SOFTWARE DESIGN DOCUMENT

FOR

KINDERCARE ASSIGNMENT MANAGEMENT SYSTEM

GitHub link: https://github.com/rukundob451/GROUP-22-KinderCare-Assignment-System-Recess-Project

GROUP 22

Supervisor: Brian Muchake

|  |  |  |
| --- | --- | --- |
| Name | Registration No. | Student No. |
| CADEN DANIELLA | 20/U/0171 | 2000700171 |
| KISAKYE ESTHER | 20/U/0170 | 2000700170 |
| NANTALE LYTON AGNES | 20/U/7758/PS | 2000707758 |
| RUKUNDO BENJAMIN | 20/U/2083/EVE | 2000702083 |

TABLE OF CONTENTS

[1 INTRODUCTION 1](#_Toc93863665)

[1.1 Purpose 1](#_Toc93863666)

[1.2 Scope 1](#_Toc93863667)

[1.2.1 Goals and Objectives 1](#_Toc93863668)

[1.2.2 Benefits 1](#_Toc93863669)

[1.3 Document Overview 2](#_Toc93863670)

[1.4 Reference Material 2](#_Toc93863671)

[1.5 Definitions and Acronyms 2](#_Toc93863672)

[1.6 System Overview 3](#_Toc93863673)

[2 System Architecture 4](#_Toc93863674)

[2.1 Architectural design 4](#_Toc93863675)

[2.1.1 Description of the architectural diagram 5](#_Toc93863676)

[2.2 Decomposition Description 6](#_Toc93863677)

[2.2.1 Description of the context diagram 7](#_Toc93863678)

[2.2.2 Description of the context diagram 10](#_Toc93863679)

[2.2.3 Description of the Functional decomposition diagram 12](#_Toc93863680)

[2.3 Design Rationale 13](#_Toc93863681)

[3 Data Design 14](#_Toc93863682)

[3.1 Data Description 14](#_Toc93863683)

[3.2 Data Dictionary 15](#_Toc93863684)

[4 Component Design 18](#_Toc93863685)

[5 Human Interface Design 23](#_Toc93863686)

[5.1 Overview of the User Interface 23](#_Toc93863687)

[5.2 Screen images 25](#_Toc93863688)

[5.3 Screen Objects and Actions 32](#_Toc93863689)

LIST OF FIGURES

[Figure 2.1.1. Architectural design of the KC assignment system 9](#_Toc93862458)

[Figure 2.2.1 Context diagram for the KC Assignment Management System 11](#_Toc93862459)

[Figure 2.2.3. Level 0 Data Flow Diagram of KC Assignment management system 13](#_Toc93862460)

[Figure 2.2.4. Functional decomposition diagram for the KC Assignment Management System. 17](#_Toc93862461)

[Figure 3.1.1. Entity relationship diagram for the KinderCare Assignment Management System 19](#_Toc93862462)

[Figure 3.2.1. Register for an account pseudocode 23](file:///C:\Users\KISAKYE\Desktop\SOFTWARE_DESIGN_DOCUMENT_GROUP_22.docx#_Toc93862463)

[Figure 3.2.2. Register pupil pseudocode 23](file:///C:\Users\KISAKYE\Desktop\SOFTWARE_DESIGN_DOCUMENT_GROUP_22.docx#_Toc93862464)

[Figure 3.2.3. Activate pupil pseudocode 24](file:///C:\Users\KISAKYE\Desktop\SOFTWARE_DESIGN_DOCUMENT_GROUP_22.docx#_Toc93862465)

[Figure 3.2.4.Deactivate pupil pseudocode 24](file:///C:\Users\KISAKYE\Desktop\SOFTWARE_DESIGN_DOCUMENT_GROUP_22.docx#_Toc93862466)

[Figure 3.2.5.Upload assignment pseudocode 25](file:///C:\Users\KISAKYE\Desktop\SOFTWARE_DESIGN_DOCUMENT_GROUP_22.docx#_Toc93862467)

[Figure 3.2.6.Attempt assignment pseudocode 26](file:///C:\Users\KISAKYE\Desktop\SOFTWARE_DESIGN_DOCUMENT_GROUP_22.docx#_Toc93862468)

[Figure 3.2.7.Comment on pupil score pseudocode 27](file:///C:\Users\KISAKYE\Desktop\SOFTWARE_DESIGN_DOCUMENT_GROUP_22.docx#_Toc93862469)

[Figure 3.2.8. Grade assignment pseudocode 27](file:///C:\Users\KISAKYE\Desktop\SOFTWARE_DESIGN_DOCUMENT_GROUP_22.docx#_Toc93862470)

[Figure 3.2.9.Send Activation request pseudocode 27](file:///C:\Users\KISAKYE\Desktop\SOFTWARE_DESIGN_DOCUMENT_GROUP_22.docx#_Toc93862471)

[Figure 5.1.1. Navigation flow of the web system. 29](#_Toc93862472)

[Figure 5.2.1.Welcome screen 30](#_Toc93862473)

[Figure 5.2.2. Add assignment interface 32](#_Toc93862474)

[Figure 5.2.3.Activation Requests screen 33](#_Toc93862475)

[Figure 5.2.4. Register Pupils screen 34](#_Toc93862476)

[Figure 5.2.5.The command line interface 35](#_Toc93862477)

[Figure 5.3.1. Screen objects and actions 39](#_Toc93862478)

LIST OF TABLES

[Table 1.5.1.Definitions and Acronyms 7](#_Toc93862605)

[Table 2.2.1. Key for the context diagram of the KC Assignment Management System 11](#_Toc93862606)

[Table 2.2.2.Key for the level 0 data flow diagram of KC Assignment system 13](#_Toc93862607)

[Table 3.2.1. Teacher 19](#_Toc93862608)

[Table 3.2.2.Pupil 19](#_Toc93862609)

[Table 3.2.3.Submitted Assignment 20](#_Toc93862610)

[Table 3.2.4. Attempted Assignments 21](#_Toc93862611)

# INTRODUCTION

## Purpose

The purpose of this software design document is to describe the architecture and system design of the KinderCare Assignment System described in the assignment document. The KinderCare system shall be designed to provide a cost-effective way of handling assignments given to lower school children at KinderCare as well as help the children to remember and create different shapes of characters of the alphabet easily.

The primary audience of this document are the software developers and the project supervisor.

