KINDERCARE CHARACTER DRAW SYSTEMS SOFTWARE DESIGN DOCUMENT

PROJECT TEAM NAME: GROUP -34

PROJECT MEMBERS

NAME	REGISTRATION NUMBER	**SUPERVISOR**
Mujabi Eriya Samson	20/U/7793/PS	
Kakoonye Anderson	20/U/2061/EVE	**Brian Muchake**
Kagimu Edward	16/U/5146/EVE	
Ademun Genevieve	20/U/2021/EVE	

PROJECT TITLE: KINDERCARE CHARACTER DRAW:

DATE: 21st January, 2022

Table of Content

Appendices

Table of Figures

1. INTRODUCTION

1.1 Purpose

This software design document describes the architecture and system design of **KinderCare Character Draw** which will help lower school children remember and create the different shapes of characters, ranging from A to Z (All in capital). The system will be used by teachers to Assign work to the students and students through the command line interface respond to the assignments according to the systems requirements stated in the different modules.

1.2 Scope

The system will be primarily used by the Teachers at the school and in different locations according to where the teachers will be and also students according to where and how each of them can access the system especially via the commandline

Goals/objectives

- To enable pupils learn characters simply in specific (A-Z) in capital.
- To enable teachers register pupils into the system.
- To enable teachers give assignments to pupils.
- To enable pupils attempt assignments.
- To enable teachers to activate and deactivate pupils if in need

Benefits of the project

- The project will increase accountability within the Kindercare character draw system by enabling the teachers monitor the pupils details, registering them, uploading an assignment for them to attempt and authorizing them to attempt an assignment, which is marked automatically by the system hence reducing corruption in awarding scores.
- The project also helps to reduce the teachers work load, involved in marking assignments manually, since this is all done by the system.
- The system will permit efficient tracking of pupils details (registration and authorization to attempt assignment) and scores.
- The system facilitates faster learning among pupils, since they will able to remember and create different shapes.

1.3 Overview

This system design document provides an insight of how the proposed system, Kindercare character draw system is to be physically realized clearly stating the development process and activities (how the user's requirements are methodologically transformed into a fully functional system, the development team and each member role of the development process.

The document entails a data dictionary which provides description of the terminologies used in the system that may not be familiar to the teachers and the pupils.

The document also clearly describes the functionality of the intended Kindercare character draw system its impact and benefits on the pupils and the teachers

1.4 Reference Material

The Kindercare character draws system study case.

Source: https://muele.mak.ac.ug/pluginfile.php/831747/mod_resource/content/1/Assignment.docx

1.5 Definitions and Acronyms

- **Pupil –** Is an entity within the system responsible for attempting assignments
- **Teacher** – is an entity within the system responsible for registering pupils, submitting an assignment, authorizing the pupils to attempt assignment and responding to pupils' request.
- **Viewall** -Displays assignment number and date , showing if attempted or not
- **Checkstatus** -Displays the status report of the pupil summarizing all assignments (how many attempted, average score, percentage missed, percentage attempted etc.))
- **Viewassignment (assignmentid)** – to see details of a specified assignment
- **Checkdates (datefrom, dateto)** -shows if there is an assignment within a specified date range
- **RequestActivation** – used for pupil to request teacher to activate him or her

2. System Overviews

The system is going to manage the KinderCare Character Draw pupils' activities. It considers whether a pupil is registered and activated to do whatever he she is commanded to do or not. If the pupil is not activated, he she sends a request to the teacher to reactivate them and therefore go on with everything such as assignments given to them.

The system consists of a database where information is kept and accessed at the time of need by the authorized users ie (the teacher to activate and register the pupil and the pupil to view and attempt the assignments) and the different application modules.

The system is designed in such a way that the pupils access it through the **welcome page** which on clicking on the Link, takes them straight to the commandline interface where they are supposed to attempt assignments and do every other thing such as entering their ID numbers for the system to

accept them attempt the assignments, the ID numbers are given to each of them by their respective teachers . For example; -

1) Once a pupil attempts and completes the assignment, the system automatically awards the marks according to the status of the attempted number ie whether wrong or right.

2) The teacher is supposed to give a maximum of eight(8) characters per assignment. The system checks whether they are 8 characters and then allows the teacher to submit it, if they are beyond 8, it prompts the teacher to reduce on the number of characters.
3) The on attempt will produce stars for every number attempted and these will follow a 4X7 matrix.

Teachers can at any time activate or deactivate a student according to any available circumstance.
The assignment to be given will be in form characters from A up to Z.
The pupils' information recorded includes; - First Name, Last Name, Phone and User Id.
The system automatically awards marks to the pupil on attempting assignments and also a comment from the teacher according to the way the pupil does the work.
System records the pupils' progress and saves it for purposes of making report by the end of the study period by the teacher.

The system has a component of command line client program which is used by the pupil to attempt the assignment submitted by the teacher and to send request for activation incase a pupil is deactivated by the teacher and if reactivated, the pupil is free to attempt,
After attempting the last character, the total amount of time taken to attempt the assignment is provided for the pupil to see. An acknowledgement of submitting the assignment is also provided for the pupil to see and all characters attempted are drawn on the screen (in form of stars as in figure 1). The method of providing a solution to the character is as below. The system generates characters like in figure below, by placing (*) in the specific locations of the matrix (4X7). The system keeps a record of the right locations of the stars (*) and bases on this to award a score. The pupil attempts the assignment by entering either a 0 or a 1. and 0 means the star should not be printed in the cell while a 1 means that the start will be printed in the cell like in the figure below.

