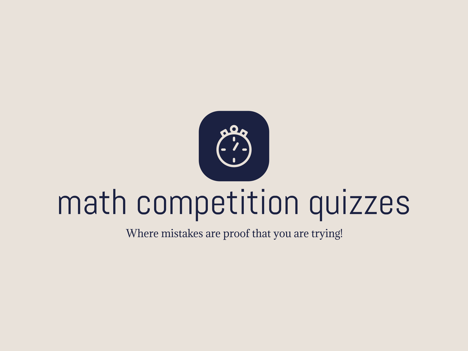
SOFTWARE DESIGN DOCUMENT

FOR

**THE MATHEMATICS CHALLENGE**



GitHub Link: <https://github.com/keithunt-35/G-7-mathematics-challenge>

G-7

Supervisor: Mary Nsabagwa

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Name** | **Registration No.** | **Email** | **Contact** |
| 1 | Nuwagaba Jerome Bill | 23/U/16213/PS | nuwagababill5@gmail.com | 778567267 |
| 2 | Aineamatsiko Desmond Cole | 23/U/05278/PS | desmondrosscole@gmail.com | 776430563 |
| 3 | Kayonga Trevor Joel | 23/U/09512/EVE | joeltrevork@gmail.com | 702804880 |
| 4 | Tugume Keith Kaganda | 23/U/17996/EVE | tugumekeith022@gmail.com | 752331549 |
| 5 | Ashaba Ahebwa Mable | 23/U/06570/PS | mableashaba03@gmail.com | 708538961 |

Table of Contents

[1 INTRODUCTION 1](#_Toc169808024)

[1.1 Purpose 1](#_Toc169808025)

[1.2 Scope 1](#_Toc169808026)

[1.2.1 Objectives 1](#_Toc169808027)

[1.2.2 Goals of the project 1](#_Toc169808028)

[1.2.3 Benefits of the project 1](#_Toc169808029)

[1.3 Overview of the document. 2](#_Toc169808030)

[1.3.1 Introduction 2](#_Toc169808031)

[1.3.2 System Overview 2](#_Toc169808032)

[1.3.3 System Architecture 2](#_Toc169808033)

[1.3.4 Data Design 3](#_Toc169808034)

[1.3.5 Component Design 3](#_Toc169808035)

[1.3.6 Human Interface Design 3](#_Toc169808036)

[1.4 Reference Material 3](#_Toc169808037)

[1.5 Definitions and Acronyms 4](#_Toc169808038)

[1.5.1 Definitions 4](#_Toc169808039)

[1.5.2 Acronyms 4](#_Toc169808040)

[2 SYSTEM OVERVIEW 5](#_Toc169808041)

[2.1 software requirements 6](#_Toc169808042)

[2.2 hardware requirements 7](#_Toc169808043)

[3 SYSTEM ARCHITECTURE 8](#_Toc169808044)

[3.1 Architectural Design 8](#_Toc169808045)

[3.2 Decomposition Description 11](#_Toc169808046)

[3.3 Design Rationale 18](#_Toc169808047)

[3.3.1 Client-Server architectural design. 18](#_Toc169808048)

[3.3.2 Advantages of Client-Server. 18](#_Toc169808049)

[4 Data Design 19](#_Toc169808050)

[4.1 Data Description 19](#_Toc169808051)

[4.2 Data Dictionary 20](#_Toc169808052)

[5 COMPONENT DESIGN 26](#_Toc169808053)

[5.1 Web Interface Module 26](#_Toc169808054)

[5.1.1 Administrator: 26](#_Toc169808055)

[5.2 CLI Module 28](#_Toc169808056)

[5.2.1 School Representative: 28](#_Toc169808057)

[5.2.2 Participant: 29](#_Toc169808058)

[6 HUMAN INTERFACE DESIGN 31](#_Toc169808059)

[6.1 Overview of User Interface 31](#_Toc169808060)

[6.1.1 Pupil interactivity 31](#_Toc169808061)

[6.1.2 School representative interactivity 31](#_Toc169808062)

[6.1.3 Administrator 31](#_Toc169808063)

[6.2 Screen Images 32](#_Toc169808064)

[6.2.1 Pupil interactivity 32](#_Toc169808065)

[6.2.2 School representative interactivity 33](#_Toc169808066)

[6.2.3 Administrator 35](#_Toc169808067)

[6.3 Screen Objects and Actions 36](#_Toc169808068)

List of figures

[Figure 3.1.1 Architectural Design 8](#_Toc169808387)

[Figure 3.1.2 Component Design 9](#_Toc169808388)

[Figure 3.2.1 Use case diagram 11](#_Toc169808389)

[Figure 3.2.2 Activity Diagram of the user 13](#_Toc169808390)

[Figure 3.2.3 Activity Diagram of the pupil 14](#_Toc169808391)

[Figure 3.2.4Activity Diagram of the pupil 15](#_Toc169808392)

[Figure 3.2.5 A Sequence Diagram 16](#_Toc169808393)

[Figure 3.2.6 Sequence Diagram 17](#_Toc169808394)

[Figure 4.2.1 Entity relationship diagram for the database structure 25](#_Toc169808395)

[Figure 6.2.1 Pupil Registration Form 32](#_Toc169808396)

[Figure 6.2.2 Challenges list 32](#_Toc169808397)

[Figure 6.2.3 Representative login 33](#_Toc169808398)

[Figure 6.2.4 Representative Menu 33](#_Toc169808399)

[Figure 6.2.5 Confirming Participants 33](#_Toc169808400)

[Figure 6.2.6 School Representative sign in 34](#_Toc169808401)

[Figure 6.2.7 Administrator Sign in 35](#_Toc169808402)

[Figure 6.2.8 Administrator homepage 35](#_Toc169808403)

List of tables

[Table 1 Schools 19](#_Toc169807622)

[Table 2 Participants 20](#_Toc169807623)

[Table 3 Challenges 20](#_Toc169807624)

[Table 4 Questions 21](#_Toc169807625)

[Table 5 Answers 21](#_Toc169807626)

[Table 6 Attempts 22](#_Toc169807627)

[Table 7 Reports 23](#_Toc169807628)

[Table 8 Verified Participant 23](#_Toc169807629)

[Table 9 Screen Images 35](#_Toc169807630)

# INTRODUCTION

## Purpose

This software design document describes the architecture and system design of The Math Challenge System. It shows the software functionality and how the team intends to build it.

It also shows the graphical and tabular descriptions of the software design for the mathematics challenge system to enable pupils to compete in the challenge.

## Scope

This software will provide a comprehensive platform for managing a national mathematics competition for primary school children, offering streamlined registration, challenge management and detailed reporting. The key players involved are the administrators, school representatives and participants.

### Objectives

* To design the system making sure it meets the user requirements
* To Implement the system
* To test the system ensuring efficient performance

### Goals of the project

* To help breed better mathematicians.
* To improve pupils’ performance in all mathematics related subjects.
* To help teachers and schools identify the pupils’ areas of strength and weakness in mathematics.

### Benefits of the project

* This project will promote quick thinking among pupils since there is a time limit on how long they can attempt a question.
* This project is inclusive as it also pupils do not have to convene in a central place to be able to attempt the challenges since they can already attempt them online.
* This project reduces the margin of error in marking or malpractice since marks are generated by the system and not by any individuals.

**1. Improved Efficiency**

Automation of registration, challenge management, and reporting processes reduces administrative workload.

**2. Increased Participation**

Simplified registration and participation processes encourage more schools and pupils to join the competition.

**3. Enhanced Learning**

Detailed feedback helps participants understand their strengths and areas for learning

**4. Recognition and Motivation**

Recognizing top performers motivates participants and schools to excel.

**5. Fair Competition**

Random question selection and clear scoring rules ensure a fair competition for all participants.

**6. Scalability**

The system can handle large numbers of participants and challenges, making it suitable for nationwide implementation.

7.**Acccessibility**

The system enables pupils even in remote area to participate in the competition.

## Overview of the document.

This document is structured to detail the entire system requirements for The Math Challenge System.

### Introduction

Identifies the purpose of this Software Design Document (SDD) and its intended audience.

### System Overview

Provides a general description of the system's functionality, context, and design. Interactivity amidst the various components that make up the system.

### System Architecture

This has been broken down into smaller sub-levels;

#### Architectural Design

Shows the modular program structure and explains the relationships between modules to achieve complete system functionality.

#### Decomposition Description

Breaks down the subsystems in the architectural design.

#### Design Rationale

Explains the reasons for the selected architecture, including critical issues and trade-offs considered.

### Data Design

this has also been divided into sub-levels:

#### Data Description

Explains how the system's information domain is transformed into data structures.

#### Data Dictionary

Provides a list of system entities or major data with their types and descriptions.

### Component Design

Provides a detailed view of the systematic operation of each component.

### Human Interface Design

Divided into sub-levels:

#### Overview of User

Describes the system's functionality from the user's perspective.

#### Screen Images

Displays screenshots showing the user interface.

#### Screen Objects and Actions

Describes the screen objects and the actions associated with these objects.

## Reference Material

R. S. Pressman, Software Engineering A Practitioner’s Approach 5th ed. 2001.

Sommerville, Ian Software engineering — 9th ed. ISBN-13: 978-0-13-703515-1 ISBN-10: 0-13-703515-2

Joseph Valacich, Joey George, 2020, Modern Systems Analysis and Design 9th Edition, Pearson, ISBN-13 : 978-0135172759

J. S. van der Ven, A. Jansen, J. Nijhuis, and J. Bosch. Design Decisions: The Bridge between Rationale and Architecture

J. Burge, J. Carroll, R. McCall, and I. Mistrík. Rationale-Based

Software Engineering. Springer-Verlag, 2008.