## Scope

The KinderCare Application shall consist of two parts; the command-line interface and the web interface.

The command-line interface shall be used by the pupils of KinderCare to attempt assignments submitted by the teachers and send activation requests to the teachers in case of deactivation.

The web interface shall be used by the school administrator to approve registration of teachers’ accounts, the teachers to register themselves and the pupils, submit assignments for the pupils, comment on pupils’ scores and also keep track of the pupils’ progress in the attempted assignments.

### Goals and Objectives

* To help pupils remember and create the different shapes of characters of the alphabet easily.
* To enable easy efficient tracking of pupils’ academic progress by the teachers.
* To access all pupils’ grades easily from a single data repository.
* To grade pupils’ assignments efficiently through automatic grading by the system.

### Benefits

* The system will reduce the time taken in grading of pupils’ assignments through automatic grading upon submission of completed assignments by the pupil.
* The system will improve decision making by the teachers through viewing reports generated by the system.
* The system will make giving assignments to pupils faster and easier since assignments will simply be created and submitted on the system for all pupils to access.

## Document Overview

This document has been organized into chapters as described below.

Chapter 1 that describes the scope of the KC assignment system project, purpose of this document, the reference material and acronyms used and their meanings.

Chapter 2 that describes the system overview, i.e., the general description of the functionality, context and design of the KinderCare Assignment Management System.

Chapter 3 that describes the architecture of the system, which consists of the system architectural design, the decomposition of the system and rationale for the choice of design.

Chapter 4 that describes the data design of the KinderCare system, which shows how the information domain of the system is transformed into data structures. It also includes the data dictionary of the system.

Chapter 5 that this chapter focuses on the component design of the system and gives a detailed functional description of each component of the system.

Chapter 6 that describes the human interface design of the system. It describes the functionality of the system from the user’s perspective accompanied with screen images to enable the audiences of the document visually understand each component of the system.

## Reference Material

## Definitions and Acronyms

Table 1.5.1.Definitions and Acronyms

|  |  |
| --- | --- |
| Acronym | Definition |
| KC | KinderCare |
| CLI | Command Line Interface |

## 

## System Overview

The KinderCare system shall be designed to provide a cost-effective way of handling assignments given to lower school children at KinderCare as well as help the children to easily remember and create different shapes of characters of the alphabet.

The KC system comprises two parts namely; the command-line interface and the web interface. These two parts communicate with each other through a database. The teachers and school administrator use the web interface while the pupils use the command-line interface. A teacher is first required to register for an account through which they will access the system. The school administrator then approves the account. After a teacher’s account has been approved, they can now access the system. When a teacher successfully logs into the system, they can register details of pupils that should be granted access to the system. The pupil’s details are stored in the database and the pupil is automatically activated. The teacher is also able to submit assignments specifying the start and end time of the assignments.

Only activated pupils are able to view and attempt the assignments, provided they are still within the attempting time frame specified. Each assignment is made up of at most eight (8) characters of the alphabet to attempted. Upon opening an assignment, the system informs a pupil of how much time is left to close the assignment and also informs the pupil of how many characters are in the assignment. The system presents the characters in an assignment one by one, for the pupil to attempt and submit. Time of attempt for each character is recorded and after the attempt of the last character, the total amount of time taken is provided by the system 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.

The system automatically grades the assignments and the teacher is able to view the scores and add comments when they log in. A pupil is also able to see the assignment score and comment on their next log in.

The teacher can also deactivate pupils and these pupils cannot attempt the uploaded assignments even if they are registered. The pupils can seek reactivation by sending an activation request to the teacher through the command-line interface.

# System Architecture

## Architectural design

The KC assignment system uses a 3-Tier client/server model, which comprises three client machines, i.e. the school administrators, the teacher’s and the pupil’s machines. These machines communicate through a database.