		*	*
	*		
*			
*			
*			
	*		
		*	*

On loging in by the students, the following commands are used to invoke the different activities.

1) To Display the assignment number and Date, showing attempted or not,
Command is: **Viewall**
2) To Displays the status report of the pupil and summarizing all assignments (how many attempted, average score, percentage missed, percentage attempted etc.))
Command is: **CheckStatus**

3) To see the details of a specified assignment,

command: ***Viewassignment assignmentid***

4) To see if there is an assignment within the specified date range

command: Checkdate dateto datefrom

5) To request a teacher to activate a pupil,

command***: RequestActivation.***

Every 5 minutes a scheduled job runs in the background checking if any assignment information added.

The system web application helps the teacher to do the following;

- To give the assignment to the pupils
- To Register the pupils with the following details: first name. Last name, phone number and the used code.
- To activate a pupil at any given time and activate them on request.
- Attaching score to the assignment which is which is shown when the student next logs in.
- To view the necessary reports that can help him or her in decision making.

3. SYSTEM ARCHITECTURE

The system registers and keeps records of all KinderCare Character Draw Pupils and their teachers. The pupils are registered through the Web interface and the teacher is able to give assignments to pupils through the same interface.

The system also allows Pupils to attempt the Assignments given by the teacher as long as they are still within the required time and they are registered. Teachers are allowed by the system to deactivate the pupils at a given time and reactivate them.

3.1 ARCHITECTURAL DESIGN

figure1

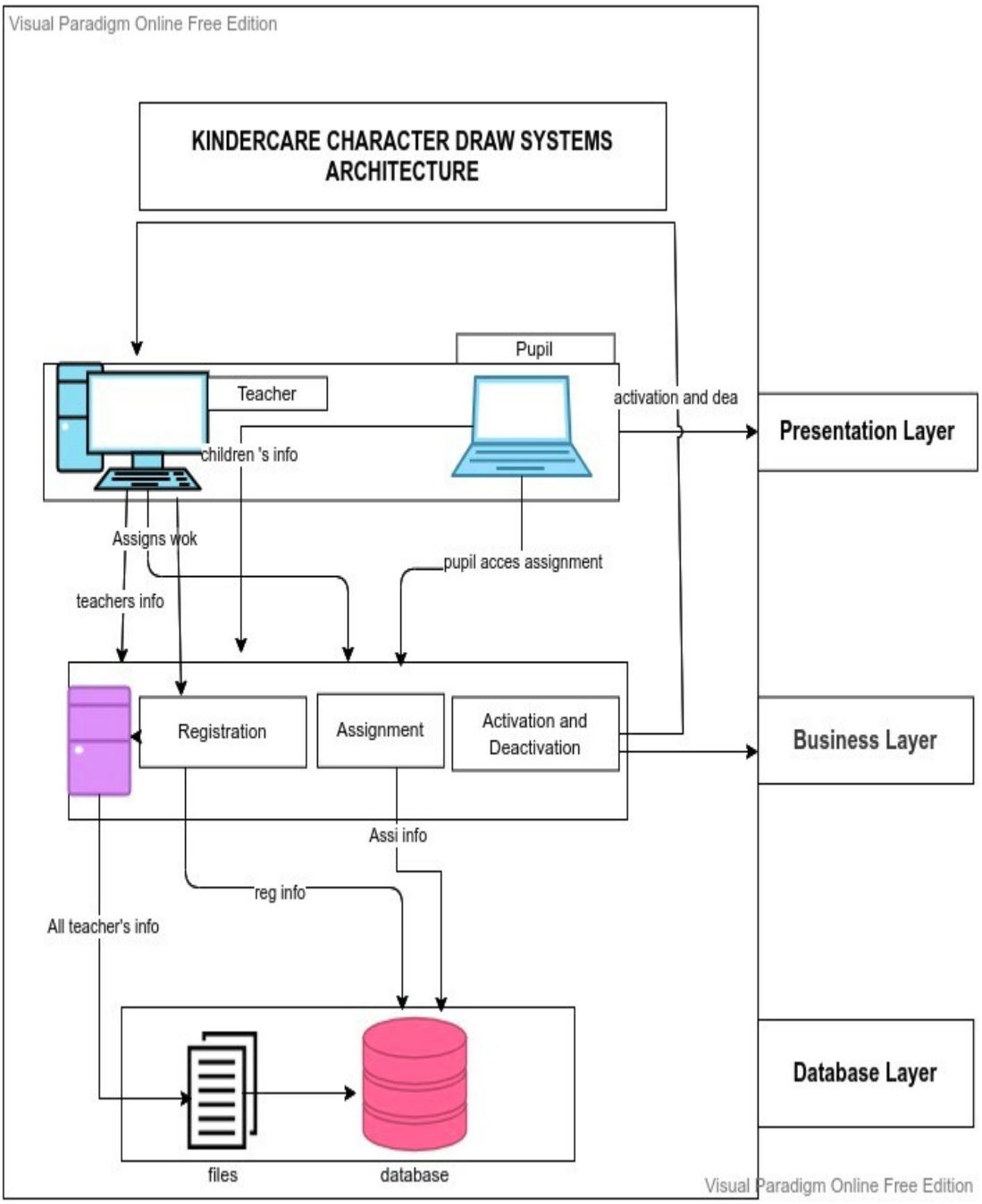

Figure 1: Layered Architecture

The image above shows the layered architecture of the **KinderCare Character Draw System**. The architecture divides the system into three layers namely;

The presentation layer: This layer is responsible for handling all user interface and browser communication logic. We have the client computer used by KinderCare Teachers to register new pupils with the aid of a client web interface that sends the data to the socket c program. The information entered by the teachers about the pupils is viewed by the respective pupils who are registered, through socket c program that sends the data to the socket server c program that is located in the business and persistence layer.
We again stress the point of command line interface which basically provides all the functionality necessary for the the pupils to interract with the system by viewing their

registration status, Assignments to attempt, sending claims to the teacher to re-activte a deactvated pupil.

