**DESIGN DOCUMENT**

**FOR**

**KINDERCARE ASSESSMENT SYSTEM.**

**Git hub link:** [**https://github.com/mwine-09/KINDERCARE.git**](https://github.com/mwine-09/KINDERCARE.git)

**G-27**

**Supervisor: Madam Mary Nsabagwa.**

**TEAM MEMBERS:**

|  |  |  |
| --- | --- | --- |
| **NAME** | **REGNO** | **STUDENTNO** |
| AMUMPAIRE TUMURAMYE | 20/U/7782/PS | 2000707782 |
| RUJUMBA LEONARD | 20/U/7781/PS | 2000707781 |
| SSUKKA JOLLY JOE | 20/U/2041/EVE | 2000702041 |
| MUSIIME BENJAMIN | 20/U/20123/PS | 2000720123 |

**LAB SECTION:**

**WORKSTATION:**

**DATE:** 01/21/2022

Table of Contents

[1 INTRODUCTION 5](#_Toc93854294)

[1.1 Purpose 5](#_Toc93854295)

[1.2 Scope 5](#_Toc93854296)

[1.3 Document Overview 6](#_Toc93854297)

[1.4 References 6](#_Toc93854298)

[1.5 Definitions and acronyms 6](#_Toc93854299)

[2 SYSTEM OVERVIEW 7](#_Toc93854300)

[3 SYSTEM ARCHITECTURE 9](#_Toc93854301)

[3.1 Architectural design 9](#_Toc93854302)

[3.2 Decomposition Description 10](#_Toc93854303)

[4 Data design 16](#_Toc93854304)

[4.1 Data description 16](#_Toc93854305)

[4.2 Data dictionary 18](#_Toc93854306)

[5 Component design 22](#_Toc93854307)

[6 Human interface design 25](#_Toc93854308)

[6.1 Overview of the human interface design 25](#_Toc93854309)

[6.2 Screen images 28](#_Toc93854310)

[6.3 Screen objects and actions 31](#_Toc93854311)

**List of figures.**

[figure 2. 1 a context level diagram of KINDERCARE system 8](#_Toc93854312)

[figure 3.1.1 shows the architectural design of our system. 9](#_Toc93854315)

[figure 3.2. 1 This shows the two entities interact with the system. 10](#_Toc93854318)

[figure 3.2. 2 A level 0 diagram showing exploded KINDERCARE assessment system 11](#_Toc93854319)

[figure 3.2. 3 decomposition diagram of process 7 13](#_Toc93854320)

[figure 3.2. 4 decomposition diagram of process 3 14](#_Toc93854321)

[figure 3.2. 5 a functional decomposition diagram of the system 15](#_Toc93854322)

[figure 4.2. 1 an Entity relationship diagram showing the structure of the database…………21](#_Toc93842161)

[figure 5.1 shows the pseudocode for registering a pupil 22](#_Toc93854323)

[figure 5.2 shows the pseudocode for uploading an assignment by a pupil 23](#_Toc93854324)

[figure 5.3 shows the pseudocode for marking an assignment submitted by a pupil 23](#_Toc93854325)

[figure 5.4 shows the pseudocode for adding a comment by a teacher 24](#_Toc93854326)

[figure 5.5 shows the pseudocode for presenting the different characters when attempting the assignment 24](#_Toc93854327)

**List of tables.**

[table 4.2.1 describes data about the pupil 18](#_Toc93854354)

[table 4.2.2 shows data about a teacher. 19](#_Toc93854355)

[table 4.2.3 holds data about an assignment 20](#_Toc93854356)

[table 4.2.4 holds data about the marks scored by the pupil in an assignment 20](#_Toc93854357)

[table 4.2.5 holds data about pupil activation. 20](#_Toc93854358)

[table 6.1.1 shows the command available to the pupil and their actions…………………….27](#_Toc93854359)

[table 6.3. 1 shows screen objects and actions………………………………………………...31](#_Toc93854360)

# INTRODUCTION

## Purpose

This software design document describes the architecture and the system design of the KINDERCARE assessment system to be developed.

It is going to be used by the development team as guide on how the system should look like and function as specified by KINDERCARE. The document is also going to be used by the project supervisor, Dr. Mary Nsabagwa and other developers that will be required to upgrade and maintain the system.

## Scope

The software is meant to enable the teacher to set an assignment for the pupils in KINDERCARE who in turn are expected to attempt and submit the attempted assignments back to the teacher.

* **GOALS & OBJECTIVES**

1. To enable pupils to remember and create different shapes of characters.
2. To allow the teacher to set and send assignments to the pupils easily.
3. To allow the pupils to attempt the assignments sent by the teacher.
4. The system should be able to mark and award mark to the submitted assignments by the pupils

* **BENEFITS**

1. The system will help the pupil to remember through attempting various assignments sent by the teacher.
2. It will also increase efficiency by marking and awarding marks automatically to the attempted assignments upon submission by the pupils.
3. The system allows the teacher to set the start and end time for the assignment which allows the pupils to manage their time while attempting those assignments.
4. The teacher is able to communicate to a specific pupil about the marks of a particular assignment through comments which are viewed by the pupil on the next login.
5. A teacher is able is to view necessary reports that help them in decision making.