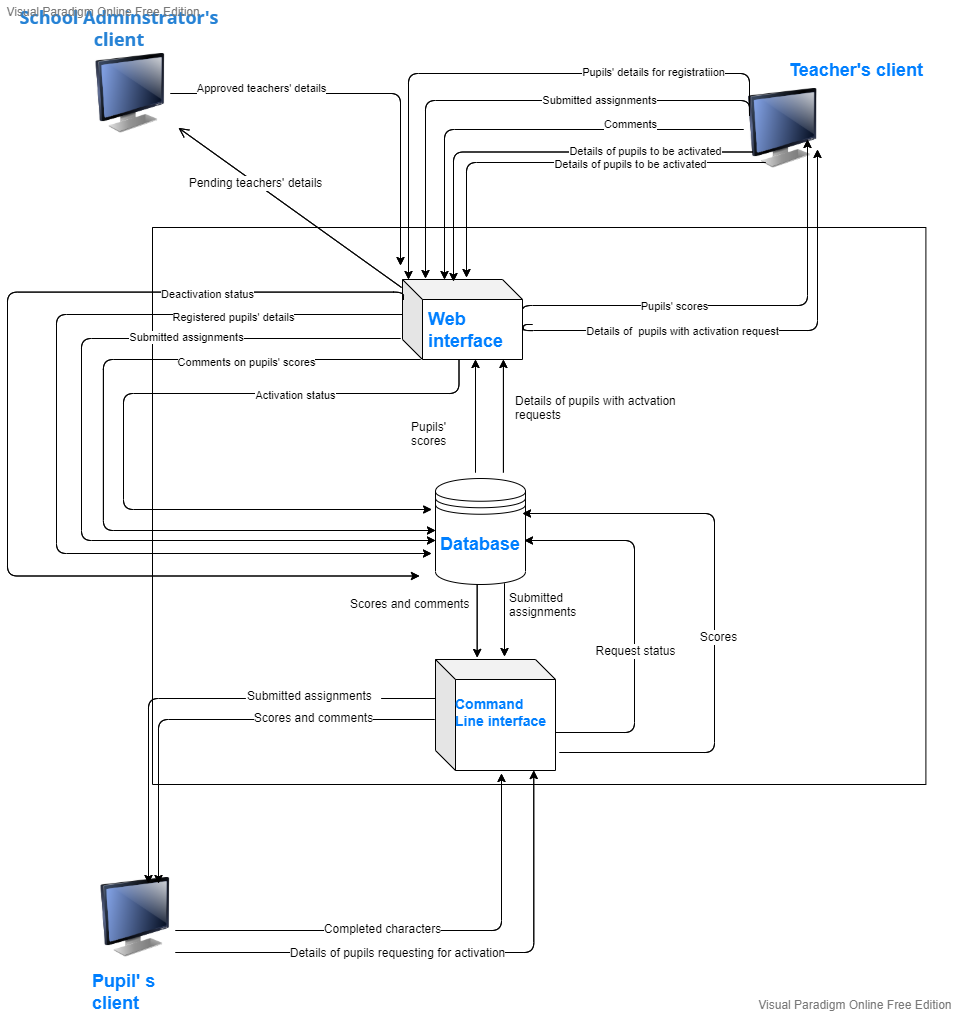


Figure 2.1.1. Architectural design of the KC assignment system

Key for the Architectural diagram

Represents a data flow

### Description of the architectural diagram

#### The web interface

The web interface will provide a platform for teachers to register for accounts as well as register the pupils. A school administrator will also be able to approve teachers’ accounts for registration through the web interface. Once approved, the teachers will be able to access the system. The teacher will also be able to activate and deactivate pupils. The teachers will be able to submit assignments, view the pupils’ scores and add comments. The teachers will receive details of pupils requesting for activation.

#### The command line interface

This interface will enable the pupils to view and attempt submit assignments submitted by the teacher as long they have not yet expired and are open for attempting. Pupils will also use the interface to view their scores and comments from the teachers on the next log in.

In addition, the pupils will be able view assignments that within a given date range. Deactivated students will be able to send activation requests to the teachers. All these activities will be made possible by simply running commands on the command line interface.

#### The database

The database will store the pupils’ and teachers’ details after registration.

The database will store the assignments submitted by the teachers. The pupils will access the assignments from the database. The scores attained on the attempted assignments will be stored in the database. The teacher will access the scores of the pupils from the database and add comments which will be stored in the database. The pupils will access the scores and the comments from the database on their next log in.

The request status of deactivated pupils who would like activated is stored in the database. The activation status of pupils who have been activated or deactivated by the teachers is changed in the database. The teachers will access the pupils’ details with activation requests from the database.

## Decomposition Description

The KC app will follow a structured design approach. Both the command-line interface and the web interface shall follow principles of structured development methodology. The command-line interface will be developed using the C programming language which is a structured programming language. We shall use Laravel framework.

**Context diagram for the KC Assignment Management System**

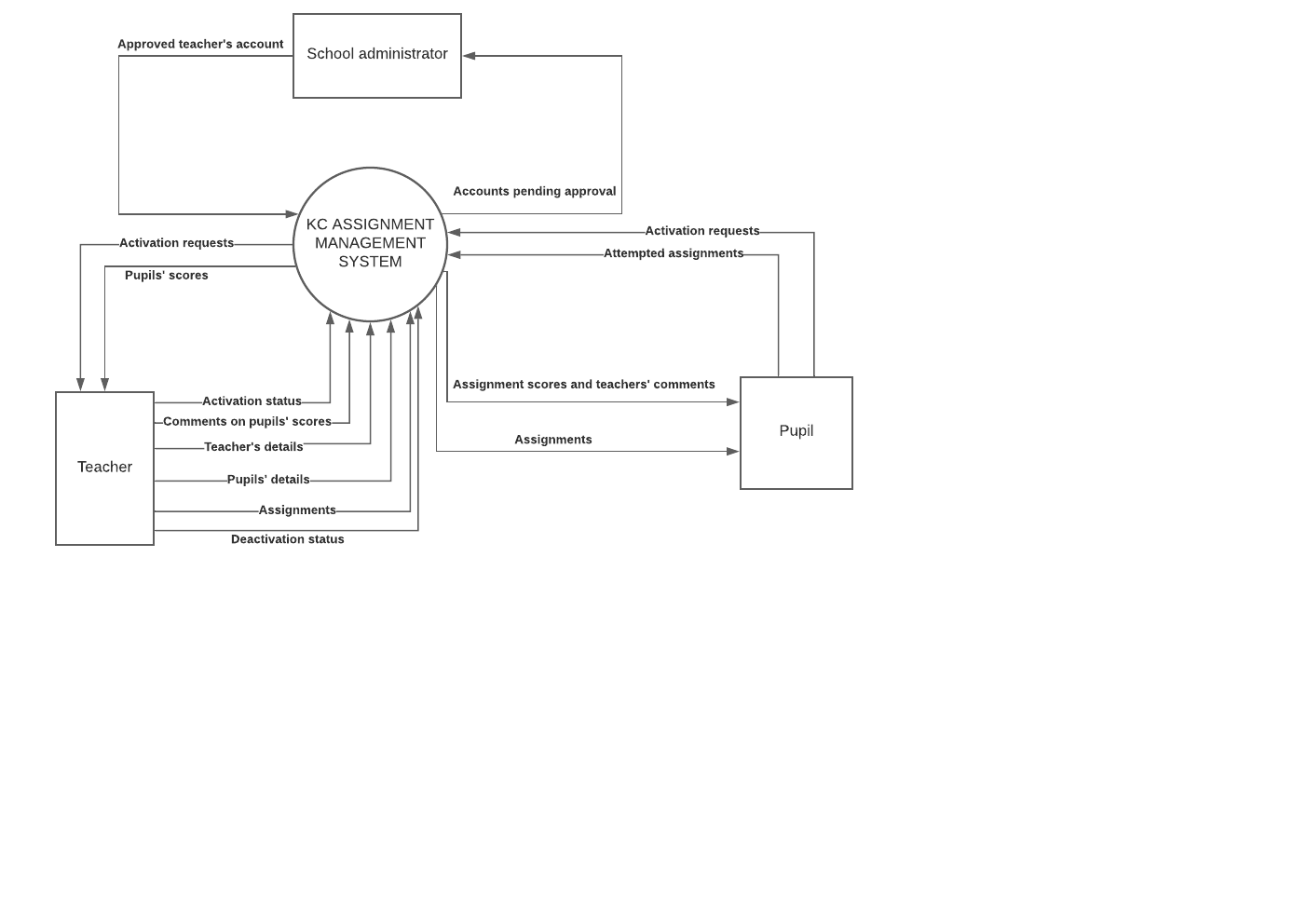


Figure 2.2.1 Context diagram for the KC Assignment Management System

Table 2.2.1. Key for the context diagram of the KC Assignment Management System

|  |  |
| --- | --- |
| Symbol | Meaning |
|  | Represents an external entity |
|  | Represents a data flow |
|  | Represents a process |

### Description of the context diagram

A teacher registers for an account by entering their details in the system via the web interface. A school administrator has to approve the teacher’s details to ensure that they are indeed a teacher at KC.

