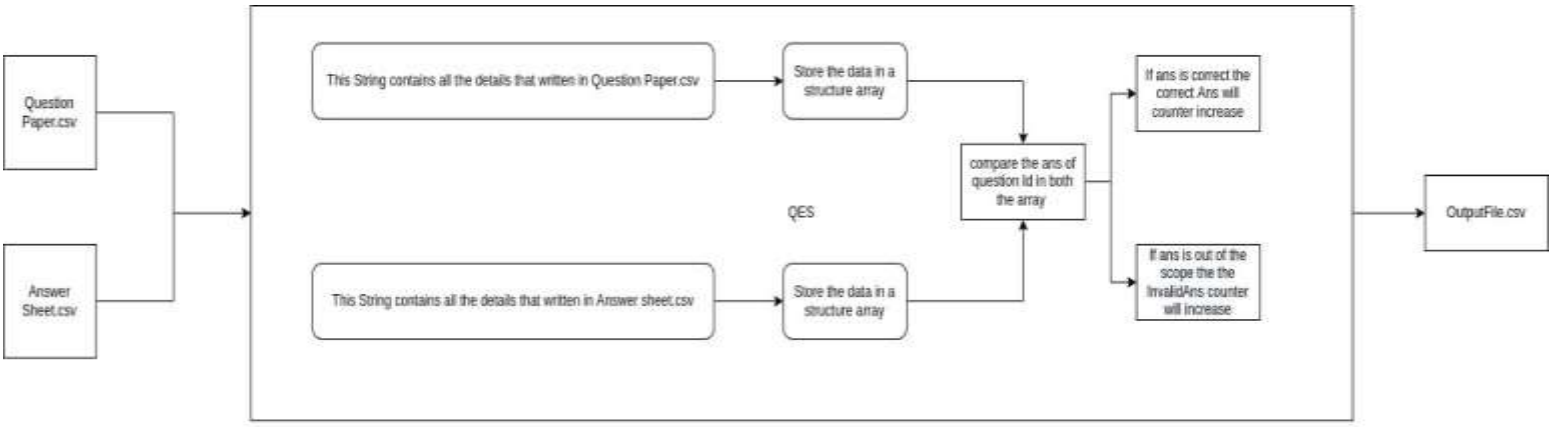
3.1. System Architecture Diagram.



3.2. Data Flow Diagram



3.3. Business Process workflow



3.4. Reliability

The system is available when the user has requested the service andit is available 24/7. The system has a very low failure rate.

3.5. Maintainability

There is no maintenance required.

3.6. Portability

This system should have the ability, once it is together, the entire system should be able to be virtual moved to any location. Code and program portability should be possible between kernel-recompiled Linux distributions. For everything to work properly, all components should be compiled from the source.

3.7. Reusability

The code written and the components used should have the ability to be reused with no problems. Should time allow, and detailed instructions are written on how to create this project, everything will be completely reusable to anyone.

3.8. Application compatibility

The QES is designed as an independent system. As it is not connected to any other components or interfaces, application compatibility is not a concern.

3.9. Resource utilization

The QES uses very limited resources.

4. Functions used

Our main logic is in evaluate function. Here are pseudo code for evaluate function is

```
For(loop)
{
found = fscanf(questionPaper, "%d,%d,%d",
&questions[count].question_id, &questions[count].num_of_option,
&questions[count].correct_ans);
if (found != 3) break;
}
fclose(questionPaper);
total_marks = count;
count = 0;
for(loop)
{
found = fscanf(answerSheet, "%[^,],%d,%d",
ans[count].participant_name, &ans[count].question_id,
&ans[count].option_chosen);
if (found != 3) break;
}
fclose(answerSheet);
for (; j < count; j++)
{

for(k = 0; k < count; k++)
{
if(questions[j].question_id == ans[k].question_id)
{
if(questions[j].correct_ans == ans[k].option_chosen)
{
scored_marks += 1;
}
if(ans[j].option_chosen > questions[k].num_of_option)

{

invalid_ans += 1;
}
break;
}
}
}
```

We are printing the csv by using output_CSV function. The pseudo code for our function is

```
if(fopen(OUTFILE, USER_PARAMS_R) == NULL)
{
output = fopen(OUTFILE, USER_PARAMS_W);
fprintf(output, "Participant_name, Total_marks, scored_marks,

invalid_answers\n");

fprintf(output, "%s, %d, %d, %d\n", name, total_marks,

scored_marks, invalid_ans);
}
output = fopen(OUTFILE, USER_PARAMS_A);
fprintf(output, "%s, %d, %d, %d\n", name, total_marks, scored_marks,
invalid_ans);
```

5. References

The references:

[1] Implementation of Quiz Evaluation Software

[2] System Requirements Specification Document

[3] Design and Implementation of a Quiz Evaluation Software

[4]Project Proposal Document.

[5]System Specification Requirement.

[6]High-Level Design Document for QES.

[7] https://stackoverflow.com/questions/41798744/reading-csv-file-into-an-array-in-c

[8]https://www.sanfoundry.com/c-program-even-odd-elements-2-separate-arrays/

[9]https://stackoverflow.com/questions/3707517/make-file-echo-displaying-path-
String