## Definitions and Acronyms

### Definitions

|  |  |
| --- | --- |
| Activity Diagram | Describes the flow of activities of the different actors in the system. |
| Entity Relationship Diagram | Provides a visual starting point for the database design. |
| Sequence Diagram | Shows the sequence of messages passed between objects of the system. |
| Use case diagram | Summarizes the details of the system and the users within the system. |

### Acronyms

|  |  |
| --- | --- |
| **Abbreviation** | **Full form** |
| ERD | Entity Relationship Diagram |
| SDD | Software Design Document |
| CLI | command line interface |
| PDF | portable document format |
|  |  |

# SYSTEM OVERVIEW

Over the years in the country, the subject of Mathematics at the primary level of education has faced a few challenges that the International Education Services is stepping in to try and tackle. Some these challenges include;

Lack of a standard grading and assessment system for students across the country. Since primary schools are spread across the country, each school evaluates and grades their pupils differently and therefore as they move on to higher levels, it is difficult how well pupils from different schools were accessed. And therefore, this system provides a standard assessment for students across the country which makes it fairly distinguish the best pupils from the rest.

Malpractice and Grading errors. There has been a problem of malpractice among schools in form of teachers favoring some students while marking or undermarking others for various reasons. In addition to that, there is also a challenge of miscounting’s as pupils' marks are being added. And so, the International Education Services is stepping in with this system to try and tackle these challenges.

Lack of inclusiveness. Since schools are spread all over the country, some schools are left out or unable to participate in challenges which need people to be physically present. And so, to counter this, with the online option pupils from anywhere across the country are able to participate in this Mathematics challenge and be evaluated accordingly.

Poor evaluation of work done by teachers. Because there has not been a standard assessment and grading system, it has been hard to properly evaluate how teachers are delivering content to the pupils since each teacher sets and marks in their own way. But with this system, teachers can properly be assessed since the setting and grading is not done by them. And school heads can also be able to know how well teachers are doing their work.

# 

The Math Challenge System is aimed at improving pupils’ problem-solving skills using mathematics.

The system is made up a web based-interface for Registered Administrators to upload questions (100) and respective answers into the system therefore a Registered Administrator must upload two excel documents one for questions and the other for corresponding answers into the web system.

The Registered Administrator is also in charge of managing the challenge. At the close of the challenge, a report of answers for all attempted questions are sent to all participants and the first two winners are recognized on the website.

Each participant is automatically sent an email providing them a PDF report of the report of right answers.

The system also contains a command line interface (CLI) for pupils to participate in the competition and School Representatives to interact with the system. Before a pupil is registered, he/she is presented with a menu to input *username, first name, last name, email Address, date of birth, school- registration-number, image file.png* and *view* challenge

*View Challenges*

*View Challenges*displays challenges, which are still valid. If there are more than one valid challenge, a prospective participant will choose which competition they wish to participate in. One can participate in more than one challenge if they wish. If the school registration number does not match registered school numbers, the pupil is informed. If there is a match, the record is added to a file and an email notification is sent to the respective school representative reminding them to confirm the applicant

School representatives log into the system via the command line interface to confirm the newly registered prospective participants via a menu item *view Applicants***.** To reject or activate an applicant, the representative enters a menu item *confirm yes/no username*. If one is rejected, they are deleted from the file and moved into a database table rejected while those that are accepted are deleted from the file and inserted into the participant table in the database. An email notification is sent for all cases. If one tries to register again under the same school after being rejected, they are informed on the command line interface and denied from registering under the school.

Once accepted, the prospective participant can log in and see details of the challenges using *view Challenges* command. If the challenge is open, the participant can issue a command *attempt Challenge* **c***hallenge Number*to start the challenge. Questions are picked randomly from those that were issued. A participant has a maximum of three chances to attempt a challenge and each attempt presents random questions. The questions will be presented one by one and each time a question is presented, the number of remaining questions and time are indicated above the question. If a wrong answer is given by a participant, 3 marks are deducted, while if a participant is not sure, they enter a negative or – and are awarded 0 for that question. Else, the participant gets the marks that were assigned to that particular question. When the time for attempting the question expires, the participant challenge is closed and the participant is given their score and report. The report shows their scores and time taken, for each attempted question and the total time they have taken to complete the challenge.

## software requirements

These include;

* java server
* database management system
* Apache
* web browser

## hardware requirements

* database server
* laptops or desktops
* internet connection

# SYSTEM ARCHITECTURE

## Architectural Design

The system follows a client-server architecture with a web server managing the web interface and a backend server handling the database operations and CLI.

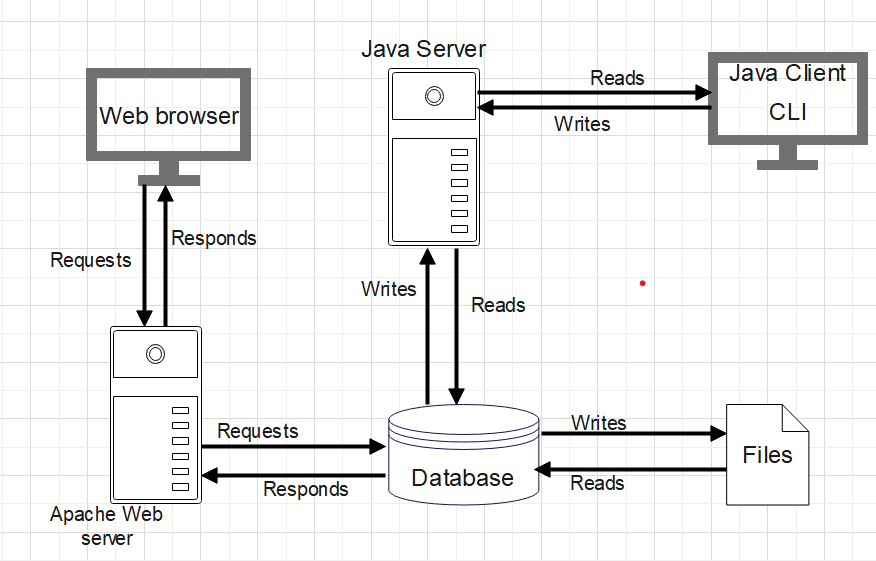


Figure 3.1.1 Architectural Design



**Web Server**: Provides information to other components. It has resources that the students and administrators use.

**Command Line Interface:** It is where the students login and enter their details.

**Internet**: Network over which the Server and the Web interface communicate.

**Database server**: This contains excel documents having the questions and the answers and the participant attempts. It also contains files of the participants and the schools.

**Web Interface:** This is where the analytics are viewed.

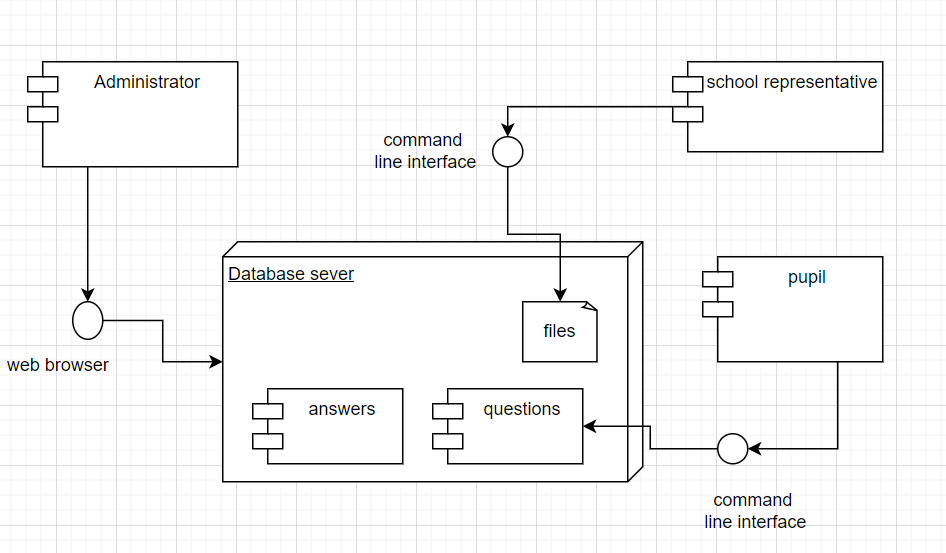


Figure 3.1.2 Component Design

This component diagram illustrates the high-level components and their interactions within the system.

* Command line Interface:

This component represents the user interface elements that interact with the participants, school representatives. It includes interfaces for registration, viewing challenges, attempting challenges, and viewing reports.

* Web Browser and web server component:

The web system component encompasses the backend logic and functionality of the system. It includes modules for handling user registrations, challenge management, question selection, scoring, and report generation.

* Database server component:

The database component stores all the necessary data for the system, including participant information, challenge details, questions, answers, scores, analytics data and many other files. It interacts with both the web browser and the CLI to store and retrieve data as needed.

* Administrator Component:

This represents the functionalities specific to the administrator role. It includes modules for uploading questions and answers, setting challenge parameters, and managing the overall competition.

* Pupil Component:

The pupil component includes modules for participant registration, challenge attempts, scoring, and report generation. It interacts with the web system through the database to access challenges, submit answers, and view results.

* School Representative Component:

This component represents the functionalities specific to school representatives. It includes modules for confirming or rejecting prospective participants, managing applicant records, and interacting with the system on behalf of their respective schools.