The business and persistence layer: This layer is responsible for executing specific business rules associated with the request and it coordinates the objects and services of your application. This is where the server program is located to receive pupils info from the the teachers and then write them to a text file so it can be uploaded to the database. We also have the registration subsystem that handles the registration process of the pupils by the Teacher and then save their info to the database in the database layer.Finally, we have the Assignment subsytem that handles the assignments given by the teacher to the pupils that each pupil must attempt and submit back.

The database layer: This layer contains the tables, indexes, and data managed by each of the modules. We have the file.txt file which is put together by the server program containing all the information of all the pupils submitted by the teachers and then uploaded to the database for storage. The database stores all the information of the system such as pupils, teachers information, assignments.

3.2 Decomposition Description

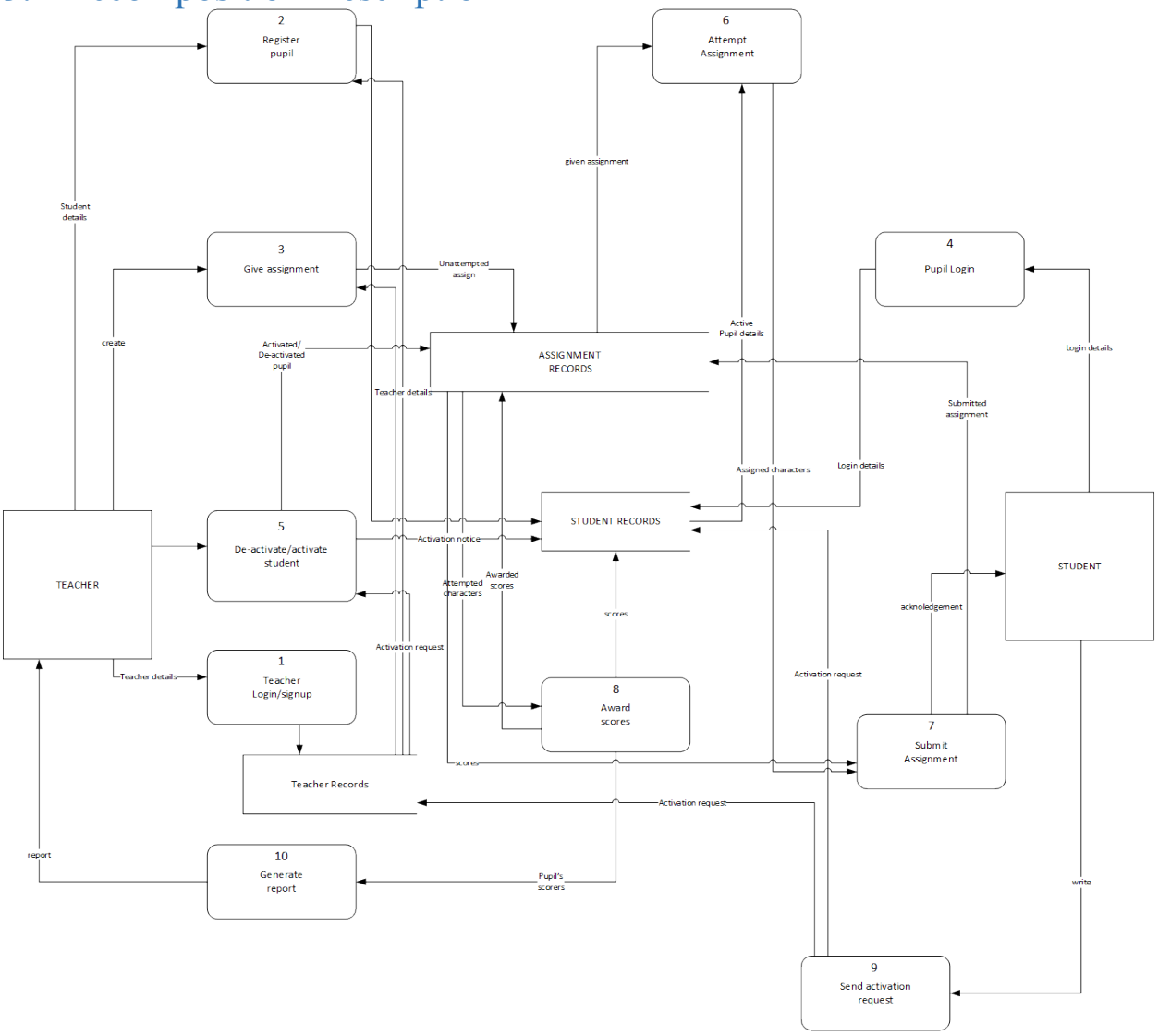

figure 2: level 0 dataflow diagram

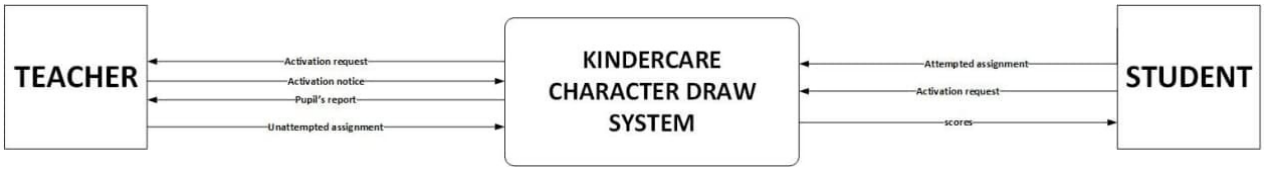

figure 3: context diagram.

3.3 DESIGN RATIONALE

REASONS FOR USING LAYERED ARCHITECTURE IN Figure1

- It provides modularity and clear interfaces as the system is divided into layers such as the presentation layer, business layer, persistence layer and the database layer.
- Implementation, simplicity, maintainability, flexibility, and scalability are maintained.
- The layered approach supports portability.
- It provides robustness and preserves stability.
- It is a time-tested approach and therefore reliable and errors are hardly made.

DISADVANTAGES OF USING LAYERED ARCHITECTURE IN Figure1

- Data overhead and processing, due to the duplication of functionality such the teacher registering different pupils at ago.
- The more layers you have the more risks you have of things to breakdown or data to get lost.
- Various issues regarding higher layers verses lower layers.
- Results in complex exploitation of user intensive applications.
- It leads to sluggish operation modes on various applications.

OTHER ARCHITECTURE THAT WERE FOREGONE

1) **Even-driven architecture**:
   Is a software paradigm promoting the production, detection, consumption of, and reaction to events. This architectural pattern may be applied by the design and implementation of applications and systems that transmit events among loosely coupled software components and services.

Why it was not chosen

1. Testing can be complex if the modules can affect each other. While individual modules can be tested independently, the interactions between them can only be tested in a fully functioning system.
2. Error handling can be difficult to structure, especially when several modules must handle the same events.

3. When modules fail, the central unit must have a backup plan.
4. Messaging overhead can slow down processing speed, especially when the central unit must buffer messages that arrive in bursts.
5. Developing a system-wide data structure for events can be complex when the events have very different needs.