## Document Overview

This document is written according to the standards of the software document explained in the “IEEE recommended practice for software design descriptions”.

Section 1 contains the scope, purpose and overview of the document.

Section 2 has the system overview describing what the system does generally.

Section 3-5 contains discussions of the designs and projects with the diagrams.

Section 6 shows samples of user interface from the system and section 7 contains a requirement matrix.

The appendix contains a list of all tools and environment variables for the entire project.

## References

* “IEEE std 1096-1998” for the full IEEE recommended practice for software design descriptions.
* Systems analysis and design ,11th Edition by Scott Tilley.

## Definitions and acronyms

* **DFD** – data flow diagram(s)

A **Data Flow Diagram** (DFD) is a graphical representation of the “flow” of data through an information system

* **SDD** – system design document

A **system design document** is a detailed description of the system requirements, operating environment, architecture, files and database design

* **CLI -**command line interface

# SYSTEM OVERVIEW

KINDERCARE requested team G-27 to develop specifications for a system that has two user interfaces, a web-based interface and a terminal interface.

Both interfaces will require their users to sign in before they can access the system.

The web-based interface is basically designed for the teacher. The teacher will register a pupil before they can attempt the assignments that will be uploaded. During pupil registration, the teacher will be required to capture the pupil’s first name, last name, phone number and user code.

While on the web-based interface, the teacher will upload assignments which will be attempted by the pupils. He/she is only allowed to add up to 8 characters to the assignment list for the pupils to attempt via CLI. Furthermore, the teacher also specifies the start and end time of the assignment.

At any given time, the teacher may deactivate a pupil and that pupil shall not have access to the uploaded assignments even when they are registered. For that pupil to be able to access and attempt the uploaded assignments again, the teacher must have to confirm their sent request first.

The interface also enables a teacher to view pupil’s scores in a given assignment and attach comments. More so, the teacher should be able to view necessary reports that can help them in decision making.

The second interface will be the Command Line where the pupils will be interacting with the system. A registered pupil can view and attempt the ongoing assignments that have been uploaded by the teacher. An ongoing assignment bears the number of characters contained in it and also shows the time left before it closes. Once an assignment closes, the pupil can no longer view or attempt it.

While on the CLI, the pupil is expected to attempt the assignment by entering either “0” or “1” in the 4x7 matrix which will be presented one character after the other. The “0” means a star (\*) should not be printed in a cell while a “1” means that a star should be printed in a cell. On completion of a character, the pupil submits that character and the time taken to complete it is recorded. The total time taken to attempt the whole assignment is calculated and printed on the screen on submission of the last character together with an acknowledgment of submission and all figures in star form.

**FIGURE 2.1 IS A CONTEXT LEVEL DIAGRAM SHOWING THE OVERVIEW OF**

**KINDERCARE ASSESSMENT SYSTEM.**

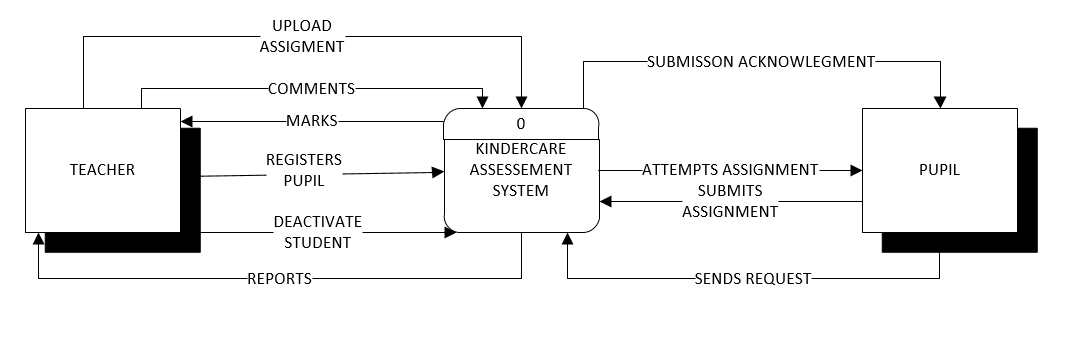


figure 2. a context level diagram of KINDERCARE system

# SYSTEM ARCHITECTURE

## Architectural design

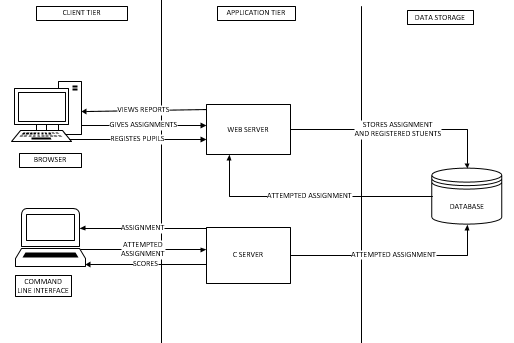


figure 3.1.1 shows the architectural design of our system.

**Browser**

The teacher is connected to a web server therefore logs in into the web interface and is able to register a student, upload assignments, deactivate a student, view scores and reports.

**Command line interface**

The pupil logs into the system via the command line and is able to view the assignments, send requests for activation, submit attempted assignment and view scores and comments made by the teacher.

