

SOFTWARE DESIGN DOCUMENT FOR MATHEMATICS COMPETITION MANAGEMENT SYSTEM

<https://github.com/OpiyoOscar01/G1-Recess-Math-Competition-Management-System->

G-1

SUPERVISOR: MARY NSABAGWA

Name	Registration Number	Student Number
Opiyo Oscar	23/U/1330	2300701330
Isio Lily	23/U/0445	2300700445
Namatovu Christine Maria	23/U/1098	2300701098
Babirye Sandra	23/U/07270/EVE	2300707270
Ogwal Sandra	23/U/16574/PS	2300716574

Table of Contents

1.Introduction	6
1.1 Purpose	6
1.2 Scope	6
1.2.1 Goals and Objectives	6
1.3 Document Overview	6
1.4 Reference Material	8
1.5 Definitions and acronyms	9
1.5.1 Definition	9
1.5.2 Acronyms	9
2.System Overview	10
2.1 System Description	10
2.2 Software requirements	10
2.3 Hardware requirements	11
3.System Architecture	12
3.1 Architectural Design	12
3.2 Decomposition Design	13
3.3 Design Rationale	17
4. Data design	18
4.1 Data description	18
4.2 Data Dictionary	20
5. Component Design	25
5.1 Administrator component	25
5.1.1 Administrator Login (Component Name)	25
5.1.2 Administrator Upload (Component Name)	25
5.1.3 Administrator Upload (Component Name)	26
6. Human Interface Design	27
6.1 Overview of the User Interface	27
6.1.1 Participant Interaction	27

6.1.2 Customer Interaction	27
6.2 Screen Images	28
6.3 Screen Objects and Action.....	35
7.Requirements Matrix	36

List of figures

Figure 3. 1 This shows the System Architecture Design.	12
Figure 3. 2 This shows the use cases of the system	13
Figure 3. 3 This shows the Administrator interaction with the system.	14
Figure 3. 4 This shows the interaction between the participant and the system.	15
Figure 3. 5 The figure above shows the interaction between the school representative and the system.	16
Figure 4. 1 This figure shows the EER design of the system.	19
Figure 6. 1 This figure shows the question view	28
Figure 6. 3 This is the CLI for the pupil registration.	30
Figure 6. 4 This shows the CLI through which the school representative logs in.	30
Figure 6. 5 This is the challenge view.	30
Figure 6. 6 This shows the CLI for the new applicants.	31
Figure 6. 7 The CLI for the challenge attempt.	31
Figure 6. 8 This is a CLI for managing the challenge.	31
Figure 6. 9 the login form view for the participants' view	32
Figure 6. 10 The figure shows the menu item Viewchallenges interface.	32
Figure 6. 11 This is the administrator log in form.	33

List of Tables

Table 4. 1 This defines the different entities of the system.	18
Table 4. 2 This table shows the administrator data and how they are stored in the database.	20
Table 4. 3 This describes the pupil's data and how it is stored in the database.	21
Table 4. 4 this describes the school data as will be stored in the database.	21
Table 4. 5 This describes the school representatives' information as will be stored in the database.	22
Table 4. 6 This describes the challenge data and how it is stored in the database.	23
Table 4. 7 This describes how the answer data is stored in the database.	23
Table 4. 8 This shows the whether a participant has been accepted or rejected for the challenge.	24
Table 4. 9 This shows the details of the questions as to be stored in the database.	24
Table 4. 10 This is sent to the participant as a PDF showing the details of what challenge they attempted.	25
Table 6. 1 This shows the administrator's dashboard.	33
Table 7. 1 The requirements matrix table.	36
Table 7. 2 The requirements matrix table continued.	37

1.Introduction

1.1 Purpose

The Mathematics Competition Management System aims to provide a user friendly and secure environment for all eligible primary pupils from registered schools in Uganda, to take part in an online mathematics challenge.

1.2 Scope

The system is being designed to enable learners access the challenges they would wish to take part in by providing a command line interface through which they will log into the system.

1.2.1 Goals and Objectives

- Enable registered administrators upload eligible schools' details.
- Enable registered administrators upload questions and their corresponding answers.
- Enable school representatives to view the registered prospective participants from their respective schools.
- The school representatives should be able to confirm that the pupils belong to their schools.
- Give access to participants who have entered their correct school registration numbers.
- Enable pupils attempt the challenges they have opted for.
- Enable pupils view their results after the completion of a challenge.

1.3 Document Overview

This software design document is organized headings to show the design of the Mathematics Competition Management System.

It includes;

Chapter 1: Introduction

This identifies the purpose, scope of the Software Design document and its intended users.

Chapter 2: System overview

This gives the general description of the functionality, context and design of the project.

Chapter 3: System Architecture

This id decomposed into sub-levels which includes;

3.1 The architectural Design

This shows a modular program structure and explains the relationships between the modules in order to attain the complete functionality of the desired system.

3.2 Decomposition description

This explains the decomposition of the sub modules in the architectural design.

3.3 Design rationale

This provides the reasons for selecting the architecture described in the architectural design including critical issues that were considered.

Chapter 4: Data design

This also has sub-stages that is;

4.1 Data description

This explains how the data domain of the system is transformed into data structures.

4.2 Data dictionary

This provides a list of the systems entities along with their types and description.

Chapter 5: Component design

This gives complete details of the systematic operation of each component in the system.

Chapter 6: Human Interface design

This has three sub-levels;

6.1 The overview of the user

This describes the functionality of the system from the users' perspective.

6.2 Screen images

This displays screenshots showing interfaces from the users' perspectives.

6.3 Screen Objects and Actions

This describes the screen objects and actions associated with these objects.

1.4 Reference Material

1.5 Definitions and acronyms

1.5.1 Definition

Use case diagram – this summarizes the details of the system and the users within the system.

Activity diagram – this describes the different actors in the system and their roles in the system.

Entity relationship diagram – this provides a visual starting point for the database design.

1.5.2 Acronyms

SDD – Software Design Document

ERD – Entity Relationship Diagram

CLI – Command Line Interface

PDF – Portable Document Format

SRS- Software Requirements Specification

2.System Overview

2.1 System Description

The system is designed to enable primary school pupils to log into the system through a command line interface which is designed using java programming language. Through the interface, they will be able to register with their username, firstName, lastname, email, Date_Of_Birth, school registration number, and add their image.png. having successfully registered, the participants are granted access to view the available challenges, and start attempting the challenge they are interested in. The system will also allow registered administrator to upload the details of the schools which will be taking part in the competition, and this includes the school's name, the district where it is located, the school representative number, their emails, and names. He or she is also able to upload questions and answers which are excel documents. All these details are stored in a java file.