## Decomposition Description

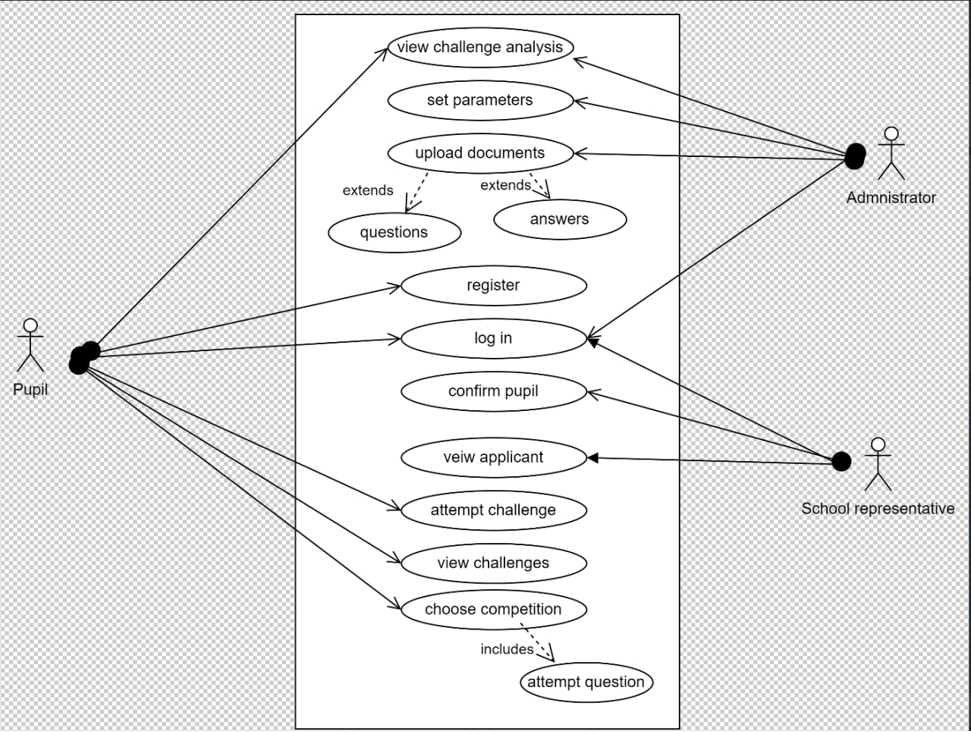


Figure 3.2.1 Use case diagram

The Use Case Diagram shows how the three actors that is; the pupils, the administrators and the school representatives interact with the system. It also shows the different tasks that each of the actors performs.

A primary school pupil registers for the mathematics competition.

* Prospective participant selects the registration option.
* Enters username, first name, last name, email address, date of birth, school registration number, and also uploads an image file.
* Views available challenges.
* Attempts a challenge

If school registration number is valid, the record is added, and an email notification is sent to the school representative for confirmation.

School representatives confirm or reject newly registered pupils.

* School representative logs in and views applicants.
* Chooses to confirm or reject an applicant by entering the username.
* Email notifications are sent for all cases.

The system generates analytics and reports for the competition.

Reports:

* Most correctly answered questions.
* School rankings.
* Performance trends of schools and participants.
* Percentage repetition of questions for participants.
* List of worst and best performing schools.
* Participants with incomplete challenges.

And others

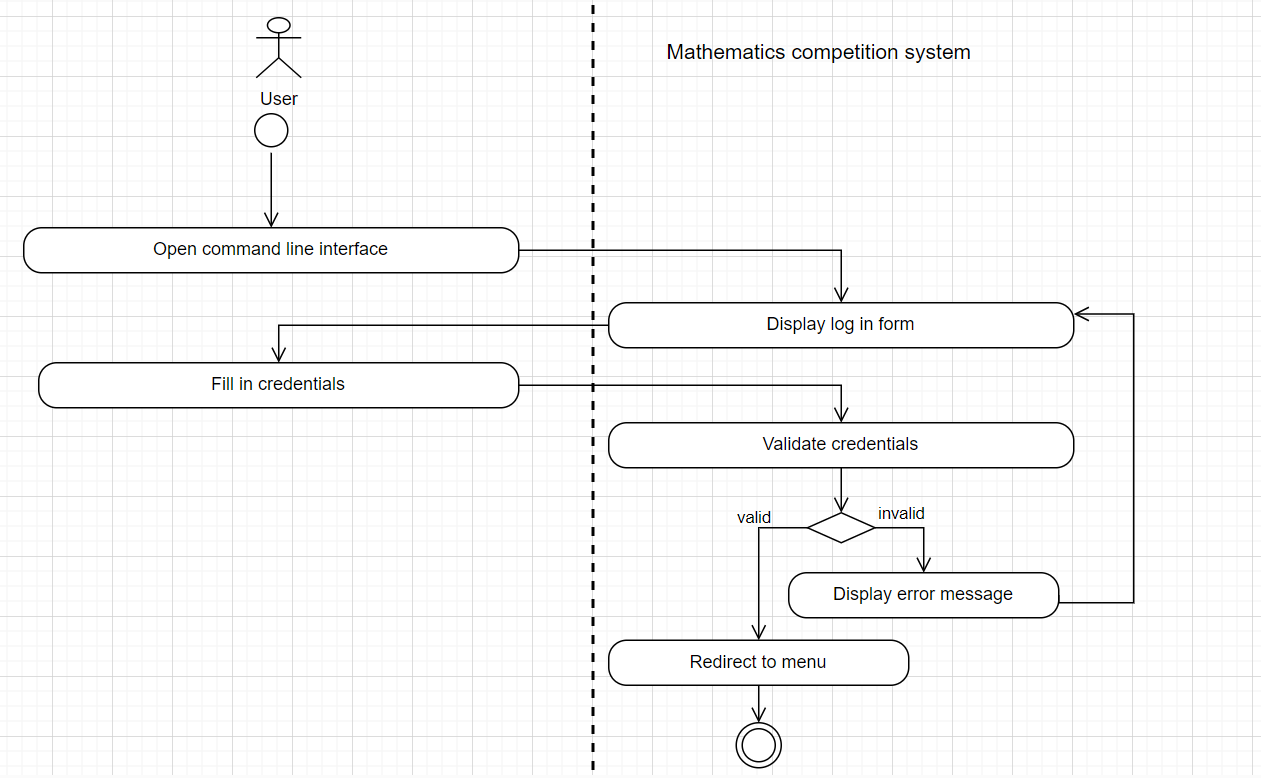


Figure 3.2.2 Activity Diagram of the user

This activity diagram shows the flow of control in the system. It shows the sequence of activities and actions between a general user and the mathematics competition system.

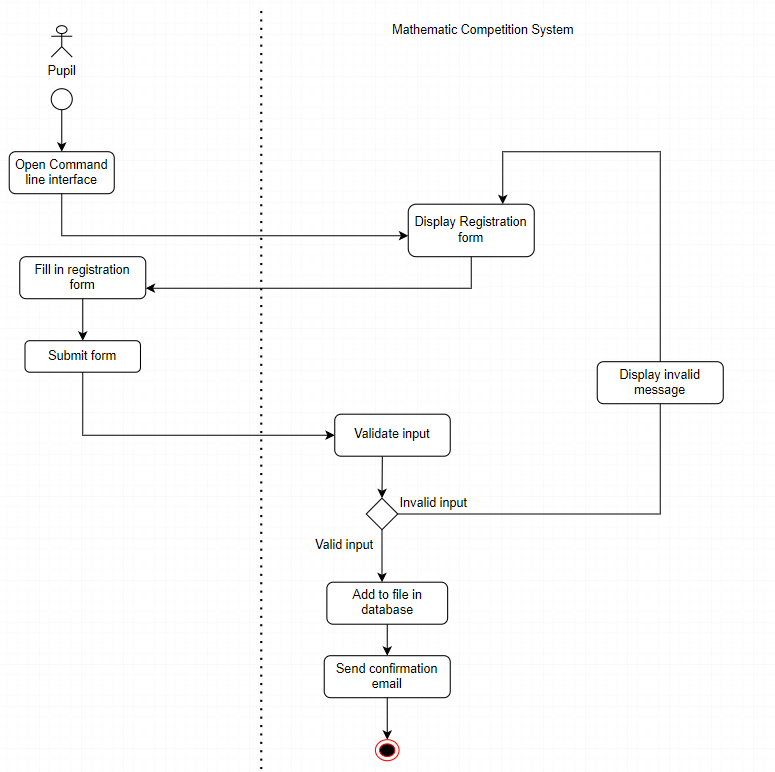


Figure 3.2.3 Activity Diagram of the pupil

This activity Diagram shows the sequence of activities and actions between a pupil and the system when the pupil is registering for the competition.

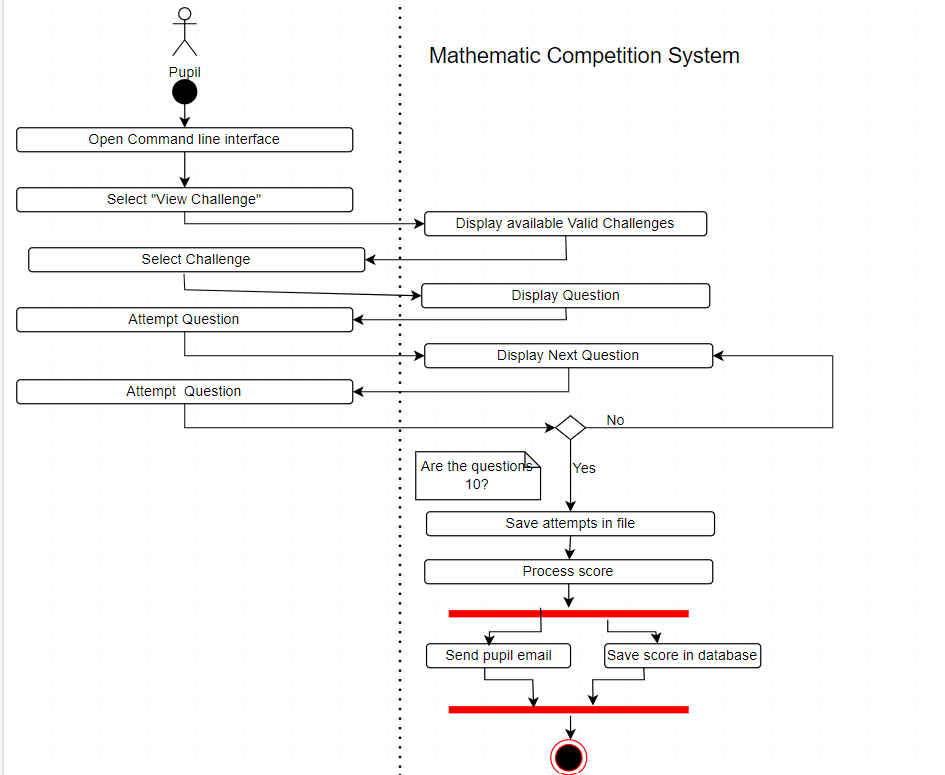


Figure 3.2.4Activity Diagram of the pupil

This Activity Diagram shows the sequence of activities and actions between a pupil and the system when the pupil wants to attempt a challenge.