## Decomposition Description

**FIGURE 3.2.1 IS A CONTEXT DIAGRAM SHOWING KINDERCARE ASSESSMENT SYSTEM.**

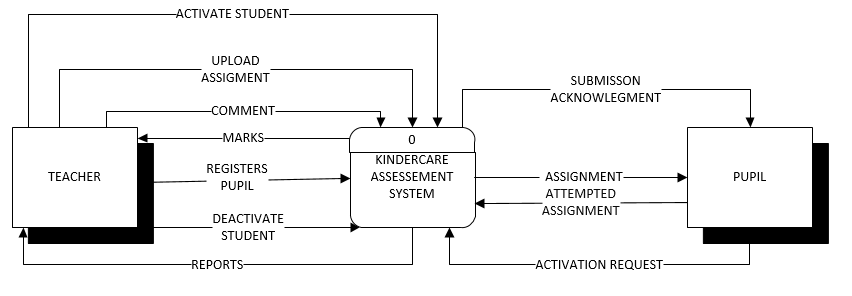


figure 3.2. This shows the two entities interact with the system.

The diagram above shows the boundaries of the software and data flows between the system and external entities. The teacher and the pupil are external entities who interact with the system.

The teacher registers a pupil by entering the pupil information namely first name, last name, phone number and user code and he/she can deactivate a registered pupil at any time. He/she also uploads assignments into the system for the pupils to attempt, views the marks, adds comments to them and views reports from the system.

A pupil attempts an assignment where they receive an acknowledgment and the mark scored on submission of that particular assignment.

**FIGURE 3.2.2 IS A LEVEL 0 DFD SHOWING THE EXPLODED KINDERCARE ASSESSMENT SYSTEM.**

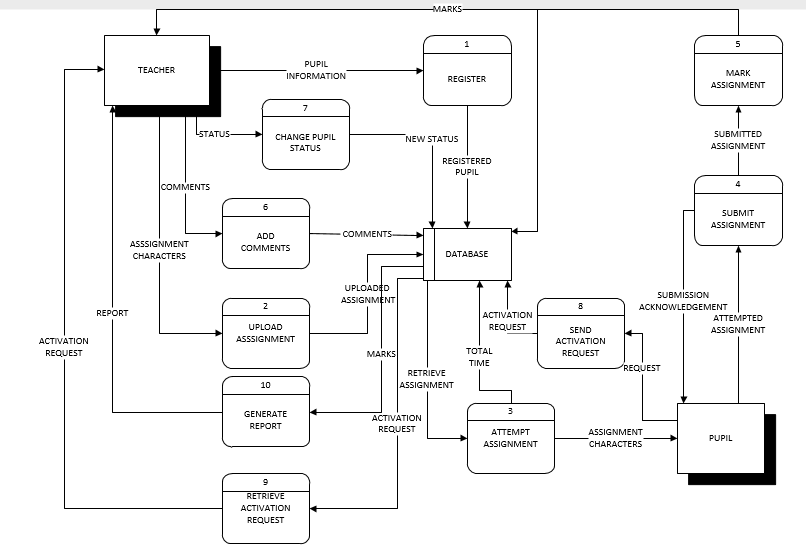


figure 3.2. A level 0 diagram showing exploded KINDERCARE assessment system

The figure 3.2.2 shows the various processes, dataflows and data stores in the system. The processes are numbered from 1-8 as shown.

Process 1:

This receives pupil information from the teacher, registers the pupil and stores the data in the system’s database.

Process 2:

This receives the assignment characters from the teacher and uploads them to the database.

Process 3:

This retrieves the assignment that was uploaded by the teacher from the database so that the pupil can attempt it. This process has been exploded as shown in the figure 3.2.4.

Process 4:

This process allows the pupil to submit their assignment after they have finished attempting all the characters in the assignment. On submitting the attempted assignment, a submission acknowledgement is sent to the pupil and the characters are printed on the pupil’s screen.

Process 5:

This process allows the system to mark the assignment automatically and sends the marks scored by a particular pupil to the database where they can be retrieved and viewed by the teacher.

Process 6:

This process allows a teacher to attach comments to the marks scored by a particular pupil in an assignment. Which comments are stored in the database and are viewed by the pupil when they next log in.

Process 7:

This process allows to change the status of a pupil namely Active and inactive. When a teacher changes the status of a pupil, the changes are also affected in the database against the specified pupil.

This is exploded in the diagram as shown in figure 3.2.3.

Process 8:

This allows the deactivated pupil to send a request to the teacher for activation. This request is sent to the database and is in turn reflected on the teacher’s screen when they next log in.

Process 9:

This process retrieves the deactivated pupil’s requests from the database and these are in turn reflected on the teacher’s screen.

Process 10:

Based on the marks scored from the assignments attempted by the pupils, a report is generated and is sent to the teacher’s screen where they can view when they next login.