The Apache web server therefore continuously fetches information from the java file and stores it in the system database. The fetched information is then posted on the web browser used by the clients.

The school representatives through the command line interface log into the system to confirm the newly registered prospective participants via a menu item ViewApplicants, which displays all their registration numbers. To reject or activate an applicant, the representative enters the menu item confirm yes/no username. If the participant is rejected, the record is picked from the file containing student details and moved into a table "Rejected" in the database, and if a participant is activated, their record is also moved from the student file and moved into the table 'Accepted" within the database. The student then proceeds to attempt the challenge.

When a challenge is opened, the participant issues a command via the command line interface "AttemptChallenge challengeNumber". He or she is given a maximum of three chances to attempt the challenge and at each attempt, random questions are presented. The number of remaining questions and time are indicated above the question. For every wrong answer, 3 marks are deducted, and where the participant is not sure, 0 marks are awarded. Otherwise, they get the marks attached to every for correct answers.

At the end of a challenge, the participant sees their scores, time taken for each attempt and the total time they taken to complete the challenge. A report showing most correctly answered questions, best performing schools and participants over the past few years, worst performing schools, participants with incomplete challenges is generated at the end of the competition.

2.2 Software requirements

- Web browser
- Database Management System (DBMS)
- Apache web server

- Java compiler

2.3 Hardware requirements

- Desktop components
- Phones
- Database server

3. System Architecture

3.1 Architectural Design

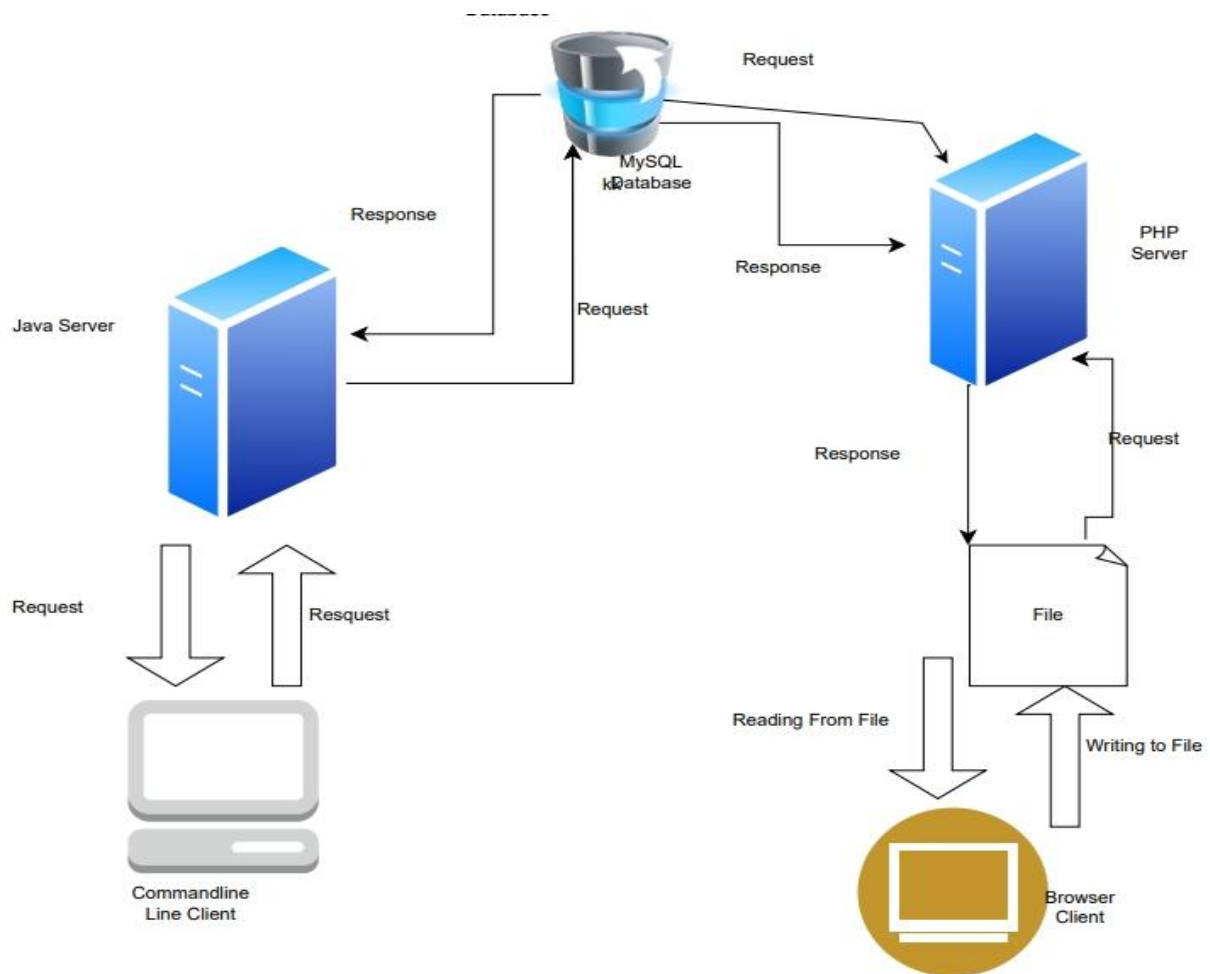


Figure 3. 1 This shows the System Architecture Design.

The command line interface provides the platform through which the participant registers into the system by entering the required information, and be able to attempt the challenge. This also enables the school representatives to view the applicants and confirm that they belong to their respective schools.

The php server controls access to the database where data about the schools, participants and administrators is stored.

The architecture we have chosen to use includes;

- Client-server architecture.
- Layered architecture.

