|  |  |  |
| --- | --- | --- |
| NAME | REGISTRATION NUMBER | STUDENT NUMBER |
| KAKOOZA EMMANUEL DAVID | 20/U/2096/EVE | 2000702096 |
| KIMERA MARVIN SHEDRACK | 20/U/7821/PS | 2000707821 |
| TENDO SHAUN KENDRONE | 20/U/7810/PS | 2000707810 |
| NANTABO HILDAH GERTRUDE | 20/U/0172 | 2000700172 |

**GitHub link**: <https://github.com/HildahN/recess-projects>

**G-15**

**Supervisor: Mary Nsabagwa**

**DESIGN DOCUMENT FOR KINDERCARE APPLICATION.**

TABLE OF CONTENTS

[**LIST OF TABLES** ii](#_Toc93835223)

[**LIST OF FIGURES** iii](#_Toc93835224)

[**1.** **Introduction** 1](#_Toc93835225)

[**1.1** **Purpose** 1](#_Toc93835226)

[**1.2** **Scope** 1](#_Toc93835227)

[**1.3** **Overview** 2](#_Toc93835228)

[**1.4** **Reference Material** 3](#_Toc93835229)

[**1.5** **Definitions and Acronyms** 3](#_Toc93835230)

[**2.** **System overview** 4](#_Toc93835231)

[**3.** **System Architecture** 6](#_Toc93835232)

[**3.1** **Architectural design** 6](#_Toc93835233)

[**3.2** **Decomposition Description** 7](#_Toc93835234)

[**3.3** **Design Rationale** 11](#_Toc93835235)

[**4.** **Data Design** 12](#_Toc93835236)

[**4.1** **Data Description** 12](#_Toc93835237)

[**4.2** **Data Dictionary** 13](#_Toc93835238)

[**5.** **Component Design** 15](#_Toc93835239)

[**6.** **Human Interface Design** 17](#_Toc93835240)

[**6.1** **Overview of user interface** 17](#_Toc93835241)

[**6.2** **Screen images** 19](#_Toc93835242)

[**6.3** **Screen objects and actions** 24](#_Toc93835243)

# **LIST OF TABLES**

[Table 1: Letter C printed on the screen 7](#_Toc93785522)

[Table 2: Commands 7](#_Toc93785523)

[Table 3: Data description for the web interface 14](#_Toc93785524)

[Table 4: Data description for the command-line interface 14](#_Toc93785525)

[Table 5: Assignment 14](#_Toc93785526)

[Table 6: Grade 15](#_Toc93785527)

[Table 7: Pupil 15](#_Toc93785528)

[Table 8: Teacher 15](#_Toc93785529)

[Table 9: Screen objects and actions 25](#_Toc93785530)

# **LIST OF FIGURES**

[Figure 3. 1: The architectural design of the KinderCare Application. 9](#_Toc93786020)

[Figure 3.2. 1 Functional Decomposition Diagram for KinderCare Application 7](#_Toc93841721)

[Figure 3.2. 2: Context diagram for the KinderCare application 8](#_Toc93841722)

[Figure 3.2. 3 Level 0 Data Flow Diagram for the KinderCare application 9](#_Toc93841723)

[Figure 3.2. 4: Registration system for the KinderCare application 10](#_Toc93841724)

[Figure 3.2. 5: Assignment system for the KinderCare application 10](#_Toc93841725)

[Figure 3.2.6: Grading system for the KinderCare application 11](#_Toc93841726)

[Figure 3.2.7: Reporting system for the KinderCare application 11](#_Toc93841727)

[Figure 4.1. 1 Entity Relationship Diagram for the KinderCare application 12](#_Toc93787006)

[Figure 5.1: Grading algorithm for the KinderCare application 14](#_Toc93786973)

[Figure 5. 2: Assignment algorithm for the KinderCare application 14](#_Toc93786974)

[Figure 5. 3 Reporting algorithm for the KinderCare application 15](#_Toc93786975)

[Figure 5.4: Registering algorithm for the KinderCare application 15](#_Toc93786976)

[Figure 6.1.1: a summary of web interface 20](#_Toc93786056)

[Figure 6.2.1: This shows the log in page 19](#_Toc93831068)

[Figure 6.2.2: This shows the sign-up page. 19](#_Toc93831069)