6. Maintaining a transaction-based mechanism for consistency is difficult because the modules are so decoupled and independent.

- **Microkernel architecture**: The **microkernel architecture** pattern (sometimes referred to as the plug-in **architecture** pattern) is a natural pattern for implementing product-based applications.

Why it was not chosen

1. Deciding what belongs in the microkernel is often an art. It ought to hold the code that's used frequently.
2. The plug-ins must include a fair amount of handshaking code so the microkernel is aware that the plug-in is installed and ready to work.
3. Modifying the microkernel can be very difficult or even impossible once a number of plug-ins depend upon it. The only solution is to modify the plug-ins too.
4. Choosing the right granularity for the kernel functions is difficult to do in advance but almost impossible to change later in the game.

- M**icroservices architecture pattern** addresses these issues by separating the application into multiple deployable units (service components) that can be individually developed, tested, and deployed independent of other service components.

Why it was not chosen

1. The services must be largely independent or else interaction can cause the cloud to become imbalanced.
2. Not all applications have tasks that can't be easily split into independent units.
3. Performance can suffer when tasks are spread out between different microservices. The communication costs can be significant.
4. Too many microservices can confuse users as parts of the web page appear much later than others.

4. DATA DESIGN

Data design describes how the information of the Kindercare character draw system is stored in the database or storage area.

4.1 Data Description

The database of Kindercare character draw system has three tables; - pupil, assignments and teacher tables.

Pupil table

Attribute	**Type**
Usercode	int
First name	Varchar(25)
Last name	Varchar(25)
Phone number	Int

Assignment table

Attribute	**Type**
AssignmentId	int
Start_time	date
End_time	Date
scores	int
comment	Varchar(50)
UserCode	int

Teacher table

Attribute	**Type**
TeacherID	Int
AssignmentId	Int
UserCode	Int

4.2 Data Dictionary

Pupil table

Attribute	**Type**	**Description**
Usercode	int	Unique, automatically generated numbers to differentiate between pupils
First name	Varchar(25)	First name of pupil
Last name	Varchar(25)	Last name of pupil
Phone number	Int	Pupil's phone number

Assignment table

Attribute	Type	Description
AssignmentId	int	Uniquely identifies an assignment and is automatically generated.
Start_time	date	Indicates the time when an assignment commences
End_time	Date	Indicates the time when an assignment ends
scores	int	Mark scored by the pupil for the assignment submitted
comment	Varchar(50)	Given by the teacher against the pupil's assignment
UserCode	int	Fk referencing the pupil's table

Teacher table

Attribute	Type	Description
TeacherID	Int	Unique,automatically identifies the teacher
AssignmentId	Int	Fk references the assignment table
UserCode	Int	Fk references the pupil's table

5. COMPONENT DESIGN

The pseudcode below shows how systematically the pupil given assignment and awarded scores and other actions that take place within KinderCare character draw system.