3.2 Decomposition Design

Server

Decomposition design

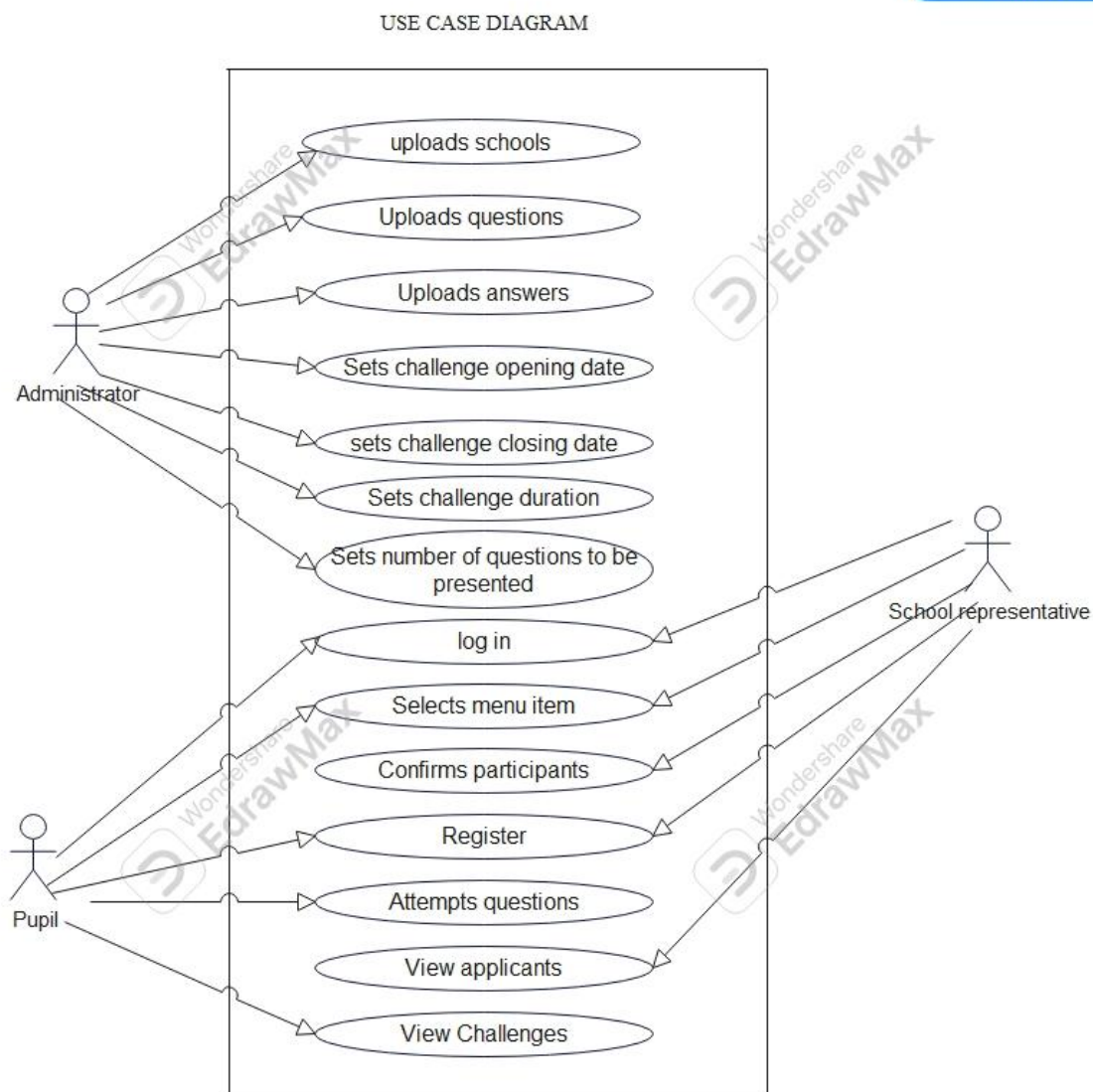


Figure 3. 2 This shows the use cases of the system

This figure 3.2 describes the interaction between the pupil, school representative and administrator with the system.

Activity diagrams

Administrator and the system.