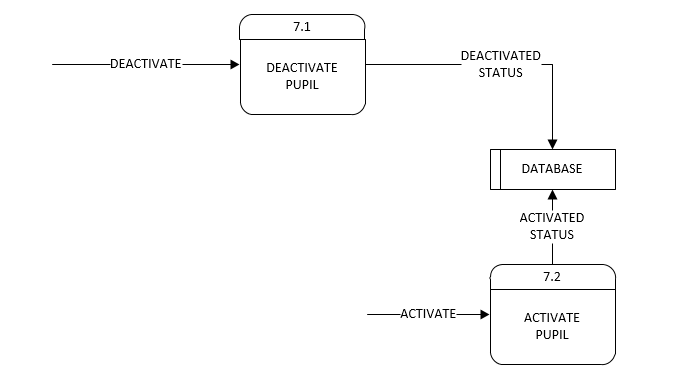


figure 3.2. decomposition diagram of process 7

A teacher can deactivate a pupil at any time. This deactivated pupil will not be able to attempt the assignments uploaded by the teacher on the system.

For the pupil to be able to access and attempt the assignments again, he/she will have to send a request to the teacher, which teacher may accept or decline the request. In case the teacher accepts the request, the status of the given deactivated pupil changes from “deactivated” to “active”. After this, the pupil will able to attempt the uploaded assignments again.

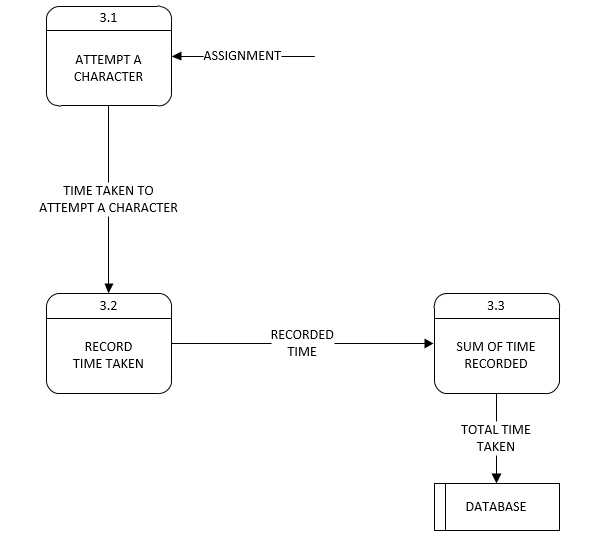


figure 3.2. decomposition diagram of process 3

After the assignment has been uploaded by the teacher, a student is able to see it on the command line interface. This assignment bears the time left until it can expire and the number of characters in that very assignment. This assignment is attempted in a way that a pupil can attempt/do one character after the other as shown in process 3.1. When the pupil starts to attempt the first character, a timer starts and it stops when he/she submits that character so they can do the next character in the assignment. The time taken to finish a particular character is recorded as shown in process 3.2 until the pupil submits the final character where the total time taken to do all the characters in the assignment is added and recorded in the database.

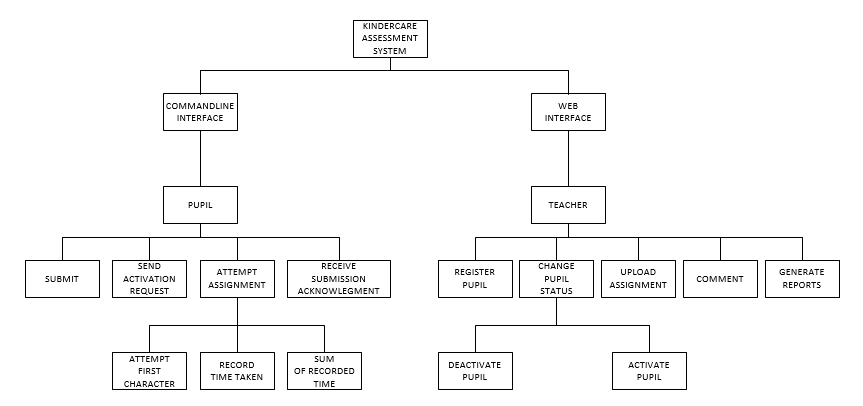


figure 3.2. a functional decomposition diagram of the system

# Data design

## Data description

The KINDERCARE assessment system has got two interfaces; The web-based interface and the command line interface

* The Web-based interface

|  |  |
| --- | --- |
| Data | Description |
| Username | * This is a string that contains different characters and uniquely identifies the users of the system. |
| Password | * This is a string that contains different characters and some special characters that identifies a user given the username. |
| Pupil’s first name | * This is a string of characters that makes the pupil’s first name. |
| Pupil’s Last name | * This is a string of characters that are in the pupil’s last name. |
| Pupil’s phone number | * This is a long integer that is used to store the pupil’s phone number |
| Pupil’s user code | * This is a string of characters that uniquely identified a pupil. |

table 4.1.1 table shows the data from the web-based interface

* Command line interface

|  |  |
| --- | --- |
| Data | Description |
| Pupil’s user code | * This is a string of characters that uniquely identifies a pupil |
| Pupil’s password | * This is a string that uniquely identifies a pupil when logging in to the system with the pupil’s user code. |
| Assignment attempting codes | * “1” and “0”, these are characters used by the pupil on the command line interface to complete the 4x7 matrix. |
| commands | * These are long texts and short texts used by the pupil to interact with the system. |

table 4.1.2 shows the data involved with the command line interface.