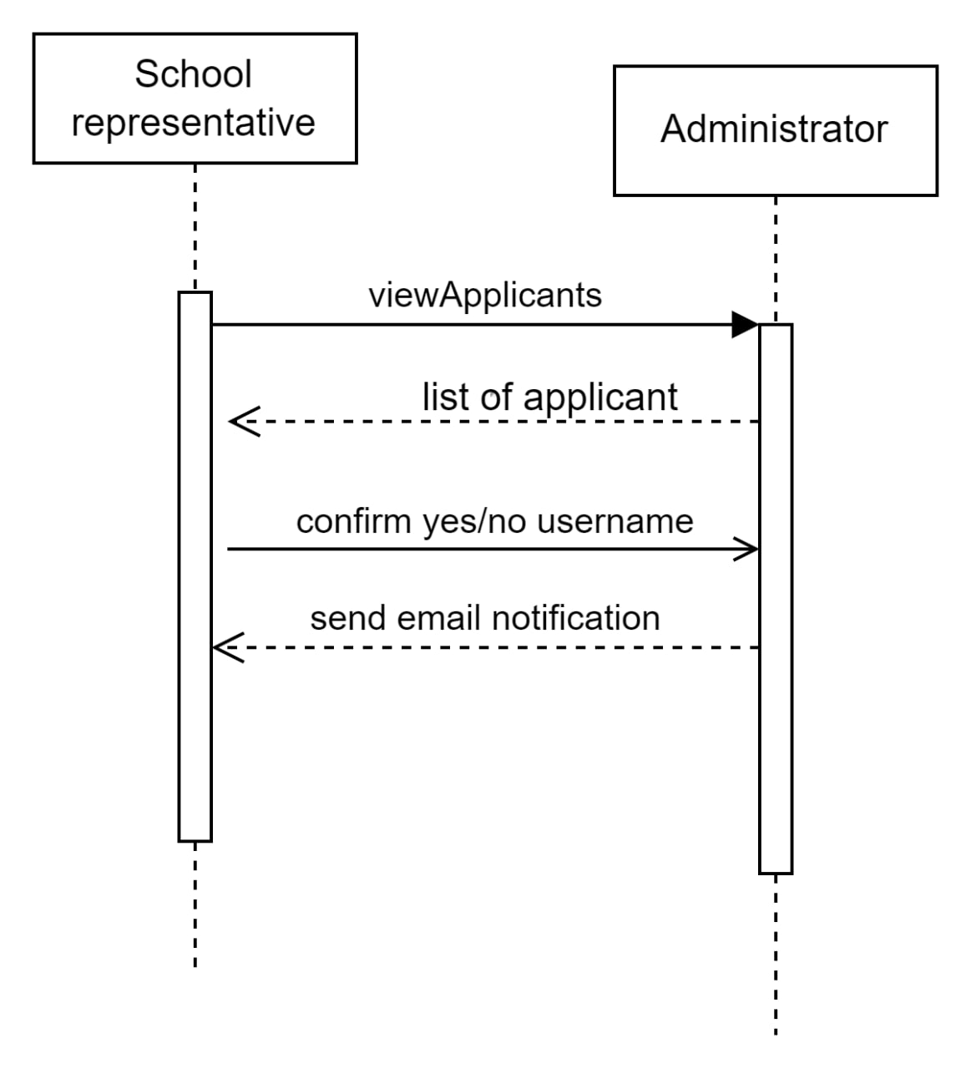


Figure 3.2.5 A Sequence Diagram

The sequence diagram illustrates the sequence of messages between a school representative and an administrator. It shows how the two actors interact in a single use case.

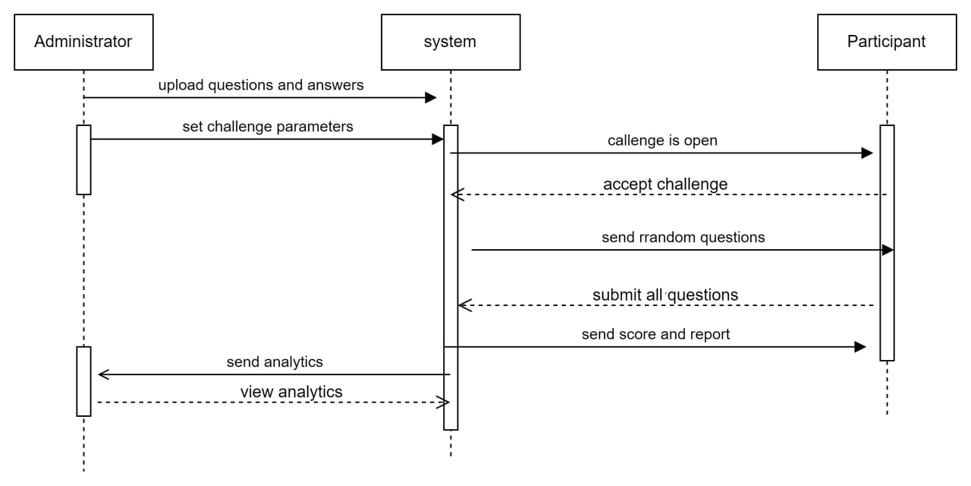


Figure 3.2.6 Sequence Diagram

This sequence diagram shows the sequence of messages between an administrator, the participant(the pupil) and the system. Where by the administrator uploads file containing the questions and the answers, followed by pupils selecting an available challenge.

## Design Rationale

The Architectural Design we opted for is the Client-Server architectural design.

### **Client-Server architectural design**.

Client-Server architectural design is a model in which the server provides resources and services to multiple clients. In this case, we have two servers which are, the Apache Web server and the Java server. And for the clients, we have the web browser, the database, the Command Line Interface and the two excel files.

### Advantages of Client-Server.

We opted for Client-Server mainly for the following reasons;

* Centralized Data and resources. The data and resources are centralized within the system, maintained in a single place, which facilitates easier management and control.
* Better security. This architecture provides better security measures, as the centralized management allows for easier implementation of security measures.
* Enhanced User Interface. It provides a good user interface, making it easier for clients to interact with the system and access the resources they need.

Other architectural designs to consider include Layered architecture and Pipe and filter.

Layered architecture. In this architectural design the units of the system are application are separated into distinct layers, such as the presentation, business logic and data access layers. One of the disadvantages of this design is performance overhead. The communication between layers can introduce performance overhead, especially in cases where data needs to passed through multiple layers.

Pipe and filter. It involves connecting a series of independent components, known as filters, to process a stream of data. The disadvantage with pipe and filter is that there’s limited flexibility in handling dynamic changes and evolving requirements.

# Data Design

## Data Description

Data is stored in a MySQL database. The main entities are:

* Schools
* Participants
* Challenges
* Questions
* Answers
* Attempts
* Reports
* Verified participants

## Data Dictionary

This shows a relational database that will be used to store data in The Math Challenge System.

Below are the different relations that will be used to store data.

Table 1 Schools

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** | **Size** | **Description** |
| district | varchar | Not Null | 20 | District where the school is located |
| email | varchar | Not Null | 30 | Email address |
| name | varchar | Not Null | 25 | Name of the School |
| representativeName | varchar | Not Null | 25 | School Representative’s Name |
| registrationNo | varchar | Alternate Key | 20 | Registration number of the school |
| schoolId | integer | Primary Key | 20 | Uniquely identifies a school |

Table 2 Participants

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** | **Size** | **Description** |
| dateOfBirth | date | Not Null |  | Participant’s date of birth |
| email | varchar | Not Null | 30 | Email address |
| firstName | varchar | Not Null | 12 | Participant’s first name |
| lastName | varchar | Not Null | 12 | Participant’s last name |
| participantId | integer | Primary Key | 20 | Uniquely identifies a pupil |
| password | varchar | Not Null | 10 | Special login access characters |
| schoolId | integer | Foreign Key | 20 | Uniquely identifies a school |
| username | varchar | Not Null | 15 | Participant’s Username |

Table 3 Challenges

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** | **Size** | **Description** |
| challengeId | integer | Primary Key | 20 | Uniquely identifies a challenge |
| duration | integer | Not Null | 4 | Email address |
| endDate | date | Not Null |  | Ending date |
| Name | varchar | Not Null | 25 | Name of the Challenge |
| noOfQuestions | integer | Not Null | 3 | Number of Questions |
| startDate | date | Not Null |  | Starting date |

Table 4 Questions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** | **Size** | **Description** |
| challengeId | varchar | Foreign Key | 20 | Uniquely identifies a challenge |
| marks | varchar | Not Null | 3 | Marks allocated |
| questionId | integer | Primary Key | 10 | Uniquely identifies a question |
| questionText | varchar | Not Null | 255 | Question |

Table 5 Answers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** | **Size** | **Description** |
| answerId | integer | Primary Key | 20 | Uniquely identifies an answer |
| isCorrect | Boolean | Null |  | True / false |
| questionId | integer | Foreign Key | 20 | Uniquely identifies a question |
| text | varchar | Not Null | 100 | The answer |

Table 6 Attempts

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** | **Size** | **Description** |
| attemptId | integer | Primary Key | 20 | Uniquely identifies an attempt |
| participantId | integer | Foreign Key | 20 | Uniquely identifies a pupil |
| challengeId | varchar | Foreign Key | 20 | Uniquely identifies a challenge |
| score | integer | Not Null | 3 | Marks obtained by the pupil on an attempt |
| timeTaken | integer | Not Null | 2 | Time used for the attempt in minutes |