Start:

If (teacher is logged in) {

Then

Register pupil
Enter assignment details
Upload assignment

If (pupil is registered and is activated) {

And If (assignment is within the required time) {

Then

Attempt assignment

}

ElseIf(not activated)

Then

Request activation

}

If (teacher accepts request) {

Then

Pupil is activated

}

}

If (assignment is attempted and submitted) {

Then

Award score automatically

}

End

6. HUMAN INTERFACE DESIGN

6.1 Overview of User Interface

The system shall have both Graphical User Interface web interface and command line interface (CLI)

The teacher will use the web interface to;

- Log into the system
- Register pupils, their firstname, lastname, phone number and usercode
- Deactivate or activate a pupil
- Specify start time and end time of the assignment
- Attach a comment to the score
- View report which help in decision making

The pupil uses the command line interface to; -

- Log into the system
- Attempt assignment submitted by the teacher as long as they meet the given conditions and the assignment is within the required time
- Send request

6.2 Screen Images

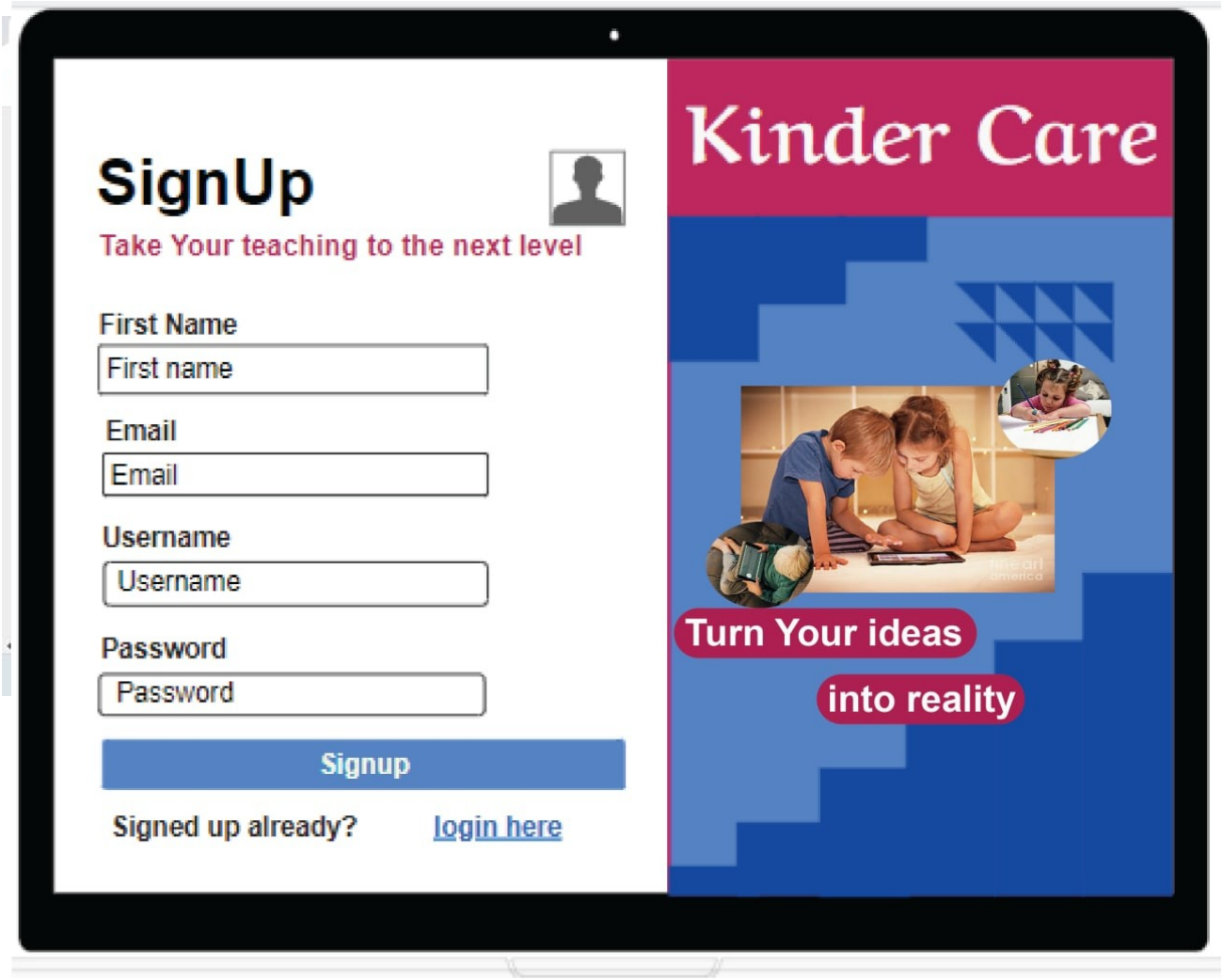

Figure 4: teacher signUp form

Figure 5: Teacher login form

Figure 6: Pupil Registration Form

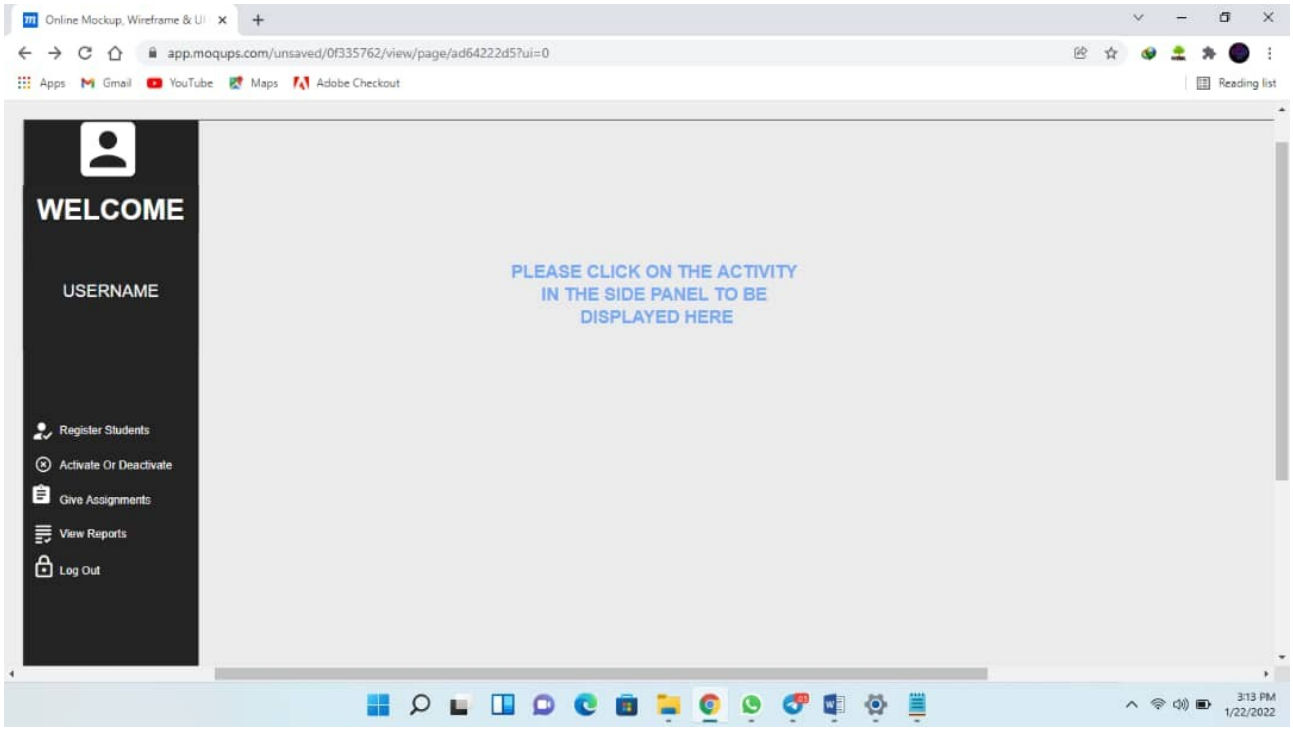

figure 7: Teacher's choice of activity to perform at a particular time.