[Figure 6.2.3: this shows registration form on the teacher's dashboard. 20](#_Toc93831070)

[Figure 6.2.4: This shows the assignment form 21](#_Toc93831071)

[Figure 6.2.5: this shows what the Grades tab. 21](#_Toc93831072)

[Figure 6.2.6: this shows what is in the report tab. 22](#_Toc93831073)

[Figure 6.2.7: this shows what is in the activation tab. 22](#_Toc93831074)

[Figure 6.2.8: command line interface 23](#_Toc93831075)

# **Introduction**

## **Purpose**

This software design document describes the architecture and systems design of the LSL system. The system is to help lower school children to remember and create different shapes of characters ranging from A to Z (All in capital letters). The system will be used by teachers and the pupils.

## **Scope**

The system is comprised of two interfaces. The web interface is used by the teacher to give assignments to pupils, to register the pupils and the teacher can also be able to activate and deactivate a registered student. A command line interface is used by the pupils to attempt the available assignments after which an acknowledgement of submission will be given to the pupils. A pupil can request for activation and get a report.

**Goals and objectives**

* The teacher should be able to give assignments that the pupils will attempt.
* The system should be able to auto grade the assignments submitted by the students and give feedback to the teacher and the pupil respectively.
* The teacher should be able to comment on the pupils’ grades which comment will be seen by the pupil.
* The assignments should be timed so that the pupils can attempt them in the projected time by the teacher.
* The pupils once activated should be able to attempt the available assignments.

**Benefits**

* The system will help pupils to remember and create different shapes of characters.
* The system will save time by auto grading results so that pupils and teachers get results back immediately.
* The auto report generation feature will help teachers in decision making.

## **Overview**

This software design document has been organized into chapters as illustrated below

Chapter 1: This chapter shows the purpose of the software design document and its intended audience, provides a description and scope of the software with explanations of its goals, objectives and benefits.

It also lists reference materials used as sources of information and provides definitions of all terms, acronyms and abbreviations used in the SDD.

Chapter 2: This chapter gives a general description of the functionality, context and design of the project and provides pieces of background information.

Chapter 3: This chapter describes details about the architectural design of the system both diagrammatically and through written down concepts, provides a decomposition of the subsystems in the architectural design and discusses the rationale for selecting the architecture described.

Chapter 4: This chapter gives the data description of the system and alphabetically lists the system entities along with their data types and descriptions in a data dictionary.

Chapter 5: This chapter shows design of each component and what it does in the system.

Chapter 6: This chapter provides an overview of the user Interface, display of screen images and a discussion of screen objects and actions associated with those objects.

Chapter 7: This chapter illustrates a requirements matrix through providing a cross-reference that traces components and data structures to the requirements in the SRS document and uses a tabular format to show which system components satisfy each of these requirements.

Chapter 8: This chapter includes the appendices directly to provide supporting details to aid understanding in the Software Design Document.

## **Reference Material**

SDD-ieee-1016-2009. (2009). New York, 3 park avenue, United States of America.

Tilley, S., & Rosenblatt, H. J. (2017). *System Analysis and Design.* Shelly Cashman.

## **Definitions and Acronyms**

SDD – Software Design Document.

SRS – Software Requirements Specification.

# **System overview**

The KinderCare system is to help lower school children to remember and create different shapes of characters ranging from A to Z (all in capital letters). It comprises two interfaces namely the web interface and the command-line interface. The teacher logs in using the web interface while the pupil logs in using the command-line interface.

The teacher uses the web interface to register pupils and give them assignments to. A pupil can attempt all assignments that have been submitted by the teacher as long as they are still within the required time once registered. The teacher at a given time can deactivate a pupil. A deactivated pupil cannot attempt assignments even if they are registered. The pupil should send a request to the teacher through the command line interface to be activated.

A teacher is only allowed to add only up to 8 characters to the assignment list for the pupils to attempt. The teacher also specifies the start and end time of the assignment. Submitted assignments are automatically awarded scores and these can be seen against the pupil and the assignment when the teacher logs in. The teacher attaches a comment to the score and this is seen when the student next logs in. The teacher should be able to view the necessary reports that can help him or her in decision-making.