Table 7 Reports

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** | **Size** | **Description** |
| reportId | integer | Primary Key | 20 | Uniquely identifies a report |
| participantId | integer | Foreign Key | 20 | Uniquely identifies a pupil |
| challengeId | varchar | Foreign Key | 20 | Uniquely identifies a challenge |

Table 8 Verified Participant

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** | **Size** | **Description** |
| chances | integer | Not Null | 2 | Allowable attempts left for a particular user |
| remainingQuestions | integer | Not Null | 2 | Questions left for the participant to complete a challenge |
| remainingTime | time | Not Null |  | Time left for a challenge session to end |
| participantId | integer | Foreign Key | 20 | Uniquely identifies a pupil |

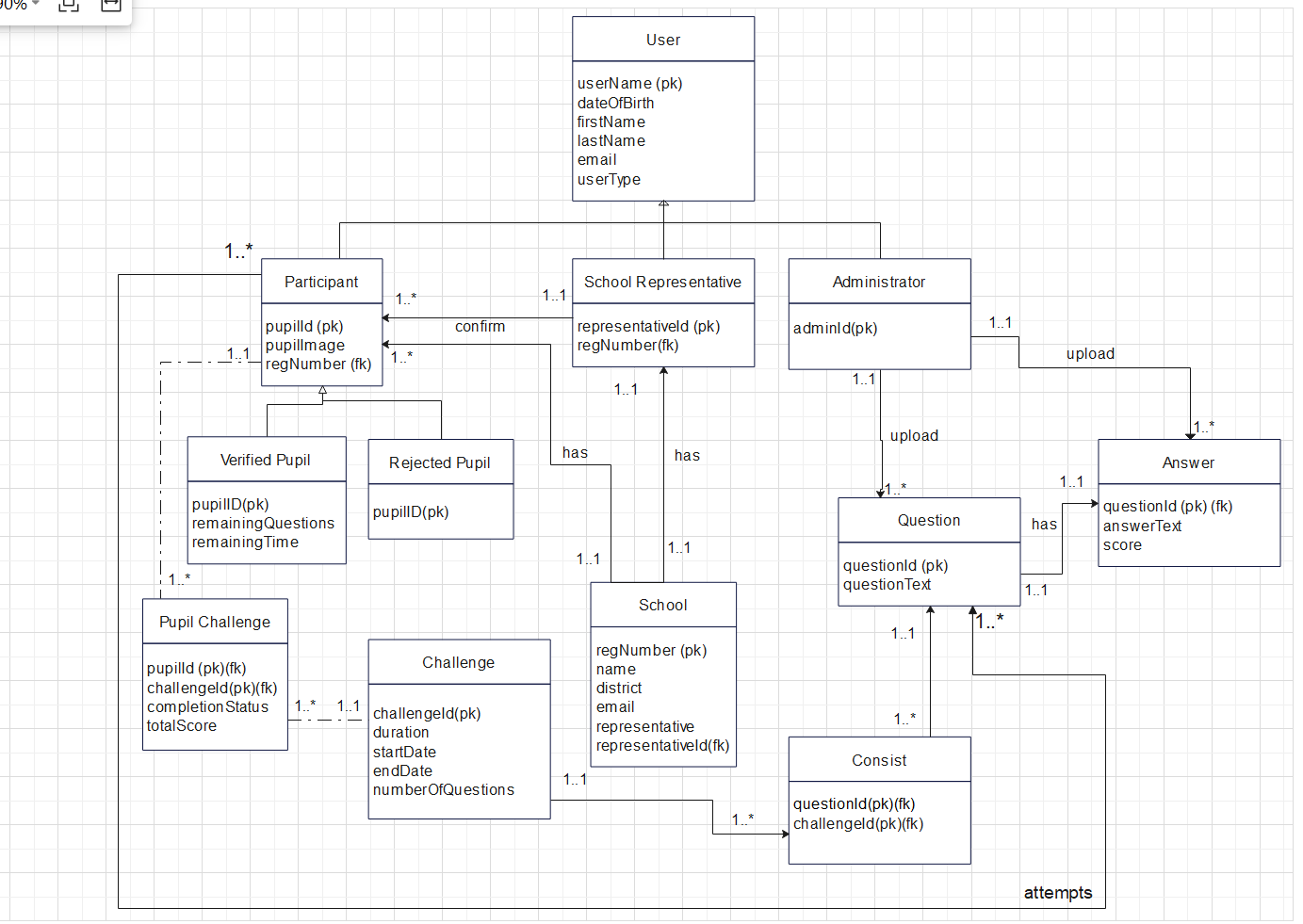


Figure 4.2.1 Entity relationship diagram for the database structure

# COMPONENT DESIGN

## Web Interface Module

### Administrator:

* login(credentials):

If authenticate(credentials) is True

Print "Login Successful"

Else

Print "Invalid Credentials"

EndIf

EndFunction

* Function uploadSchoolDocuments(documentDetails)

If isLoggedIn(admin) is True

uploadDocument(documentDetails)

Print "School Documents Uploaded Successfully"

Else

Print "Please login to upload documents"

EndIf

EndFunction

* Function confirmParticipants(participantDetails)

If isLoggedIn(admin) is True

For each participant in participantDetails

If validate(participant) is True

confirm(participant)

Print "Participant Confirmed"

Else

Print "Invalid Participant Details"

EndIf

EndFor

Else

Print "Please login to confirm participants"

EndIf

EndFunction

* Function register(adminDetails)

If isValid(adminDetails) is True

addAdmin(adminDetails)

Print "Administrator Registered Successfully"

Else

Print "Invalid Administrator Details"

EndIf

EndFunction

## CLI Module

### School Representative:

* Log In(credentials):

Function logIn(credentials)

If authenticate(credentials) is True

Print "Login Successful"

Else

Print "Invalid Credentials"

EndIf

EndFunction

* Upload Excel Documents(excel File Details):

Function uploadExcelDocuments(excelFileDetails)

If isLoggedIn(rep) is True

uploadDocument(excelFileDetails)

Print "Excel Documents Uploaded Successfully"

Else

Print "Please login to upload documents"

EndIf

EndFunction

### Participant:

* register(participant Details):

Function register(participantDetails)

If isValid(participantDetails) is True

addParticipant(participantDetails)

sendEmailToRepresentative(participantDetails.school)

Print "Participant Registered Successfully"

Else

Print "Invalid Participant Details"

EndIf

EndFunction

* View Challenges():

Function viewChallenges()

challenges = getOpenChallenges()

Display(challenges)

EndFunction

* Attempt Challenge(challenge Details):

Function attemptChallenge(challengeDetails)

If isLoggedIn(participant) is True

challenge = getChallenge(challengeDetails)

StartTimer()

For each question in challenge

DisplayQuestion(question)

answer = getAnswer()

If answer is correct

score += getQuestionMarks(question)

ElseIf answer is '-'

score += 0

Else

score -= 3

EndIf

updateTimer()

If timerExpired() Then

break

EndIf

EndFor

saveScore(participant, score)

sendReport(participant, score)

Else

Print "Please login to attempt challenge"

EndIf

EndFunction

# HUMAN INTERFACE DESIGN

## Overview of User Interface

### Pupil interactivity

The participants are to use the command line interface to register for the challenge

The pupils also type the “ viewChallenge” command in order to see and then select the one they are to attempt.

### School representative interactivity

Use the command line interface to view all applicants and confirm the participating pupils

Use the web browser to view their score.

### Administrator

Use the command line interface to set parameters and also upload the documents containing the questions and the answers

The administer uses the web browser to view the challenge analytics

## Screen Images

### Pupil interactivity

The participants are to use the command line interface to register for the challenge