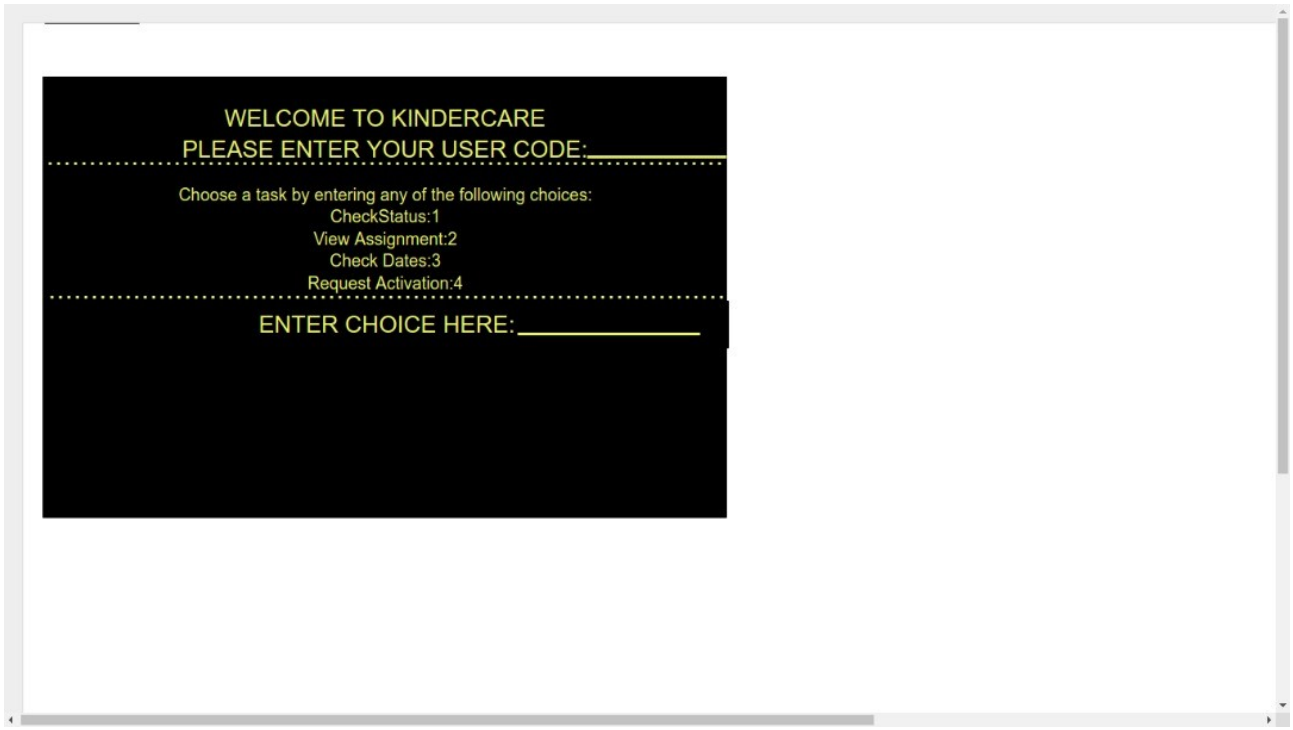

Figure 8: pupil's login to choose a task by checking the available choces

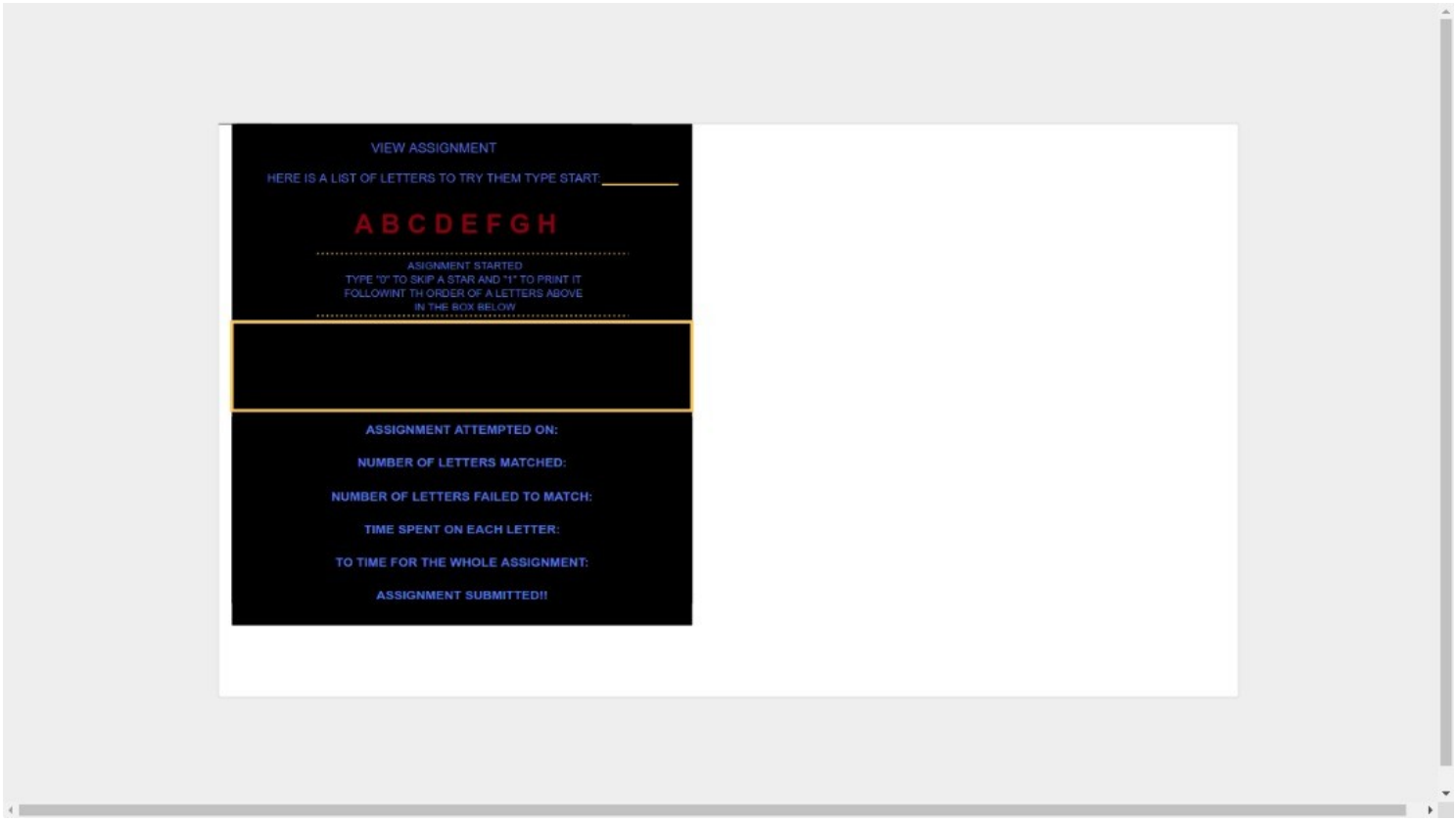

figure 9:pupil's screen to view assignment and to attempt that particular assignments plus records of previous attempts