An approved teacher is able to register the pupils into the system. The teacher also submits assignments for the pupils, comments on pupils’ scores and also activates and deactivates pupils.

The pupil uses the command-line interface to attempt uploaded assignments, request for activation if deactivated and also view scores of the attempted assignments and the attached comments from the teacher

**Level 0 Data Flow Diagram of KC Assignment management system**

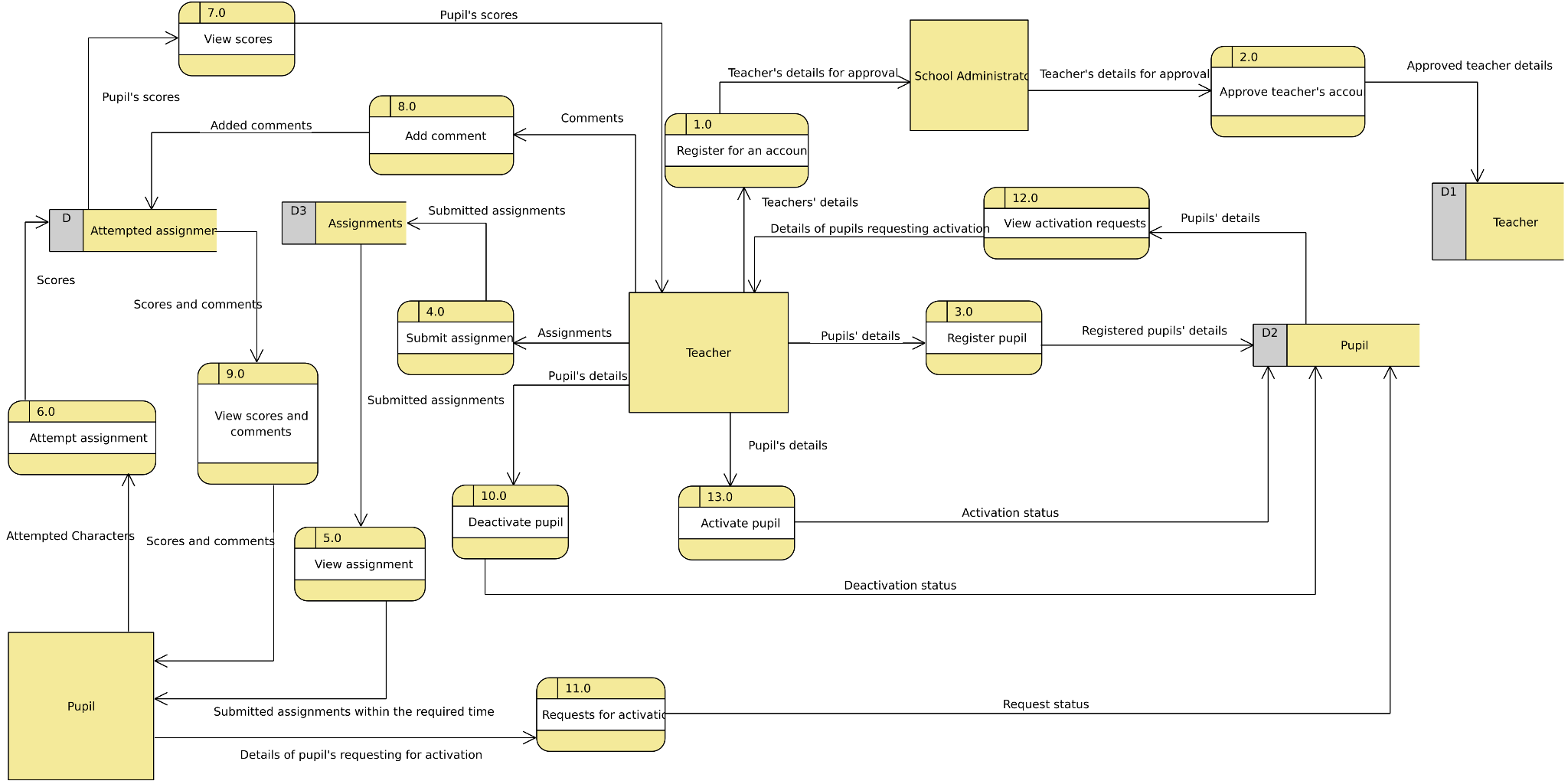
****

Figure 2.2.3. Level 0 Data Flow Diagram of KC Assignment management system

Table 2.2.2.Key for the level 0 data flow diagram of KC Assignment system

|  |  |
| --- | --- |
| Symbol | Meaning |
|  | Data flow |
|  | External entity |
|  | Process |
|  | Data Store |

### Description of the context diagram

**Process 1: Register for an account**

A teacher enters their credentials to register for an account with the system and they await approval of the account from the school administrator. The details of the teacher are stored in the teacher data store and will have a status called pending.

**Process 2: Approve teacher’s account**

The school administrator approves a teacher’s account. In the teacher data store, the status of an approved teacher is changed from pending to approved.

**Process 3: Register pupil**

A teacher registers pupils in this process by entering their details in the system and these details are also stored in a pupil data store their activation status in the pupil data store is automatically set to activated.

**Process 4: Submit assignment**

The teacher submits assignments specifying the start time and end time which are stored in the assignment data store. These assignments are accessed by the pupils from the assignment data store.

**Process 5: View assignment**

A pupil sees all the assignments submitted by the teacher as long as they are still available for attempting.

**Process 6: Attempt assignment**

An activated pupil submits the attempted characters and the system automatically grades the pupils. The scores are then stored in the attempted assignment data store

**Process 7: View scores**

The teacher sees the scores of each pupil for each assignment that has been attempted through accessing the attempted assignment data store.

**Process 8: Add comments**

The teacher adds comments on the scores of the pupil in a given assignment and these comments are stored in the attempted assignment data store.

**Process 9: View scores and comments**

A pupil sees their assignment scores and the comments added by the teacher on their next log in.

**Process 10: Deactivate pupil**

The teacher submits the pupil’s details who have to be deactivated. The activation status in the pupil data store changed to deactivated for these particular pupils.

**Process 11: Request for activation.**

The details of the pupils requesting for activation are submitted. The request status is changed in the pupils’ data store which shows they have requested for activation.