## Data dictionary

This represents a relational database which will be used to store the data in the KINDERCARE assessment system. Below are the relations that are to be found in the database

Pupil

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Constraint | Description |
| Pupil\_usercode | Varchar (5) | Primary key | Uniquely identifies a pupil |
| Pupil\_password | Varchar (15) | Not null | Identifies the pupil uniquely with the pupil\_usercode |
| First Name | Varchar (20) | Not null | Contains the first name of the registered pupil |
| Last Name | Varchar (20) | Not Null | Contains the last name of the pupil being registered |
| Phone number | Varchar (20) | Not null | Contains phone number of the pupil being registered |

table 4.2.1 describes data about the pupil

Teacher

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Constraint | Description |
| Teacher\_ID | Integer | Primary key | Uniquely identifies a teacher |
| TeacherName | Varchar(50) | Not null | Contains the name of the teacher |
| Password | Varchar(15) | Not null | Contains the password of the teacher to be used when logging in. |

table 4.2.2 shows data about a teacher.

Assignments

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Constraint | Description |
| Assignment\_ID | Integer | Primary key | Uniquely identifies an assignment |
| AssignmentName | Varchar (20) | Not null | Contains the name of an assignment that has been uploaded |
| StartTime | Date | Not null | Contains the start date of the assignment |
| EndDate | Date | Not null | Contains the end date of the assignment |
| CharacterNumber | Integer | Not null | Number of characters contained in a specific assignment. |

table 4.2.3 holds data about an assignment

Scores

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Constraints | Description |
| Pupil\_usercode | Varchar (5) | Foreign key | References the pupil in the pupil table |
| Assignment\_ID | Integer | Foreign key | References the assignment in the assignment table |
| Score | Integer | Not null | Contains the scores from the assignment |

table 4.2.4 holds data about the marks scored by the pupil in an assignment

Activation Requests

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraints | Description |
| Request\_ID | Integer | Primary key | Uniquely identifies a request |
| Pupil\_usercode | Varchar (5) | Foreign key | References a pupil in the pupil table |
| Request | Varchar (30) | Not null | Contains the message to be sent t the teacher requesting for activation. |

table 4.2.5 holds data about pupil activation.

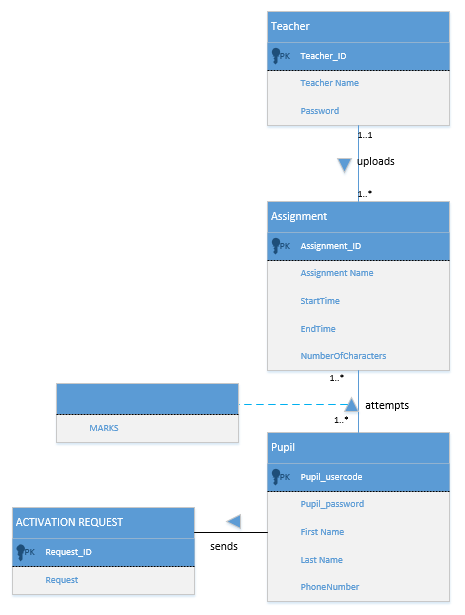


figure 4.2. an Entity relationship diagram showing the structure of the database.

# Component design

Register Pupil

This shows how a pupil is registered by a teacher

|  |
| --- |
| Start:  Enter pupil details (first name, last name, pupil\_usercode, phone number)  If details are valid  Enter details to database  Else  Request for correct details  Stop |

figure 5.1 shows the pseudocode for registering a pupil

Upload assignment

|  |
| --- |
| Start  Assignment\_ID is auto increment by the system  Enter characters (from A to Z in capital letters)  Enter start time and end time  If(characters<=8)  Store assignment in the database  Else  Request the teacher to input only 8 characters  Stop |

figure 5.2 shows the pseudocode for uploading an assignment by a pupil

Marking Assignment

|  |
| --- |
| Start  Let b= true  Count= 1  For every character submitted by pupil  If (character submitted is equal to character in database)  B= true  Count is incremented  Else  B= false  Marks will be equal to count  Stop |

figure 5.3 shows the pseudocode for marking an assignment submitted by a pupil

Adding a comment

|  |
| --- |
| Start  Retrieve marks from the database to teacher’s interface  Enter comment besides the mark  The comment is stored against the mark by student  Stop |

figure 5.4 shows the pseudocode for adding a comment by a teacher

Attempting an assignment

|  |
| --- |
| Start  Retrieve assignment character from the database  While number of characters submitted <total number of characters in the assignment  Present next character  Stop |

figure 5.5 shows the pseudocode for presenting the different characters when attempting the assignment

# Human interface design

## Overview of the human interface design

This system basically consists of two major interfaces, the web-based interface which is to be used by the teachers and the command line interface which id to be used by the pupils.

**Web interface**

When this page has loaded, the teacher will be prompted to create an account so as to be registered in the system. This requires the teacher’s name, username and a password. On submitting these details, the teacher’s account will be created in case they filled all the fields correctly or the page will reload and ask them to enter the data and fill all the fields provided.

A login page is then loaded where the teacher will be required to fill on the username and password as they filled in while registering. The home page for the website is loaded if the login information matches the credentials in the database.

While on the home page, the teacher will be able to upload assignments, set the start and end time for those particular assignments and specify the number of characters in that assignment.