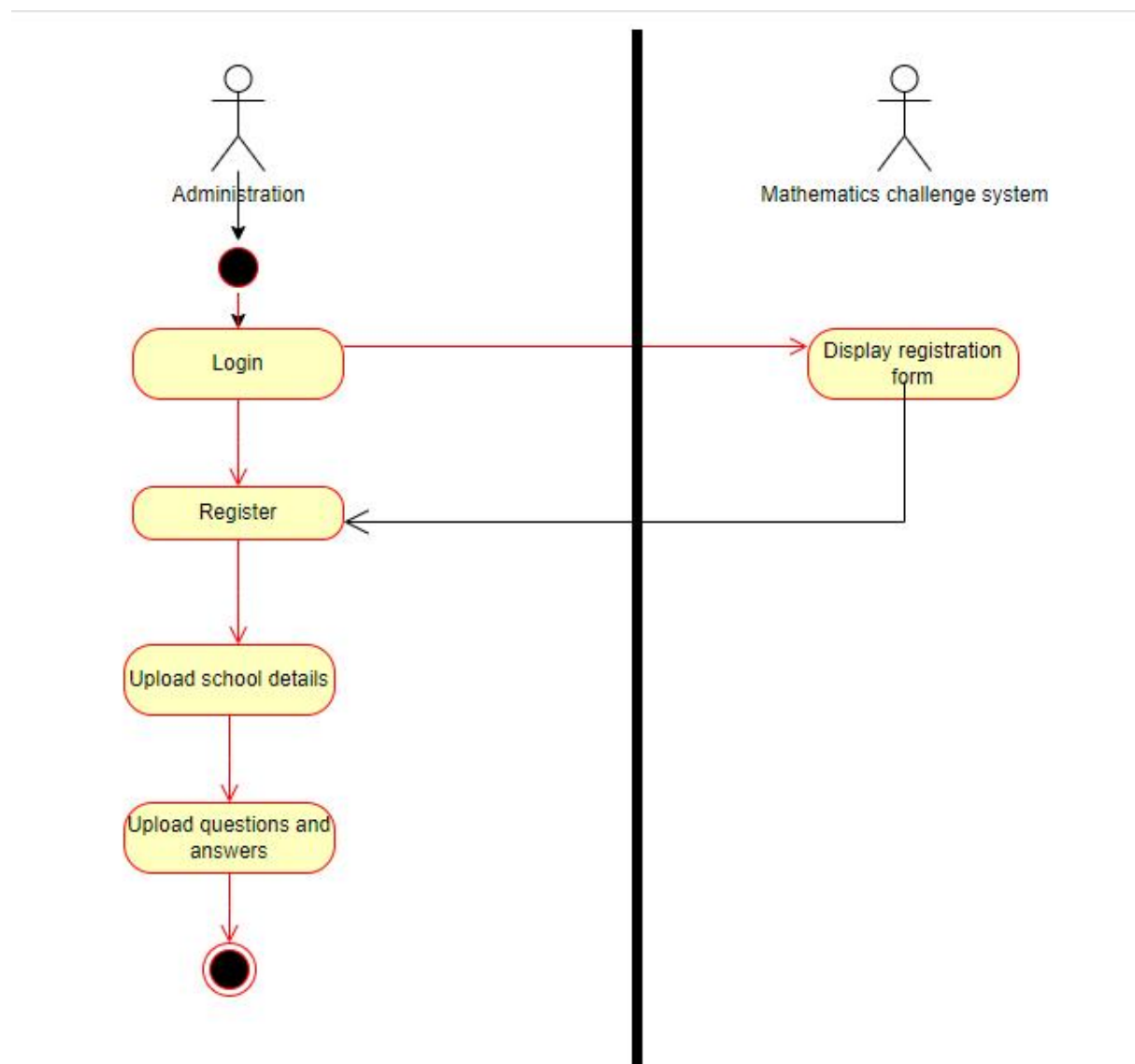


Figure 3. 3 This shows the Administrator interaction with the system.

Participant and the system

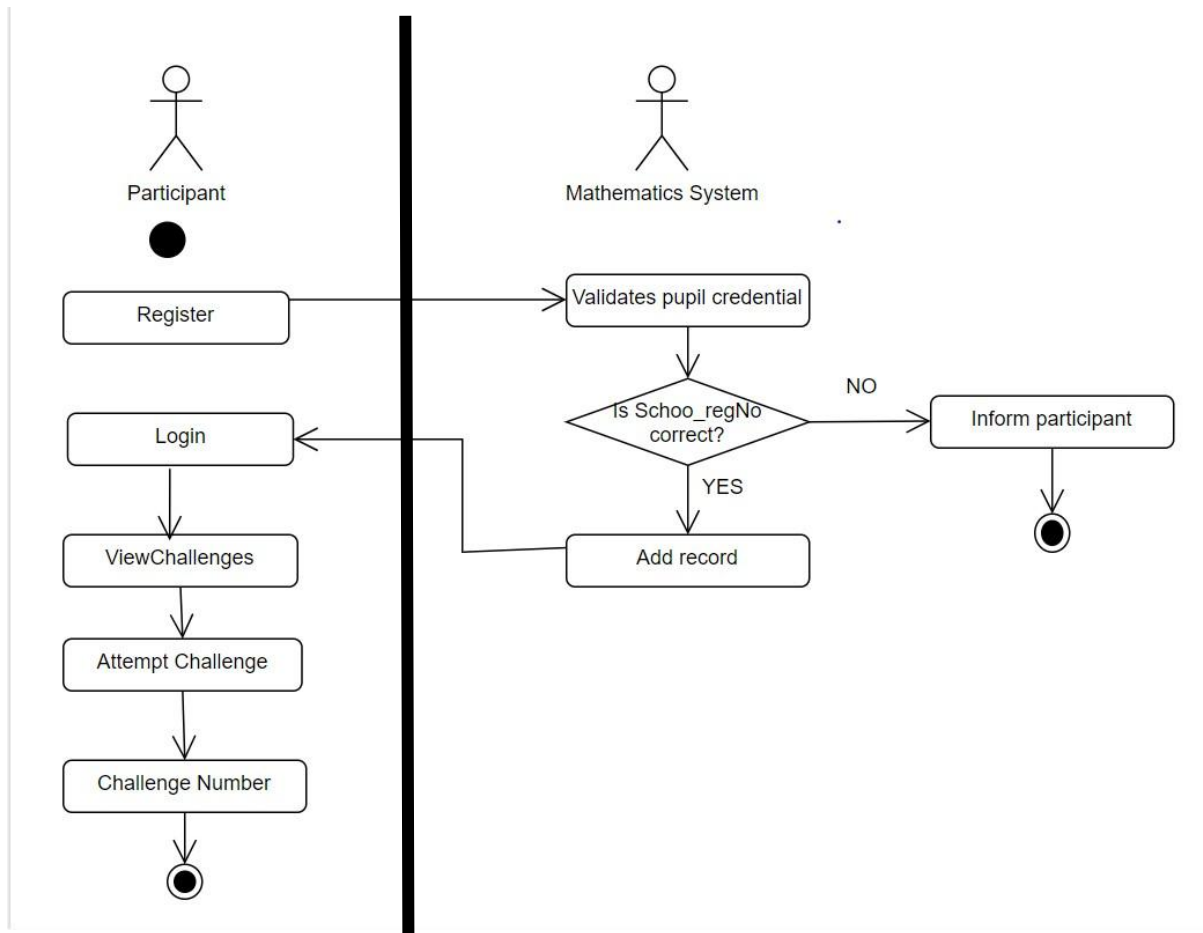


Figure 3. 4 This shows the interaction between the participant and the system.