**Process 12: View activation requests**

The teacher sees the details of all pupils requesting for activation.

**Process 13: Activate pupil**

The teacher submits the details of the pupils requesting for activation. The activation status for these particular pupils is changed to activated in the pupils’ data store.

**Functional decomposition diagram for the KC Assignment Management System**

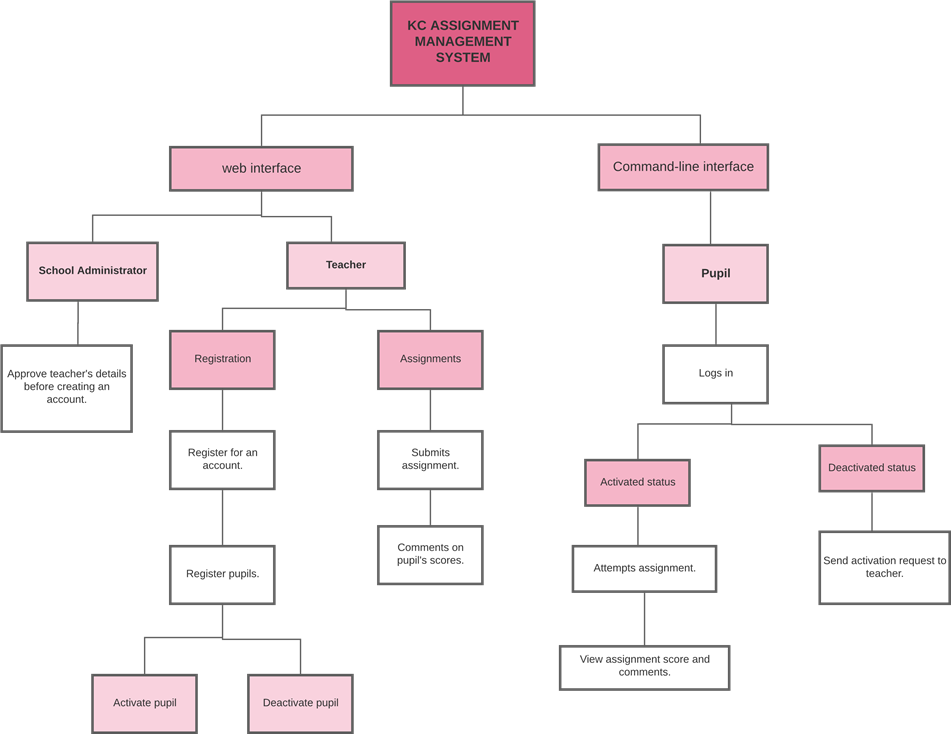
****

Figure 2.2.4. Functional decomposition diagram for the KC Assignment Management System.

### Description of the Functional decomposition diagram

The functional decomposition diagram above represents the breakdown of the processes within the KC assignment system.

The web interface is used by the teacher to register for an account and the school administrator to approve the teacher’s account before it is created. Upon approval, the teacher’s account is created and he is able to register accounts for the pupils. The teacher also submits assignments for the pupils and comments on the pupils’ scores in the assignments via the web interface. The teacher can also activate and deactivate pupils via the web interface.

On the other hand, the pupil logs into the system via the command line interface and only an activated pupil can attempt assignments and view their scores and comments. A deactivated pupil cannot access the assignments and has to seek re-activation by sending an activation request to the teacher.

## Design Rationale

The architecture in 3.1 was selected to enable each client machine to process their functions separately and then allow seamless communication between the two machines through the database.

The teacher can execute their functions such as registration and submission of assignments on the web interface with no interruption from the pupil’s functions like logging into the system and attempting assignments on the command-line interface.

The database in the architecture acts as a central data repository which allows the client machines to store, access and share data within the system.

# Data Design

## Data Description

The KC assignment management system uses data about pupils and teachers. Information obtained from the teachers to be registered is recorded into the system using a web application while the information about the pupils such as attempting an assignment is recorded into the system using the command line interface.

**Entity relationship diagram for the KinderCare Assignment Management System**

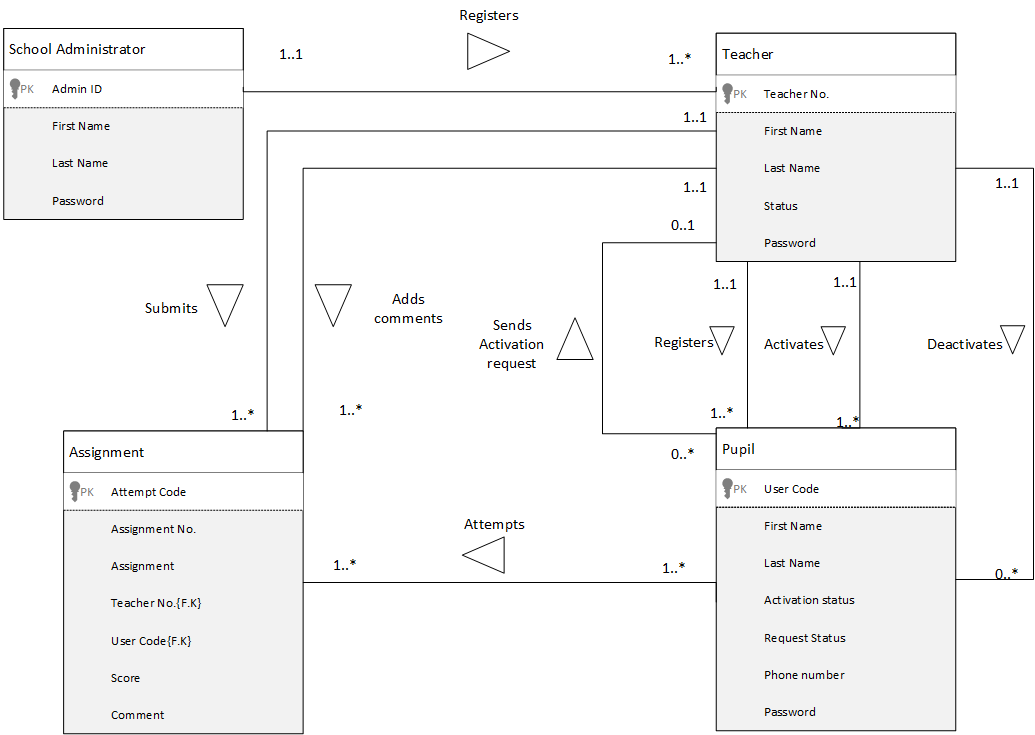
****

Figure 3.1.1. Entity relationship diagram for the KinderCare Assignment Management System