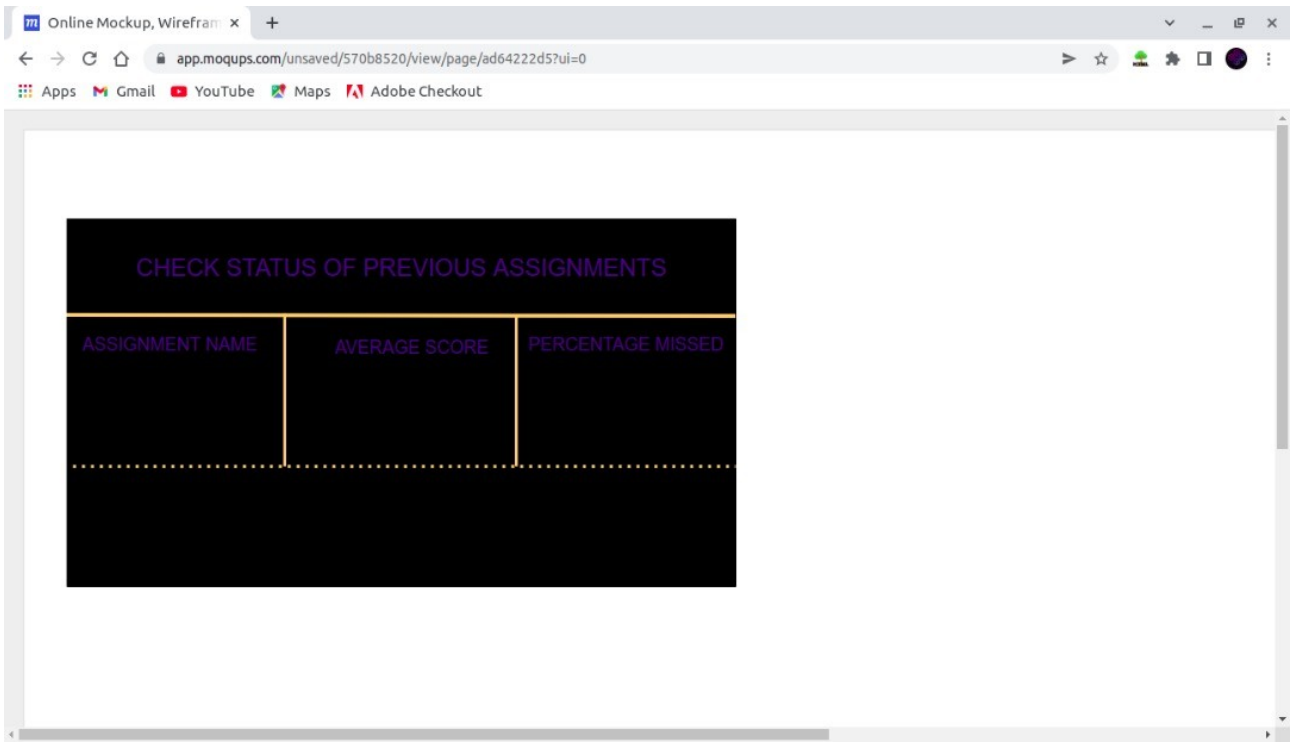

figure 10: children's screen to check status of the previous assignments

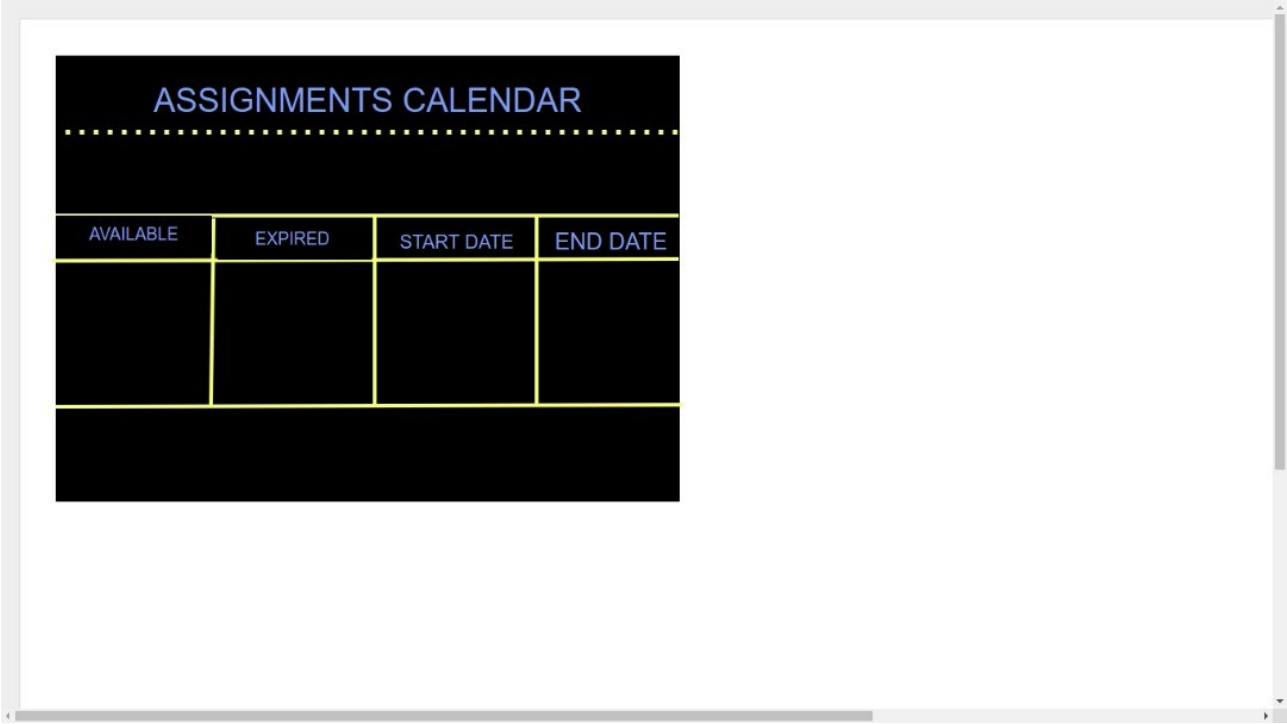

figure 11: calender showing assignments that are both available for attempting and those that are expired plus their start dates and enddates.