School representative and the system

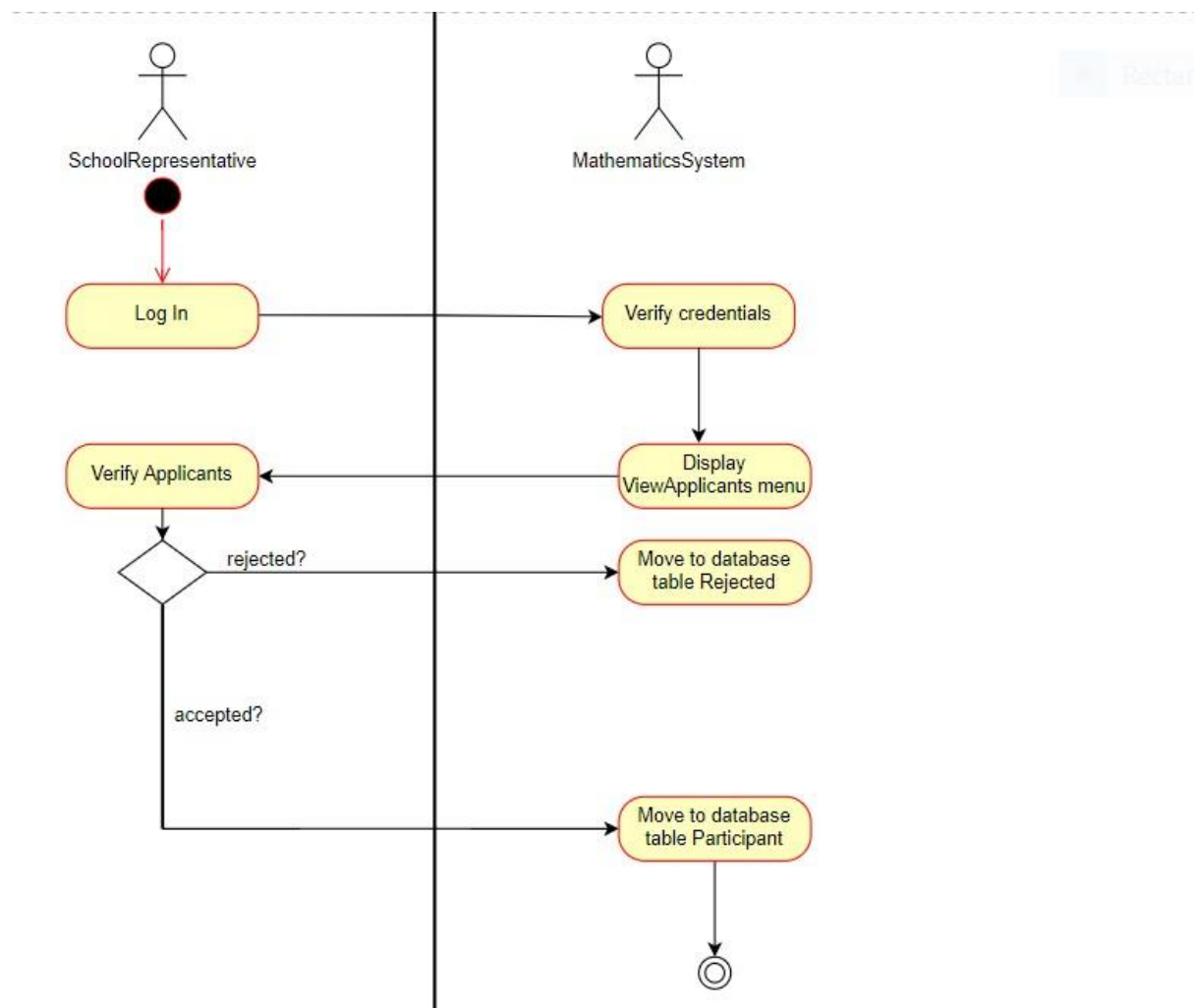


Figure 3. 5 The figure above shows the interaction between the school representative and the system.

3.3 Design Rationale

We considered the architectural design because with the layered architecture, there is separation of concern, ensures code modularity and codes are easy to maintain. The client-server architecture provides a centralized access to data, and the relational database management system controls the movement of information to and from the text file on the participants' side and the web interface on the administrator side.

4. Data design

4.1 Data description

Entity	Attributes
Questions	questionNumber (PK) questionText marks
Challenge	challengeID (PK) openingDate closingDate duration numOfQuestion [1...10]
Administration	username password
School	schoolRegistrationNumber (PK) name district
Pupil	registrationNumber (PK) username firstName lastName email date-of-birth image
Answers	questionNumber (FK) answerText
Confirmation	Pupil_reg_number (FK) Confirmationstatus (Yes/no)
Report	Pupil_reg_number (FK) Image challengeID (FK) pupilScores Answerspdf
SchoolRepresentative	representativeID (PK) Name Email

Table 4. 1 This defines the different entities of the system.