## Data Dictionary

Table 3.2.1. Teacher

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Constraint** | **Description** |
| First Name | varchar | Not null | The first name of the teacher. |
| Last name | varchar | Not null | The last name of the teacher |
| Teacher Number | varchar | Primary key | This uniquely identifies each teacher in the database. |
| Password | varchar | Not null | This is the password of the teacher’s account created. |
| Status | varchar | Not null | This specifies whether a teacher’s account is pending or has been verified. |

Table 3.2.2.Pupil

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Constraint** | **Description** |
| First Name | varchar | Not null | This is the first name of the pupil. |
| Last name | varchar | Not null | This is the last name of the pupil |
| Phone number | varchar | Not null | This is the pupil’s phone number |
| User Code | varchar | Primary Key | This uniquely identifies each pupil and it is what the pupils use to log into the system. |
| Activation Status | varchar | Not null | This is identifying a pupil as either activated or deactivated |
| Request Status | varchar | Null | This informs a teacher of an activation request that has been sent. |

Table 3.2.3.Submitted Assignment

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Constraint** | **Description** |
| Assignment Number | varchar | Primary Key | This uniquely identifies every assignment. |
| Assignment | JSON | Not null  Check length < 9 | This is the assignment to be attempted. |
| Teacher Number | varchar | Foreign Key | This references the specific teacher who has submitted the assignment. |
| Number of characters | integer | Not Null | This specifies the number of characters in the assignment. |
| Start time | Time & Date | Not null | This specifies the time the assignment should open for attempting |
| End time | Date & time | Not null | This specifies the closing time of the assignment. |

Table 3.2.4. Attempted Assignments

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Constraints | Description |
| Assignment Number | varchar | Foreign Key | This references the specific assignment. |
| User Code | varchar | Foreign Key | This references the pupil who has done the assignment. |
| Attempt Code | varchar | Primary Key | This uniquely identifies each assignment that has been attempted by a specific pupil. |
| Score | integer | Not Null | This is the mark scored on the assignment by a pupil. |
| Teacher Number | varchar | Foreign Key | This references the specific teacher who has submitted the assignment. |
| Comment | varchar | Null | This is the comment given to the pupil based on the score they attained. |
| duration | time | Null | This specifies the amount of time taken to attempt the assignment. |

# Component Design

**Register for an account pseudocode**

Figure 3.2.1. Register for an account pseudocode

START

Enter first name, last name, teacher number, password

status = pending

Add first name, last name, teacher number, password and status to database

STOP

**Register pupil pseudocode**

START

Enter user code, first Name, Last Name, phone Number

Check if entered data already exists in database

If data doesn’t exist already

Add user code, first Name, Last Name, phone Number to database

If successfully added

Print “Registration Successful”

Else

Print “Failed to register pupil, try again”

Else

Print “Pupil already registered”

STOP

Figure 3.2.2. Register pupil pseudocode

**Activate pupil pseudocode**

Figure 3.2.3. Activate pupil pseudocode

START

Read user code, Activation Status

if Activation Status = deactivated

Set Activation Status = activated

print “Pupil successfully activated”

STOP

**Deactivate pupil pseudocode**

Figure 3.2.4.Deactivate pupil pseudocode

START

Read user code, Activation Status

If pupil is active

Set Activation Status = deactivated

print “Pupil successfully deactivated”

STOP

**Upload assignment pseudocode**

Figure 3.2.5.Upload assignment pseudocode

START

Number of characters = 0

Submit = false

Enter Assignment Number auto generated by the system

Enter start time, end time

do {

Enter character

Add character to assignment

Number of characters = Number of characters +1

If submit = True

Break

} while(N<=8)

Add Assignment Number, start time, end time, characters, Number of characters to Assignment table in database

If added successfully

Print “Assignment successfully added.”

Else

Print “Failed to add assignment, please try again.”

**Attempt assignment pseudocode**

Figure 3.2.6.Attempt assignment pseudocode

START

If pupil is registered

If pupil is deactivated

Print “Cannot attempt assignment because you have been deactivated”

Else

Print time left to close assignment and number of characters in assignment

Duration= 0

While (number of attempted characters < number of characters in assignment) {

Display character from the assignment

Capture pupil attempt

Record time taken to attempt character

Duration = Duration + time to attempt character

}

Print Duration

Print “Assignment submitted successfully”

Print all attempted characters on screen.

STOP

**Comment on pupil score pseudocode**

START

Enter comment

Save comment to database

STOP

Figure 3.2.7.Comment on pupil score pseudocode

**Grade assignment pseudocode**

START

Figure 3.2.8. Grade assignment pseudocode

Determine number of characters in assignment N

Determine correctly attempted characters P

Final grade = P/ N \* 100

Save final grade to database

STOP

**Send activation request pseudocode**

Figure 3.2.9.Send Activation request pseudocode

START

Read user code, pupil name, activation status

If activation status = deactivated

Print “Activation request sent successfully.”

Else

Print “Already active. Request not sent.”

STOP

# Human Interface Design

## Overview of the User Interface

The system will be used by three users. These are.

Pupils (Use CLI)

Teachers (Use web interface)

School administrator (Use web interface)

The school administrator will use the web application to activate accounts for teachers that have created an account because they are using the system for the first time.

The pupils will use the Command line interface to attempt assignments.

The pupil will log in with their user code and password.

The pupil will enter the following commands to complete the different tasks needed

Viewall. This command will display assignment number and date and show if the pupil attempted the assignment or not.

Checkstatus. This command will display the status report of the pupil summarizing all assignments. (How many attempted, average score, percentage missed, percentage attempted among others). The status report will be displayed in a table format.

Viewassignment assignmentid. This will be used to see details of a specified assignment. for example, Viewassignment AS004 will display a table showing the details (Number of characters, Closing date, time left till the assignment AS004 closes and whether the student attempted or not)

Checkdates datefrom dateto. This will be used by a pupil to display all assignments between a specified date range. The assignments will be displayed in a table format showing all their details.

In case there are no assignments within the specified date range, the system will display a message “There are no assignments within the specified date range”

RequestActivation. This will be used by a deactivated pupil to request the teacher to activate him or her.