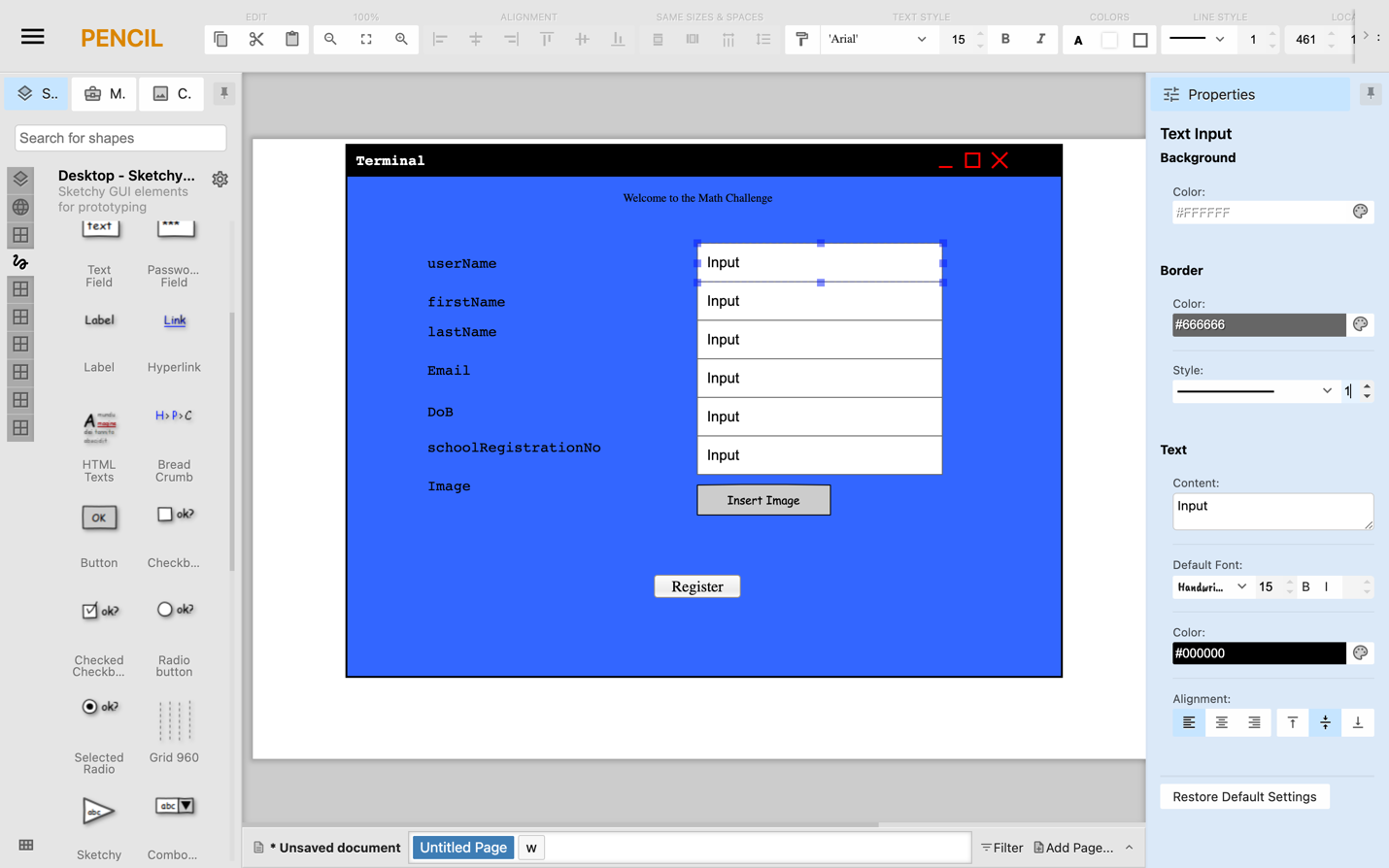


Figure 6.2.1 Pupil Registration Form

For a pupil to participate in the competition, they should register using a command line interface

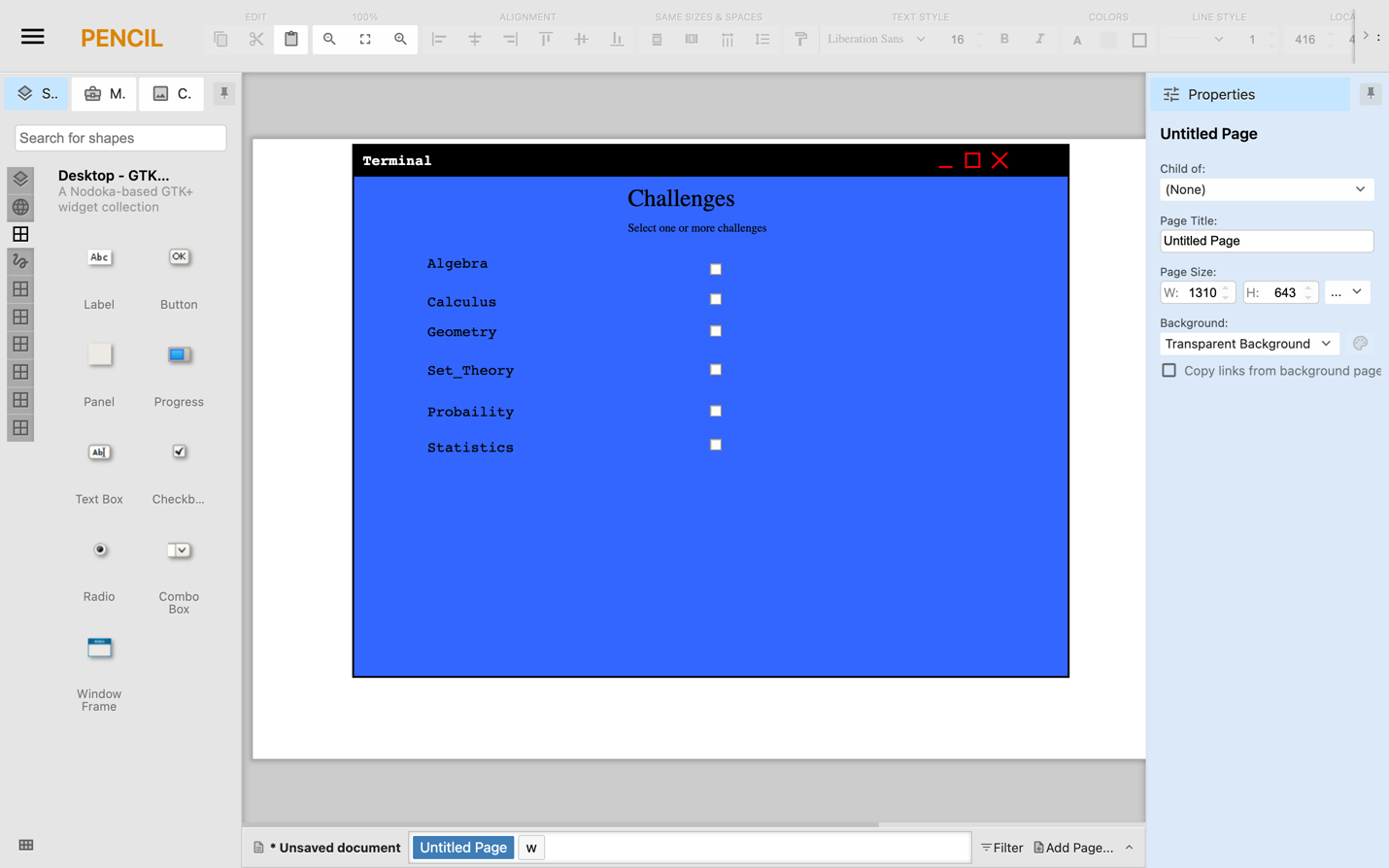


Figure 6.2.2 Challenges list

The pupils also type the “ viewChallenge” command in order to see and then select the one they are to attempt.

### School representative interactivity

Use the command line interface to view all applicants and confirm the participating pupils