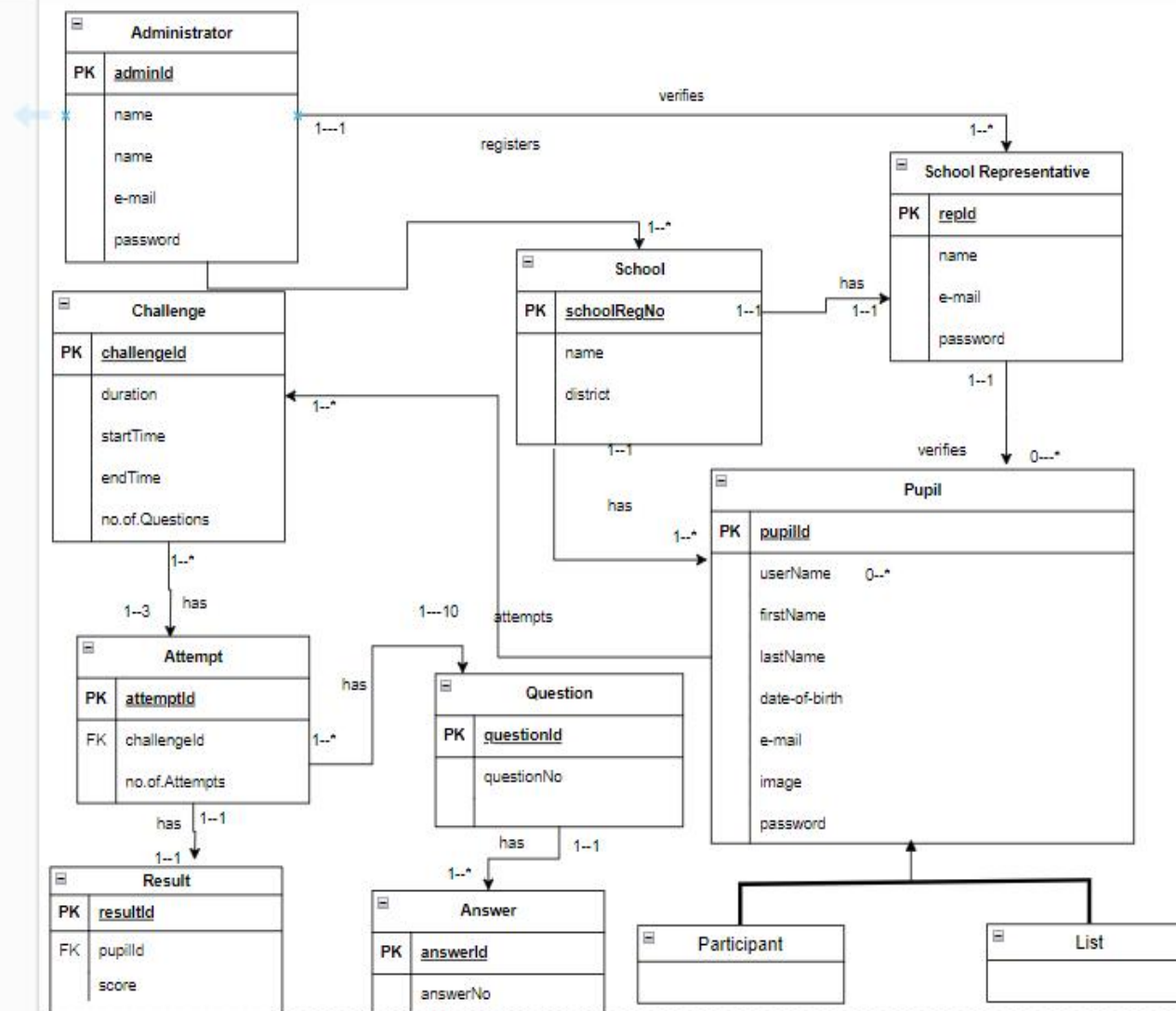


Figure 4. 1 This figure shows the EER design of the system.

4.2 Data Dictionary

Administrator

Attribute	Datatype	Size	Constraint	Description
Username	VARCHAR	25	Primary Key	This shows the name used by the administrator While logging in.
Password	VARCHAR	25	UNIQUE NOT NULL	This has to be UNIQUE and must not be left empty at any point.

Table 4. 2 This table shows the administrator data and how they are stored in the database.

Participant

Attribute	Datatype	Size	Constraint	Description
Reg_No	VARCHAR	25	PRIMARY KEY	This is the pupil's registration number given by the school at the time of registration
username	VARCHAR	25	NOT NULL	This indicates the name they used to create an account to enter the competition and must not be left empty.
firstName	VARCHAR	25	NOT NULL	The participant's first name and must not be left empty.
lastName	VARCHAR	25	NOT NULL	The participant's last name and must not be

				left empty.
Email	VARCHAR	25	UNIQUE	The participant's email which must be unique.
DateOfBirth	DATE		NOT NULL	The date of birth of the participant and must not be left empty.
Image.png	BLOB			The participant's image.

Table 4. 3 This describes the pupil's data and how it is stored in the database.

School

Attribute	Datatype	Size	Constraint	Description
School_regNo	VARCHAR	25	PRIMARY KEY	This identifies the school as unique and distinguishes it from other schools.
Name	VARCHAR	25	UNIQUE	The school's name and must be unique.
District	VARCHAR	25	NOT NULL	The location of the school and must be given.
representativeName	VARCHAR	25	NOT NULL	The school's representative's name which must be given.
representativeEmail	VARCHAR	25	NOT NULL	His email which must be provided.

Table 4. 4 this describes the school data as will be stored in the database.

School Representative

Attribute	Datatype	Size	Constraint	Description
representativeID	VARCHAR	10	PRIMARY KEY	This uniquely identifies each representative of the different schools.
representativeEmail	VARCHAR	25	UNIQUE	These must be unique.
representativeName	VARCHAR	25	NOT NULL	The representatives must provide their name.

Table 4. 5 This describes the school representatives' information as will be stored in the database.

Challenge

Attribute	Datatype	Size	Constraint	Description
challengeID	VARCHAR	10	PRIMARY KEY	This identifies the challenge number or availability.
openingDate	DATE		NOT NULL	This shows the date when the challenge will be opened to the participants.
closingDate	DATE		NOT NULL	This shows the date when the challenge will be opened to the participants.

Duration	TIME		NOT NULL	This shows the time the challenge will take once opened.
numQuestion [1...10]	INT		NOT NULL	The random questions picked from the uploaded questions.

Table 4. 6 This describes the challenge data and how it is stored in the database.

Answer

Attribute	Datatype	Size	Constraint	Description
questionNumber	INT		PRIMARY KEY FOREIGN KEY references Question	The number of the question to be attempted.
answerText	VARCHAR			The answer that corresponds to the question.

Table 4. 7 This describes how the answer data is stored in the database.

Confirmation

Attribute	Datatype	Size	Constraint	Description
registrationNumber	VARCHAR	25	PRIMARY KEY FOREIGN KEY references the student.	This uniquely identifies the pupil.
Image	BLOB			This shows the pupil's image.
confirmationStatus	YES/NO			This indicates whether a pupil has been activated or rejected for the challenge.

Table 4. 8 This shows the whether a participant has been accepted or rejected for the challenge.

Questions

Attribute	Datatype	Size	Constraint	Description
questionNumber	INT		PRIMARY KEY	This identifies every question uniquely in the challenge.
questionText	VARCHAR			This shows the question to be attempted.
marks	INT			This shows the marks attained by the participant after the attempted of a given challenge.

Table 4. 9 This shows the details of the questions as to be stored in the database.

Report

Attribute	Datatype	Size	Constraint	Description
registrationNumber	VARCHAR	25	FOREIGN KEY	This uniquely identifies

			references Pupil	the pupil.
Image	BLOB			This shows the pupil's image.
challengeID	VARCHAR		FOREIGN KEY references Challenge	This references the attempted challenge.
pupilScore	INT			This shows the pupil's total score.
Answers	VARCHAR			This shows the answers the participant submitted.

Table 4. 10 This is sent to the participant as a PDF showing the details of what challenge they attempted.

5. Component Design

This shows the detailed summary of the object functions of the system basing on the decomposition of the system in the decomposition design section. This includes component diagrams, component name, object functions, component description, inputs, processing, outputs and component interfaces.

5.1 Administrator component

5.1.1 Administrator Login (Component Name)

Object function: ValidateAdministrator ()

Description: This function validates the administrator and grants them access to the administrator dashboard given that they entered the correct credentials.

Input: Username, Password.

Processing: The system validates the administrator's credentials against the administrator's table in the database, and then they are granted access to the dashboard if their username and password match those in the database.

Output: The authentication results whether it was a success or failure.

Interface: User Interface Component which takes in the administrator's inputs and outputs, and Administrator's database for storing and retrieving the credentials.

5.1.2 Administrator Upload (Component Name)

Object function: uploadSchoolData ()

Description: This object function is used by the administrator to upload the school details, process the information including data validation, storage and updating the database.