When the request is successfully submitted the system will display the message “Request successfully sent” otherwise “Failed to send request.”

When attempting a given character in the assignment, the pupil will enter a 1 where they want a star to be printed and a 0 where a star should not be printed.

**Navigation flow of the web system**

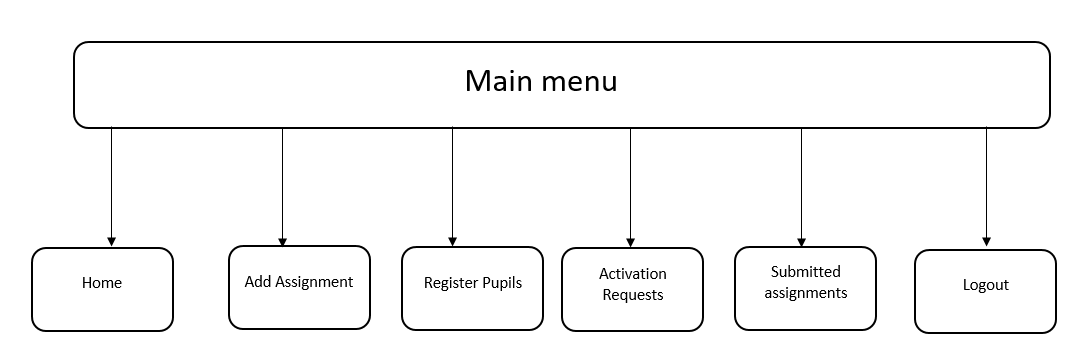


Figure 5.1.1. Navigation flow of the web system.

The teachers will use the main menu to navigate the web system.

The Home link in the main menu will display a table with registered students and a table summarizing the assignments.

The Add Assignment link will display a grid for the teacher to select characters (by checking the boxes beside them) to the assignment and specify the start and end time. The assignment code will be automatically generated.

The Register Pupils link will display a form for the teacher to enter the details of the pupil to be registered.

The Activation Requests link will display a table with the details of the pupils with activation requests.

The Submitted Assignments link will display a table with the details of assignments that have been attempted by pupils, showing their grades and provide an input field for adding a comment.

## Screen images

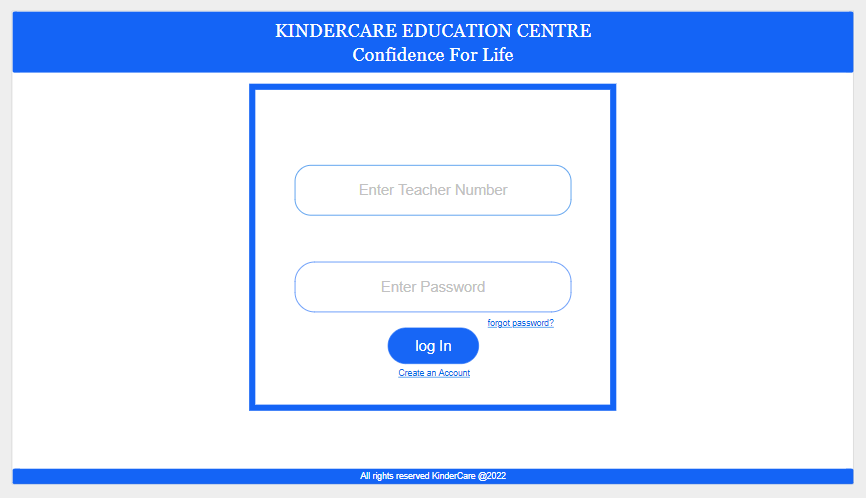


Figure 5.2.1.Welcome screen

The welcome screen has two input boxes; “Enter Teacher Number” and “Enter Password”. The teacher will be required to enter their teacher number and password in the respective input boxes.

The “log In” button submits the login details after the teacher has entered them. The details are then verified to see if they match any entries in the database.

If yes, log in is successful and the teacher will be able to view the home screen.

In case the log in details don’t match any in the database, the teacher will get a message “Wrong Teacher number or password, try again!”.

The screen also has a “forgot password” link. On clicking this link, the teacher will be able to reset their password.

The “Create Account” link will enable first time users to create accounts.

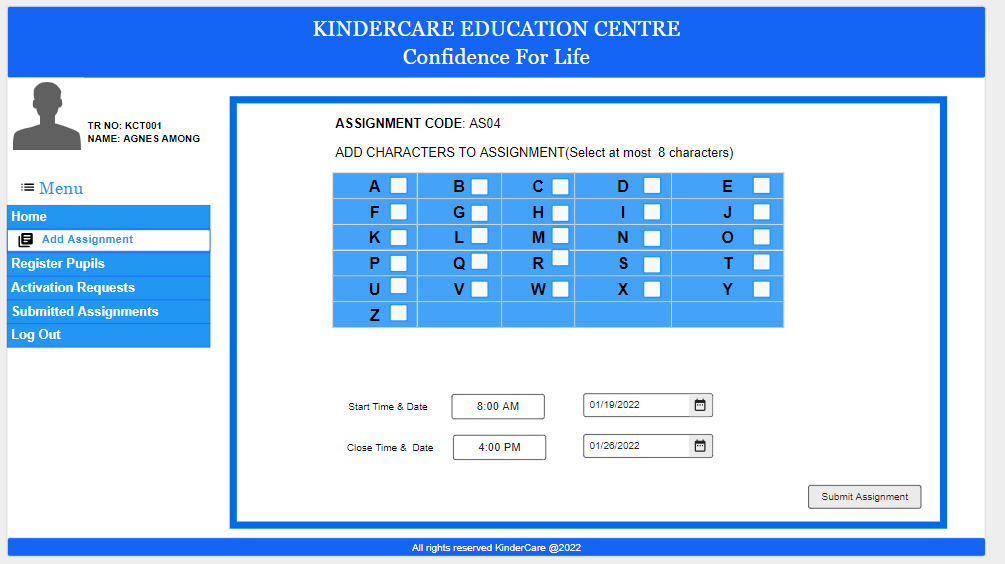


Figure 5.2.2. Add assignment interface

On clicking the add assignment link, the system will auto generate the assignment code.

The teacher will then be required to select at most 8 characters by checking the checkboxes besides the characters.

The teacher will specify the “Start Time & Date'' by entering the time in the input field beside the label and selecting a date from the date picker beside the time input field.