The page has also got a menu where the teacher can access all the other functionalities that cannot be seen while on the home page. This menu has objects like Dashboard, assignments, register pupil, report, grades and settings.

While the menu, the teacher can select the register object which will allow him/her to register a particular pupil by entering the pupil’s first name, last name, phone number and user code.

The teacher can select the dashboard object, this will load the home page where the he/she can set and upload assignments. The teacher can also select the assignments to view the ongoing assignments and the time left to expire. The teacher can also delete an ongoing assignment which will then be deleted form the pupil’s screen.

The teacher can view the marks scored in a particular assignment, time taken by a particular pupil to attempt that assignment and he/she can attach a comment which will be seen by the pupil when they next login via command line interface. A report can also be accessed by the teacher by pressing on the report button which will be generated consisting of the summary of the previously attempted assignments.

A teacher will click the pupil status under menu where they can deactivate a pupil. The deactivated pupil can then send a request to be activated and this request will be found under the same page.

**Command Line Interface**

This interface is designed to be used by the pupil. Firstly, a registered pupil will be required to create an account. Using the user code that was registered by the teacher, the pupil will only be required to set a password which will be used with the user code to login for every session.

After the pupil has logged in, they will be able to view and attempt the uploaded assignments. This assignment shows the number of characters contained in it and the time left for that assignment to expire. An expired assignment will not be visible to the pupil and therefore cannot be attempted.

While attempting the assignment, the pupil will be presented with one character at a time. This means that a pupil will attempt one character at a time until the assignment is complete. To attempt this assignment, the pupil will be presented with a 4X7 matrix(table) where they will be required to draw those characters by placing “1” and “0” in the different cells of the matrix.

Upon submission of the last character in the assignment, the pupil will then receive a submision acknowledgment. This message will come along with the characters in the finished assignment and will be printed on the screen in form of stars for the pupil to see.

A pupil can be deactivated by the teacher at any time. He/she will not be able to attempt anymore assignments until they have been reactivated by the teacher. For a pupil to be activated by the teacher, he/she will have to use the “RequestActivation” command to send a request to the teacher who will then decide whether to activate them or not.

The pupil can also use the following commands to do the following tasks;

|  |  |
| --- | --- |
| **COMMANDS** | **ACTION** |
| Viewall | Displays assignment number and date showing if attempted or not. |
| Checkstatus | Summarizes all assignments (how many attempted, average scored, percentage missed, percentage attempted etc.) |
| Viewassignment [assignmentid] | See details of a specified assignment |
| Checkdates [datefrom] [dateto] | Returns assignments within a specified range |
| RequestActivation | Request teacher to activate the pupil |