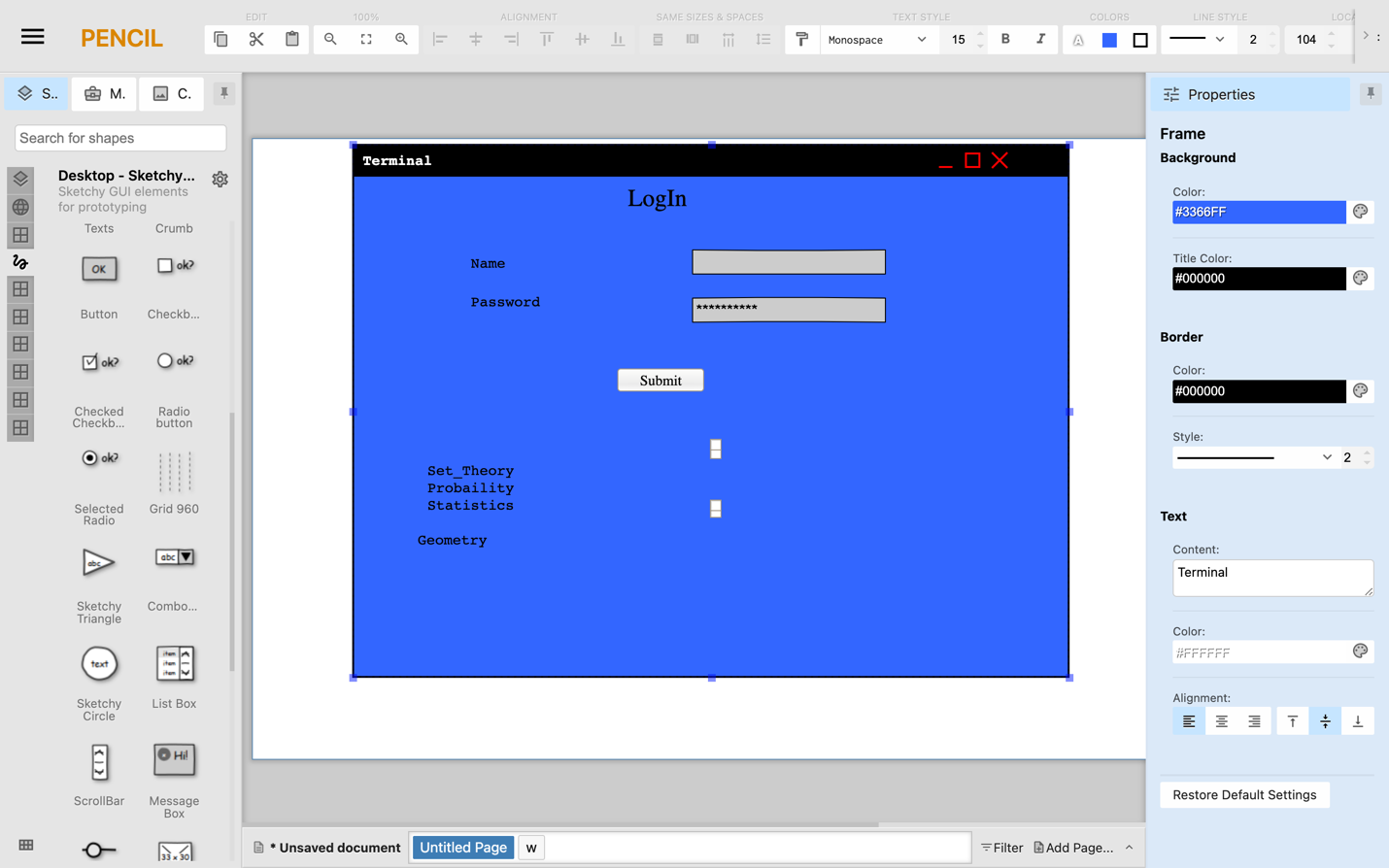


Figure 6.2.3 Representative login

School representatives log into the system via the command line interface

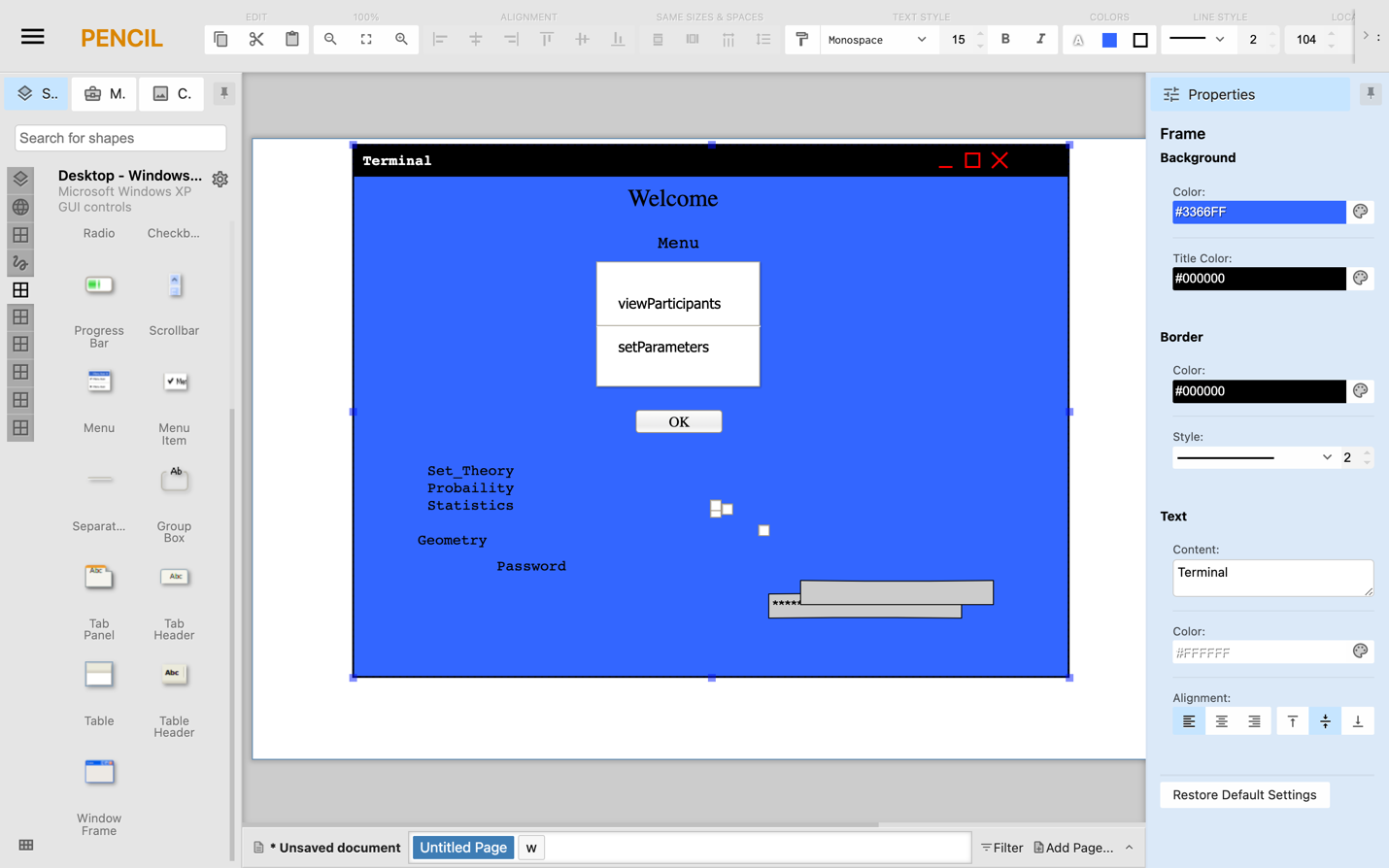


Figure 6.2.4 Representative Menu

To confirm the newly registered prospective participants school representatives a menu item **viewApplicants**

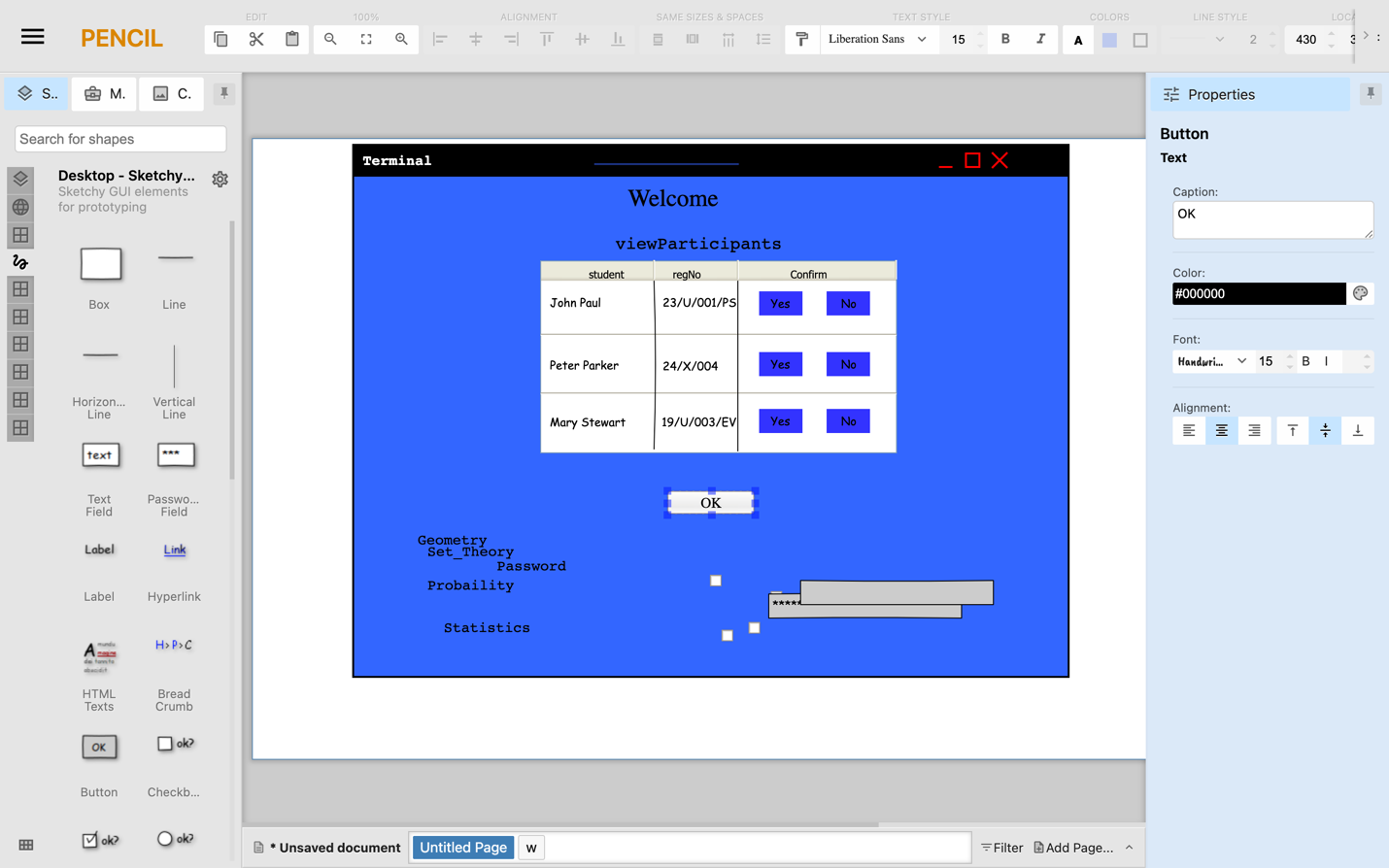


Figure 6.2.5 Confirming Participants

To reject or activate an applicant, the representative enters a menu item **confirm yes/no username**.

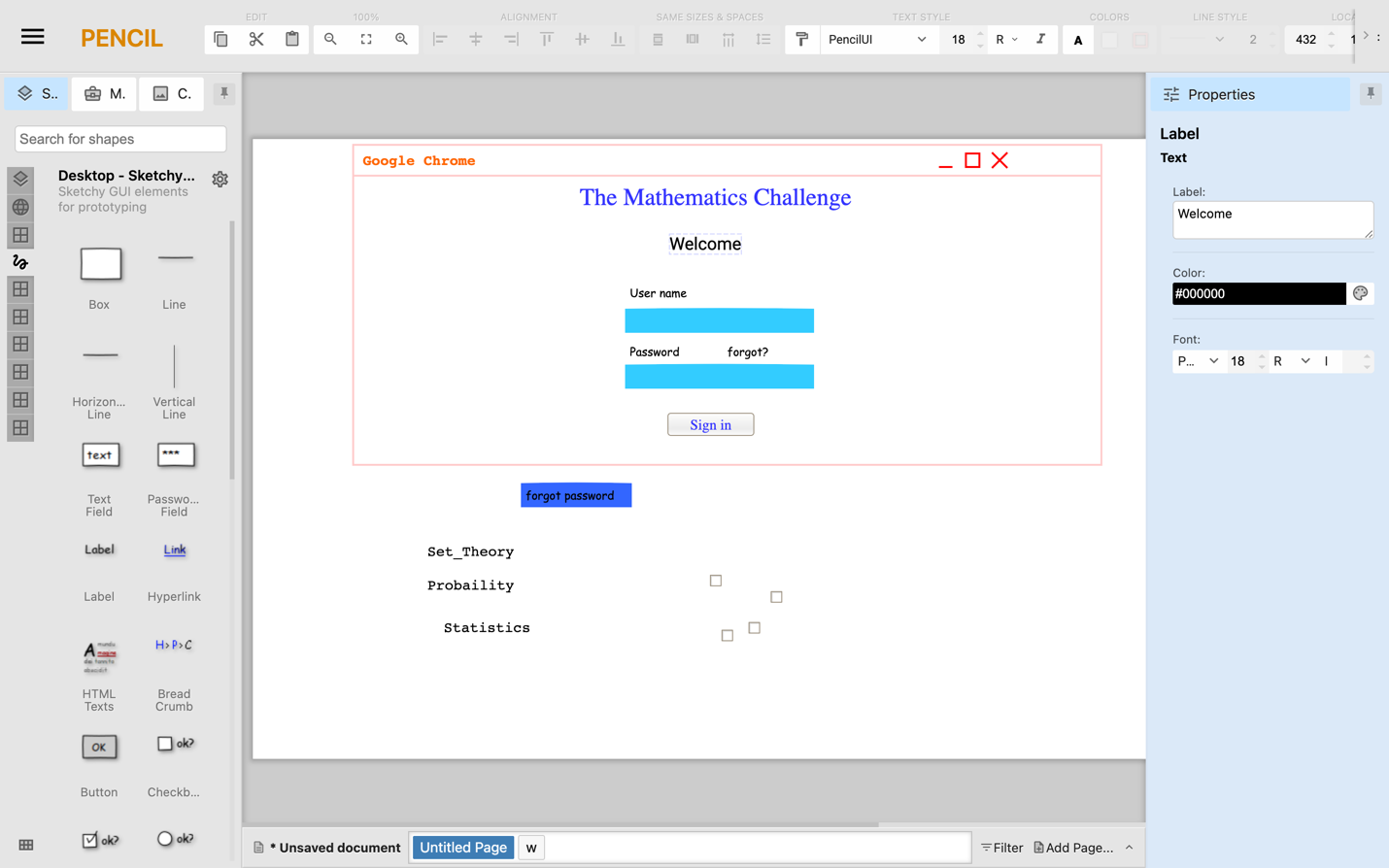


Figure 6.2.6 School Representative sign in

The School Representative uses the web browser to view pupil’s score.