Input: School name, district, school registration number, representative name, representative email, and administrator authentication credentials.

Processing: Validate school data, verify administrator credentials and upload the school information to the database and where need be, update existing records or insert new records.

Output: Upload results whether it was a success or failure.

Interface: User Interface Component which takes in the administrator's inputs and outputs, Administrator's Authentication Component which verifies the credentials and School Database Component for storing and retrieving the school data.

5.1.3 Administrator Upload (Component Name)

Object function: `ValidateSchoolRepresentatives ()`

Description: This object function is responsible for validating the school representatives' information upload by the administrator to ensure that the data is accurate and consistent.

Input: The school representatives' name, email and the administrator's authentication credential.

Processing: validate the school representatives' information against predefined rules and formats, check for duplicates and verify consistency with existing school data in the database, and reject inconsistent data.

Output: Validate results whether it was a success or failure.

Interface: User Interface Component which takes in the administrator's inputs and outputs, Administrator's Authentication Component which verifies the credentials, School Database Component for storing and retrieving the school data and Validation Rules Component which accesses the validation rules and formats.

6. Human Interface Design

6.1 Overview of the User Interface

6.1.1 Participant Interaction

A. Pupil

The eligible pupils from all registered schools in Uganda are to register using the command line interface by entering their registration_ number, username, firstname, lastname, password. Having successfully registered, they can go ahead to view the available challenges by selecting a menu item ViewChallenges, and they can attempt the challenges they want.

B. School Representative

The school representatives use the command line interface as well to log into the system and verify that a specific registered pupil belongs to their respective schools. If they successfully login, they choose a menu item ViewApplicants which displays a list of registers pupils with their registration numbers and he verifies whether they belong to their respective schools. Then he goes ahead to confirm the pupils by choosing a menu item confirm yes/no username, this will either activate or reject a pupil depending on whether they entered correct school registration numbers.

C. Administrator

The administrator logs into the system. He uploads the schools with their details after successfully logging in. he also uploads the questions and their corresponding answers, sets the challenge opening and closing dates, the duration a challenge will take and the number of questions to be presented at each challenge.

6.1.2 Customer Interaction

A. Administrators

They oversee the whole competition and interact with the pupils and school representatives regarding any issues that arise during and after the competition.

B. School Representatives

They interact with the pupils to ensure they are successfully accepted for the competition. They also interact with the administrators to ensure their school details are correctly entered into the system.

6.2 Screen Images

7/10 questions Remaining time 20 mins

7. What is the area of a circle with a radius of 6.79 cm?

A. 140.45cm^2 B. 26.7 cm^2 C. 144.77 cm^2 D. None

Previous Next

Figure 6. 1 This figure shows the question view

A question is presented to the pupil one at a time and he or she selects the answer he or she feels is the correct answer. By clicking the Previous or next buttons, he or she is taken to the respective page they would like to view.


```
Enter the following details to register:
UserName:
FirstName:
LastName:
Email Address:
Date-of-birth(YYYY-MM-DD):
School Registration Number:
Upload profile image (image-file.png):
Registration successful. Please wait for confirmation from your school representative.
```

Figure 6. 3 This is the CLI for the pupil registration.

Through this interface in figure 6.3, the pupil enters their required credentials which include the username, firstname, lastname, among others.

```
School Representative Login:
UserName:
Password:
```

Figure 6. 4 This shows the CLI through which the school representative logs in.

In the figure 6.4, he enters his username and password in a predefined format.

```
Available Challenges:
1.Challenge #001(June 10, 2024)-Duration:60 minutes,Questions:10
2:Challenge #002(June 15, 2024)-Duration:45 minutes,Questions:10
Input the challenge number to attempt:
```

Figure 6. 5 This is the challenge view.

After the pupil successfully logging in, he or she writes a command to view the available challenges, and the above challenge view is displayed.

```
New Applicants:
UserName          Registration Number
-----
user1             ABC123
user2             XYZ456

Confirm or reject applicants?(CONFIRM/REJECT):
```

Figure 6. 6 This shows the CLI for the new applicants.

This enables the school representative to view the new participants which helps them to confirm and verify them.

```
Challenge #001- Question 1/10
What is 2+2?
A) 3
B) 4
C) 5
D) 6
Your Answer:[input field]
Time remaining: 55minutes
```

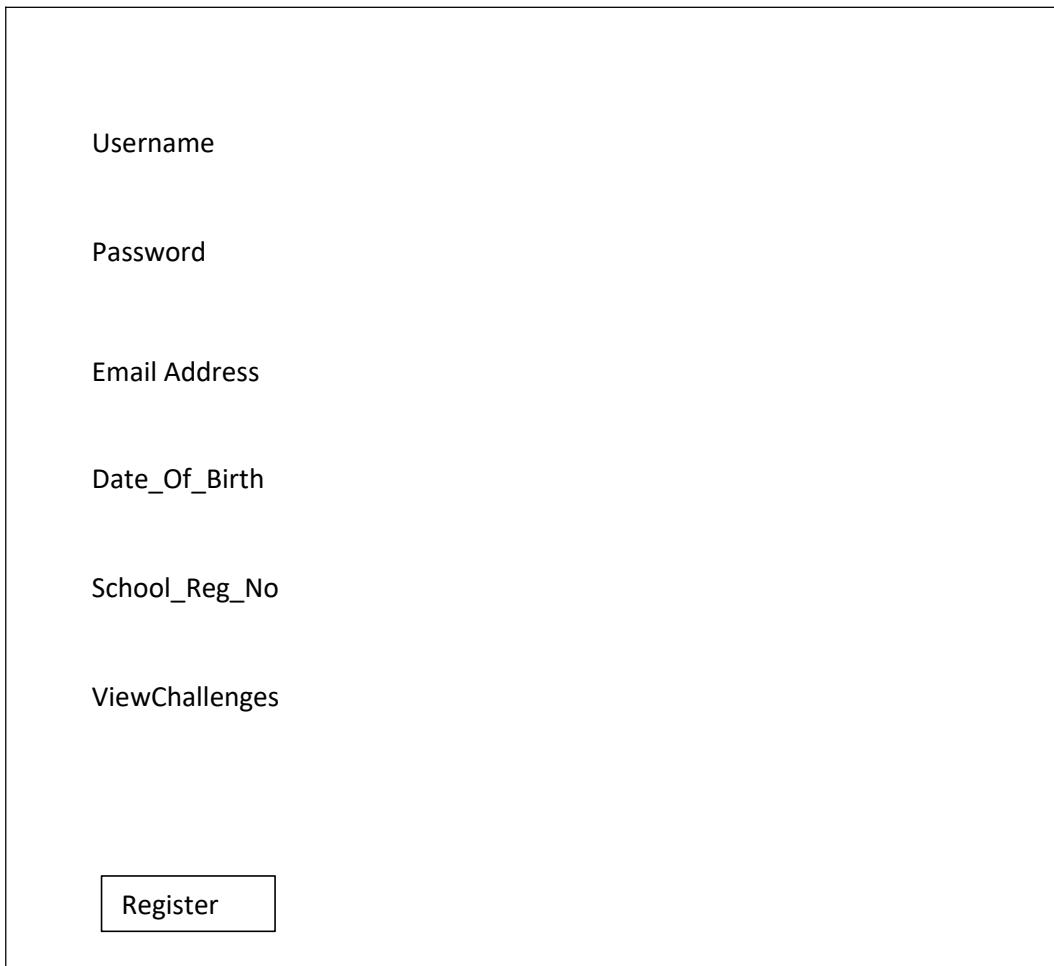
Figure 6. 7 The CLI for the challenge attempt.