The teacher will specify the “Close Time & Date” by entering the time in the input field beside the label and selecting a date from the date picker beside the time input field.

The teacher will then click the “Submit Assignment” button to add the assignment for pupils to attempt.

On successful addition of the assignment, the system will display a message saying “Assignment successfully added”. In case the system fails to add the assignment, the message displayed will read “Sorry, the assignment has not been added, try again!”

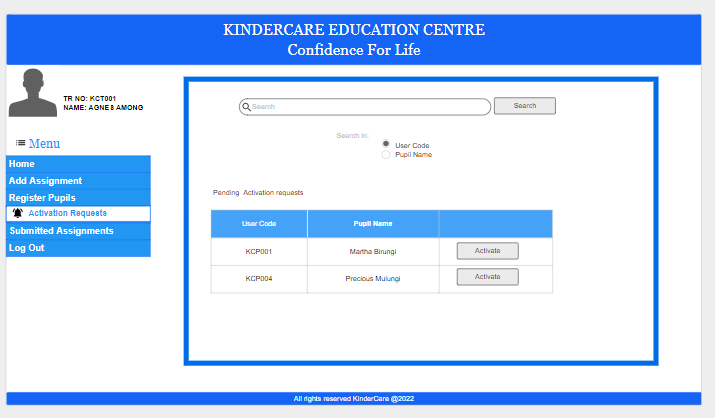


Figure 5.2.3.Activation Requests screen

On clicking the “Activation Requests” link, the teacher will be presented with a table showing all the pending activation requests.

The table will display the “User Code” and “Pupil Name” for the pupil that sent the request.

Beside each Pupil Name will be an “Activate” button to enable the teacher to activate the pupil.

On successful activation, the Pupil details will be removed from this table and the “Status” of the pupil in the “Registered Pupils” table on the home screen will change to “Activated”.

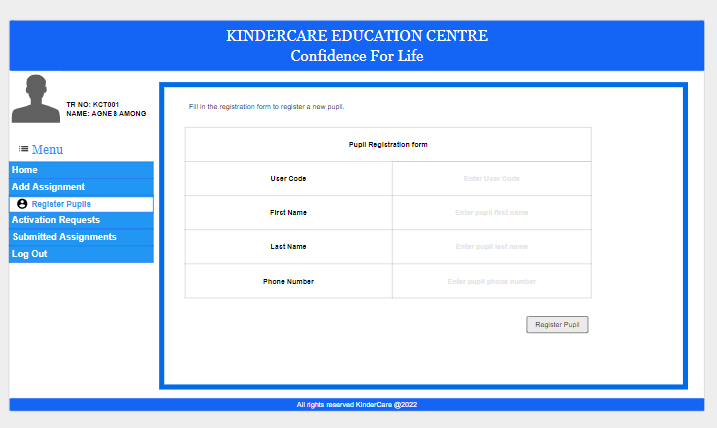


Figure 5.2.4. Register Pupils screen

On clicking the “Register Pupils” link, the teacher will be presented with a “Pupil Registration form”.

The teacher will be required to enter “User Code”, “First Name”, “Last Name” and “Phone Number” for the pupil.

The teacher will then click the “Register Pupil” button to complete the registration process. The details of the pupil will then be added to the database.

Registration will only be successful if all the form fields have been filled.

On successful pupil registration, the system will display the message “Registration Successful” otherwise, “Failed to register pupil, try again”.

The newly registered pupil details will then appear in the “Registered Pupils” table on the “Home” screen.

The search pane will enable the teacher to search for a specific pupil using their user code or name.

The Command line interface

Prompt>Enter your user code

>>> KP001

Prompt>Enter your password

>>>\*\*\*\*\*

Prompt:>logged in successfully

Figure 5.2.5.The command line interface

The pupil will be prompted to enter their user code. The pupil will enter the user code and press the enter button.

The pupil will then be prompted to enter their password. On entering the password, the pupil will press the enter button. To submit the details for verification by the system.

Once the details are verified and found accurate, the system will display the message “logged in successfully”

Otherwise, the system will display the message “Wrong user code or password.”

The pupil will enter the different commands into the command line interface to perform the different tasks. The system will display results in tables.

The pupil will enter the following commands to complete the different tasks needed

Viewall. This command will display assignment number and date and show if the pupil attempted the assignment or not.

Checkstatus. This command will display the status report of the pupil summarizing all assignments. (How many attempted, average score, percentage missed, percentage attempted among others). The status report will be displayed in a table format.

Viewassignment assignmentid. This will be used to see details of a specified assignment. for example, Viewassignment AS004 will display a table showing the details (Number of characters, Closing date, time left till the assignment AS004 closes and whether the student attempted or not)

Checkdates datefrom dateto. This will be used by a pupil to display all assignments between a specified date range. The assignments will be displayed in a table format showing all their details.

In case there are no assignments within the specified date range, the system will display a message “There are no assignments within the specified date range”

RequestActivation. This will be used by a deactivated pupil to request the teacher to activate him or her.

When the request is successfully submitted the system will display the message “Request successfully sent” otherwise “Failed to send request.”

When attempting a given character in the assignment, the pupil will enter a 1 where they want a star to be printed and a 0 where a star should not be printed.

## Screen Objects and Actions

|  |  |  |
| --- | --- | --- |
| Object | Name | Action |
|  | Search pane | Will enable the teacher to search for specific information from the different tables and columns |
|  | search button | to search for the required information |
|  | Radio button | Specifies search table and search field |
|  | Side menu | This is used to display the home screen of the system and its links to the different pages |
|  | Date input field | This will specify the start time and date by entering the time in the input field beside the label and selecting a date from the date picker beside the time input field. |
|  | Log in button | Submits the login details after the teacher has entered them |
|  | Checkbox | Used by the teacher to select characters to add to an assignment. |
|  | Submit assignment button | Adds the assignment for pupils to attempt. |
|  | Activate button | button to enable the teacher to activate the pupil. |
|  | Toggle switch | This is used to enable the teacher to activate and deactivate pupils |
|  | Comment input field | This adds a comment beside a pupils grade. |
|  | Register pupil button | This is used to complete the registration process only if all the form fields have been filled. |
| Prompt> |  | Prompts the user to enter input. |
| >>> |  | Takes in user input from the command line. |
| Prompt:> |  | Displays a message from the system |

Figure 5.3.1. Screen objects and actions