table 6.1.1 shows the command available to the pupil and their actions

## Screen images

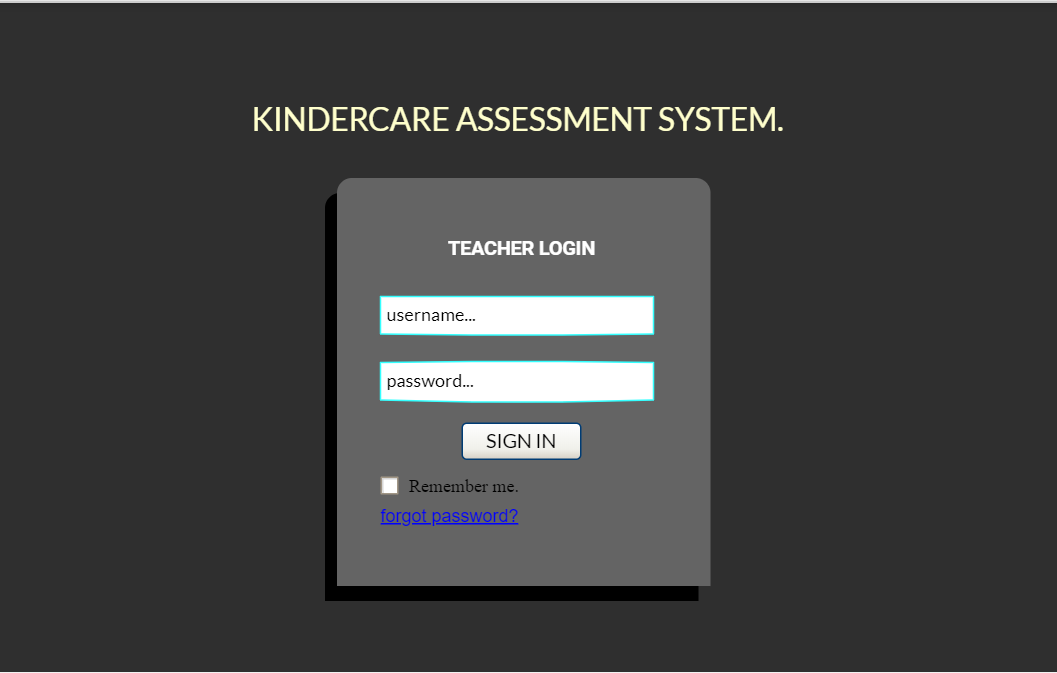


figure 6.2.1 shows the teacher’s login page

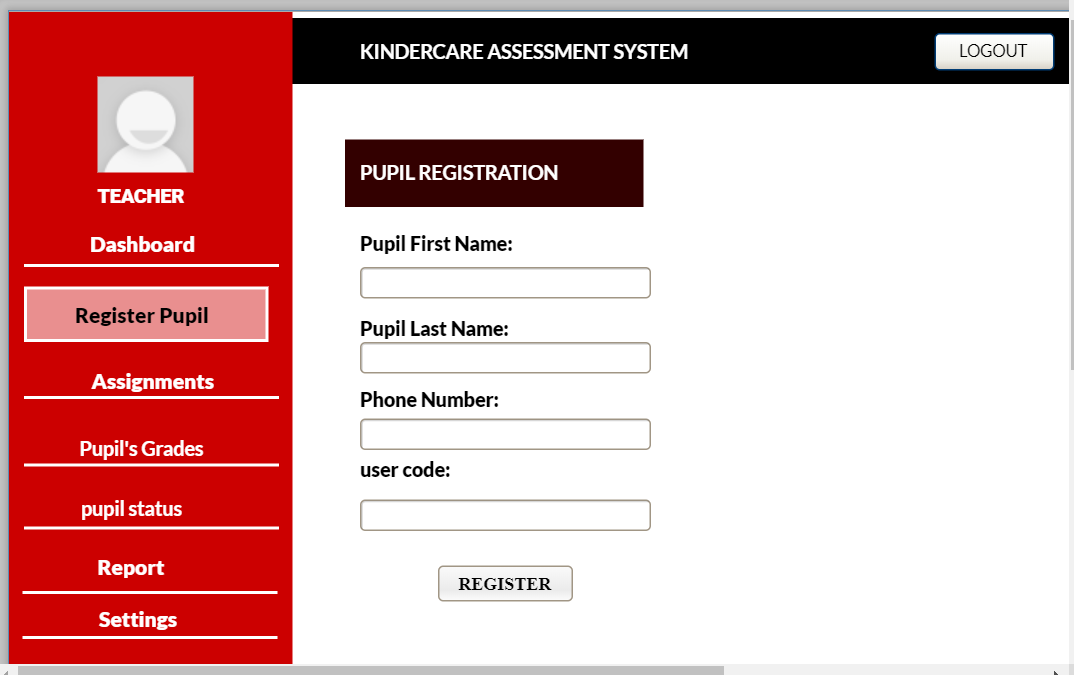


figure 6.2.2 shows how the teacher registers a pupil

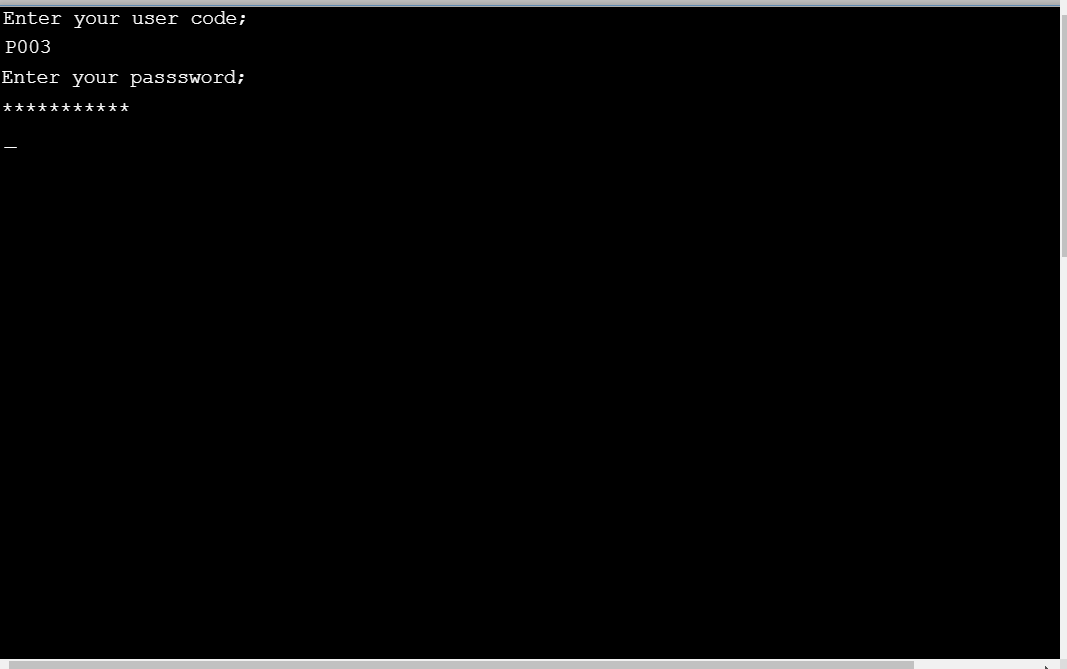


figure 6.2.3 shows login page for a pupil via command line interface

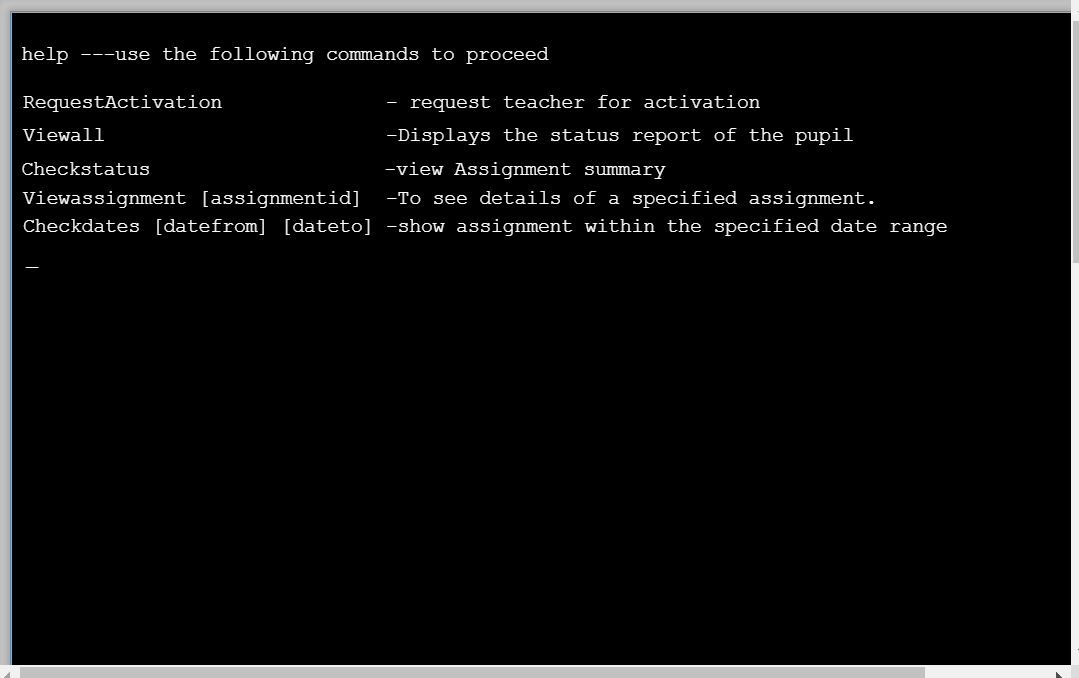


figure 6.2.4 show commands available for the pupil on the command line interface

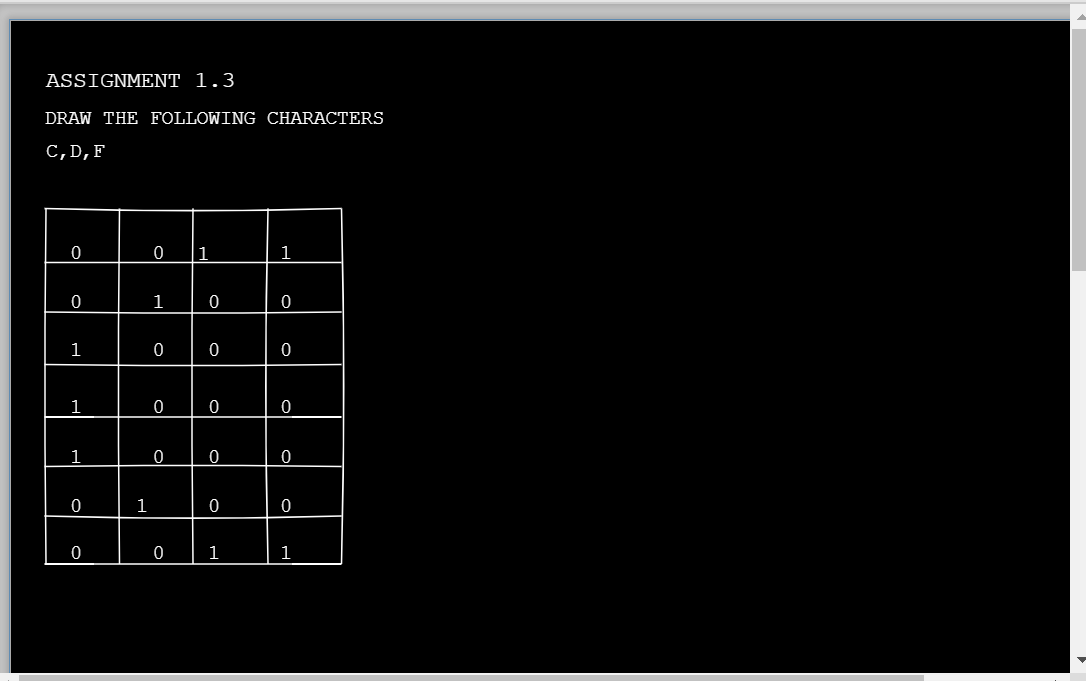


figure 6.2.5 shows how the pupil will be completing the matrix

## Screen objects and actions

|  |  |  |
| --- | --- | --- |
| **Object** | **Action** | **Page** |
| Text box | Provide a place for the user to enter data | Teacher login page, pupil registration page |
| Register Button | To confirm operations of the user by entering the user data into the system | pupil registration page |
| Sign in button | To enter the data entered by the user for verification by the system | Teacher login page |
| Logout button | Enable a teacher to exist the system | Teacher homepage |
| Check box | Enable the user to select an available option on the system | Teacher login page |

table 6.3. shows screen objects and actions