### Administrator

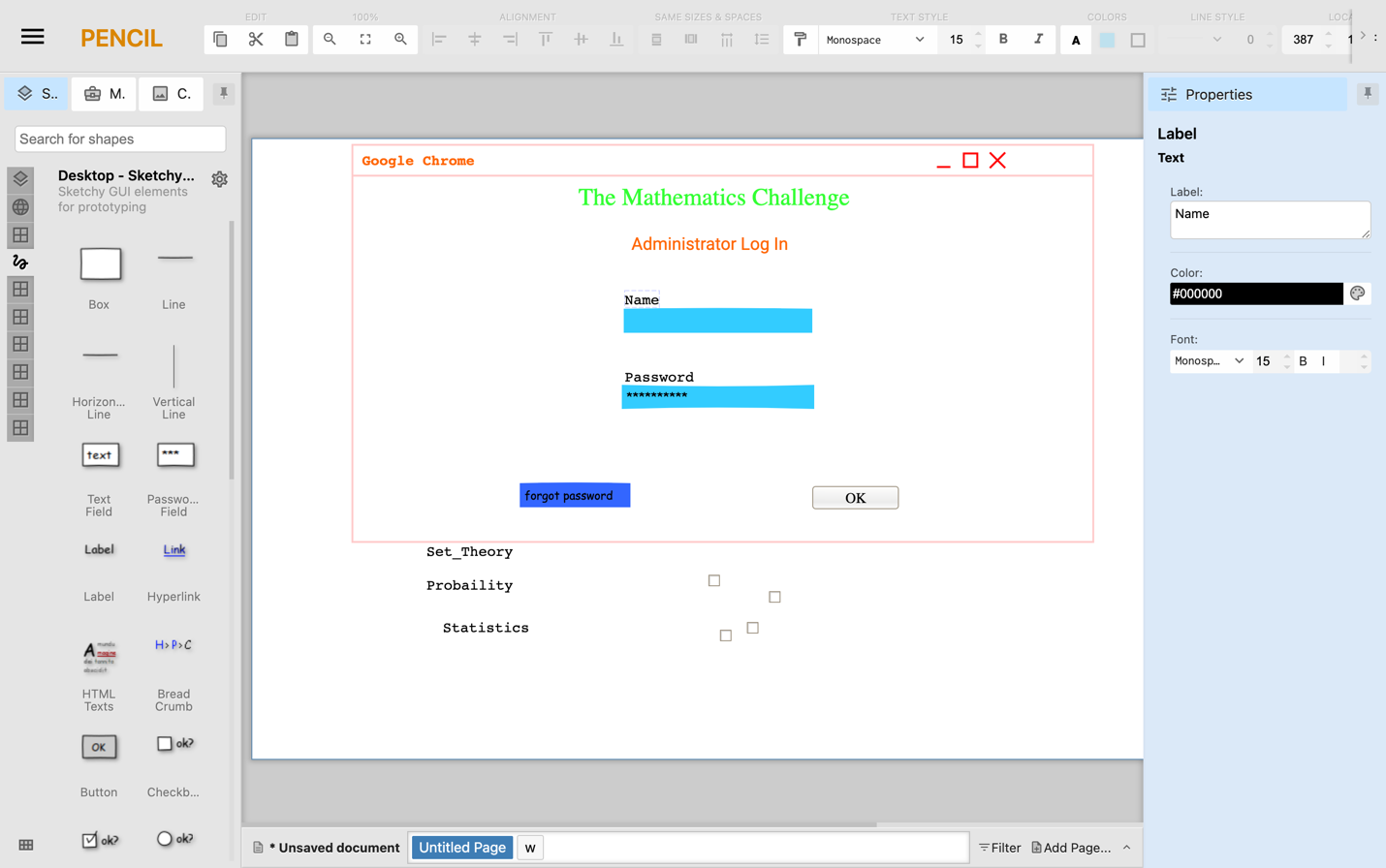


Figure 6.2.7 Administrator Sign in

The administrator signs into the webpage

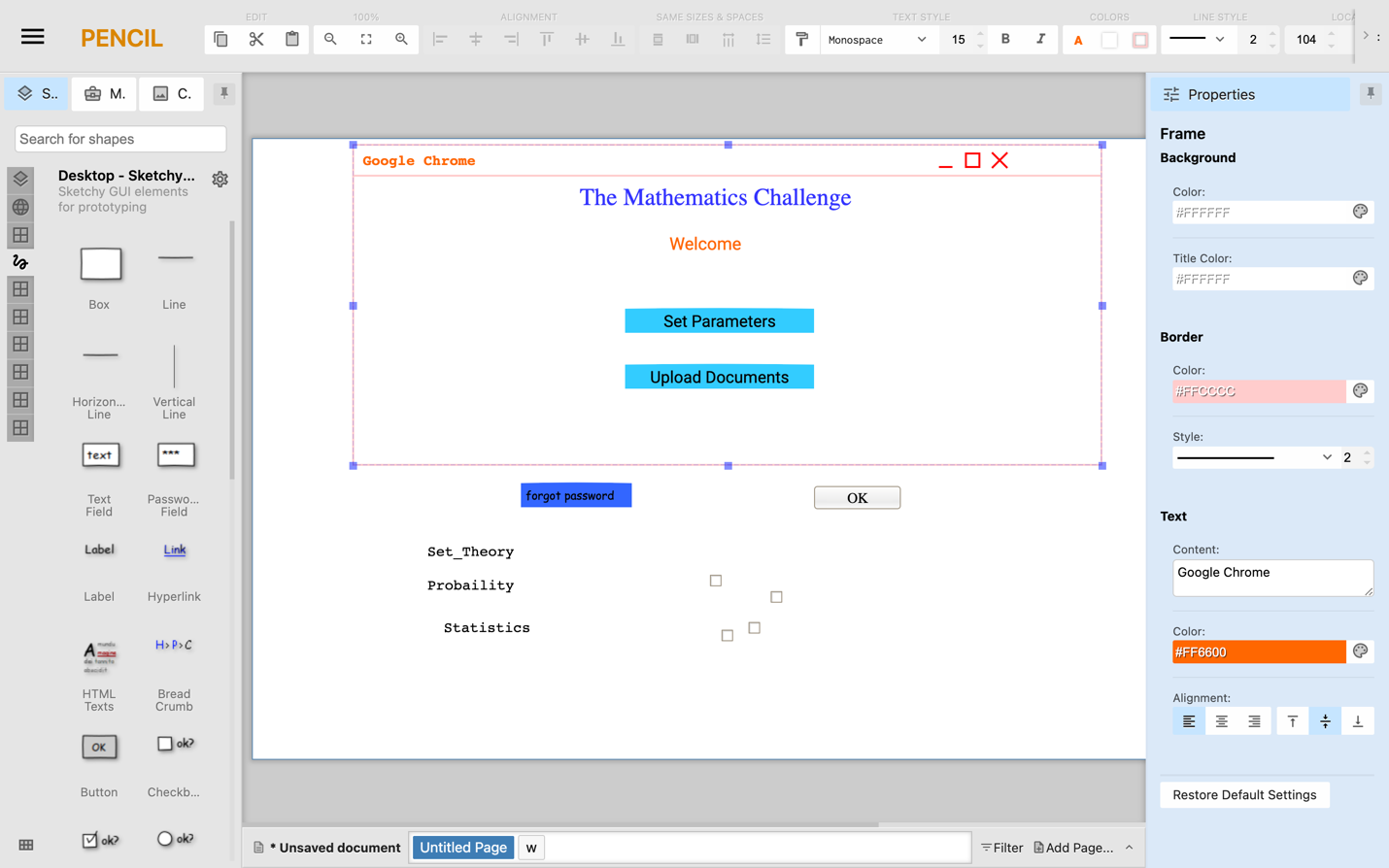


Figure 6.2.8 Administrator homepage

The administrator uses the web browser to set parameters and also upload the documents containing the questions and the answers

The administrator uses the web browser to view the challenge analytics

## Screen Objects and Actions

Table 9 Screen Images

|  |  |  |
| --- | --- | --- |
| **Object** | **Action** | **Result** |
| Submit | Click | A request is sent to the server to validate entered credentials |
| Register | Click | A request is issued to the database to save the entered credentials |
| Sign in | Click | A request is issued to the database to verify the input credentials |
| Insert Image | Click | Opens the library to give access to images |
| Forgot password | Click | Opens a link to recover or set new password |
| Set Parameters | Click | Enables setting of starting time, Ending time and duration of the challenge |
| Upload Documents | Click | Opens a window to insert documents to be uploaded to the database |