This displays the selected challenge and the question number, the answer alternative and the remaining time for the challenge.

```
Manage Challenges:
1. Create New Challenge
2. View/Edit Existing Challenges
Select an option:[input field]
```

Figure 6. 8 This is a CLI for managing the challenge.

This figure enables the administrators to manage the challenge by creating, viewing or editing the existing challenges.

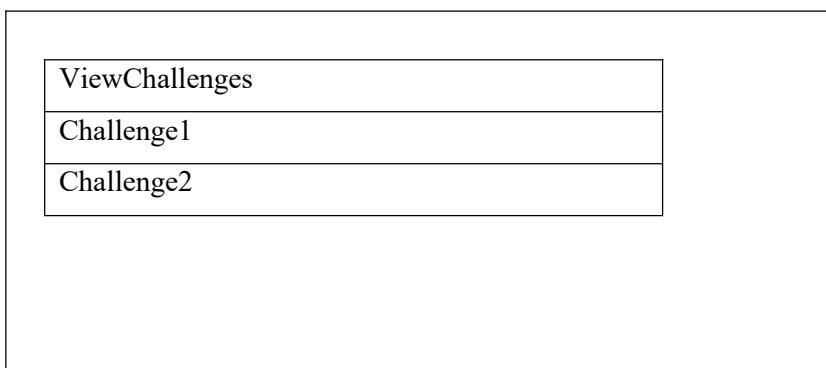


A login form for participants' view. It contains six text input fields labeled 'Username', 'Password', 'Email Address', 'Date_Of_Birth', 'School_Reg_No', and 'ViewChallenges'. Below these fields is a 'Register' button.

Username
Password
Email Address
Date_Of_Birth
School_Reg_No
ViewChallenges
Register

Figure 6. 9 the login form view for the participants' view

This form is where the participant enters their credentials and view the challenges, and they are verified against those in the database to ensure that they are correct. The highlighted text indicates a link to the available challenges.



A view challenges interface showing a table with three rows: 'ViewChallenges', 'Challenge1', and 'Challenge2'.

ViewChallenges
Challenge1
Challenge2

Figure 6. 10 The figure shows the menu item Viewchallenges interface.

In this view, the participant views the available challenges and attempts a challenge of their choice when they click on the submit button.

Admin Username

Password

Submit

Figure 6. 11 This is the administrator log in form.

This is the administrator ‘s dashboard

Administrator																								
Ranking	<table><tr><th>Pupil_regNo</th><th>schoolReg_no</th><th>Points</th></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>			Pupil_regNo	schoolReg_no	Points																		
Pupil_regNo				schoolReg_no	Points																			
Graphs																								
Percentage Performance																								

Table 6. 1 This shows the administrator’s dashboard.

The figure above shows the administrator's dashboard when he or she login into the system, he or she views the summary of performance of each and every school participant which includes the pupil registration number, their image among others.

6.3 Screen Objects and Action

Login

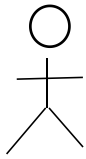
Once clicked, a request is sent to the database to verify the entered credentials.

Register

When clicked the database saves the entered credentials.

Submit

This works the same way as the login button.



This shows the different participants in the system.

7.Requirements Matrix

The requirements matrix maps each functional requirement from the SRS to the corresponding system components. This ensures that each requirement is included in the system design and can be traced throughout the development process.

The functional requirements describe what the system should do that is, the specific behaviors and functions the system must support, whereas the non- functional requirements specify how the system is going to achieve all the functions it must support.

The requirements matrix is as shown below.

Functional requirement	Component	Description
Register pupils.	CLI	This enables pupils to register into the system through the command line interface.
Verify if the school registration number is correct.	Participant Manager	This checks if the participants enter their correct school registration number.
Upload questions and answers excel documents	File Handler	This enables the administrator to upload the challenge questions and answers from excel into the system.
Sets the opening and closing dates, the duration of the challenge.	Competition Manager	This helps the administrators to schedule the challenges.
Display randomly selected questions at every attempt	Competition Manager	This ensures that the participant does not get to attempt the same questions, hence at each attempt, the questions are unique.
Shows the participant their scores after a challenge	CLI	The shows the participant the instant results after finishing an attempt, or when they are closed out of a challenge when the challenge duration elapses.

Table 7. 1 The requirements matrix table.

Functional Requirement	Component	Description
Send email notifications to participants and respective school representatives	Web Interface/ laravel	This sends the participants email at registration informing them whether they have granted access to attempt challenge or not, and the representatives are informed to confirm the registered participants.
Display competition results on the website	Web Interface/ Laravel	At the end of the competition, analytics of the competition are displayed on the website which includes best schools, best pupils, among others.
Generate analytics reports	Report Generator	This generates detailed analysis of the competition such as most correctly answered questions, the school rankings according to the participants' performance, etcetera.
Handling user authentication	Laravel/Web Interface	This ensures that all actors that will interact with the system enter correct credentials.

Table 7. 2 The requirements matrix table continued.