The pupils use a command line interface to attempt the assignment and only see those assignments, which have not expired and which are open for attempting. The pupil is informed of how much time is left to close the assignment in case the assignment is open for attempting and also how many characters they have in their assignment. The characters are presented one by one, enabling the pupil to attempt and submit.

The time taken to attempt that character is recorded for each of the characters attempted. The total amount of time taken to attempt the assignment is provided for the pupil to see after attempting the last character. An acknowledgement of submitting the assignment is provided for the pupil to see and all characters attempted are drawn on the screen (in form of stars as in Table 1). The method of providing a solution to the character is as below. The system generates characters like in table 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. 0 means the star should not be printed in the cell while a 1 means that the start will be printed in the cell such as Table 1.

Table 1: Letter C printed on the screen

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | \* | \* |
|  | \* |  |  |
| \* |  |  |  |
| \* |  |  |  |
| \* |  |  |  |
|  | \* |  |  |
|  |  | \* | \* |

A pupil can enter the following commands when he or she logs in;

Table 2: Commands

|  |  |
| --- | --- |
| 1. Viewall | To display assignment number and date showing if attempted or not |
| 1. Checkstatus | To display the status report of the pupil summarizing all assignments showing how many attempted, average score, percentage missed, percentage attempted |
| 1. Viewassignment assignmentid | To see details of a specified assignment |
| 1. Checkdates datefrom dateto | To show if there is an assignment within a specified date range |
| 1. RequestActivation | To request the teacher to activate him or her |

# **System Architecture**

## **Architectural design**

**The system consists of two client machines i.e., the pupil and the teacher, the application components and the database.**

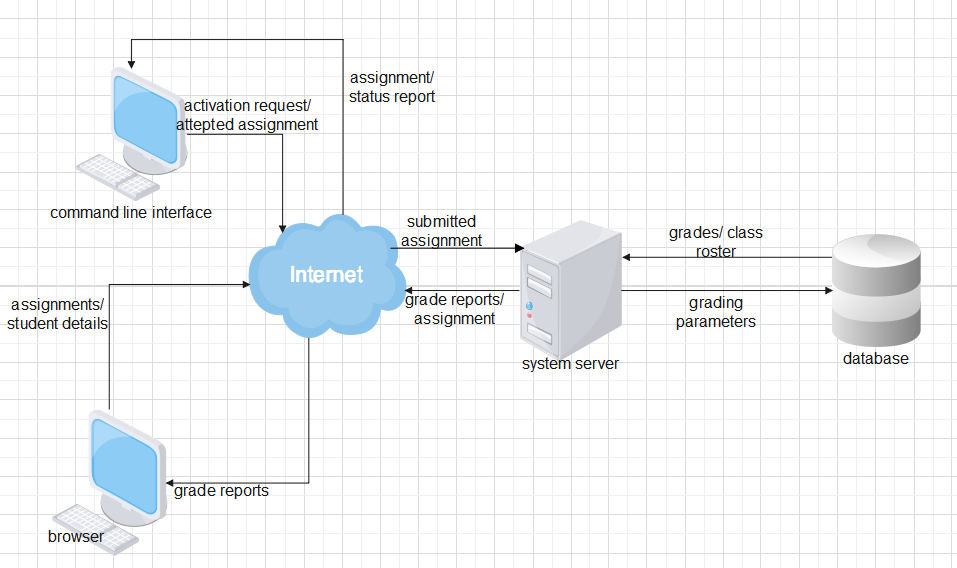


Figure 3. 1: The architectural design of the KinderCare Application.

This system has a three tiers architecture that has the clients i.e., browser and the command line, the server and the database. The internet is used to connect the server to the clients and enable seamless transfer of information between the two tiers

The command line interface which is used by the pupil sends the activation request and attempted assignment to the internet which sends it to the server. The system server sends the grading parameters to the database to grade the submitted assignment to get the pupils’ grades which are stored in the database of the system. The server also sends back the required reports and available assignments to the command line interface through the internet.

The browser (web interface) which is used by the teacher sends the student details for registration into the system and assignments to the internet which sends them to the server where they are processed. The server then sends the class roster of the registered students to the database to be stored. The server also generates grade reports according to the pupils’ grades and sends them to the browser. (SDD-ieee-1016-2009, 2009)

## **Decomposition Description**

Figure 3.2. 1 Functional Decomposition Diagram for KinderCare Application

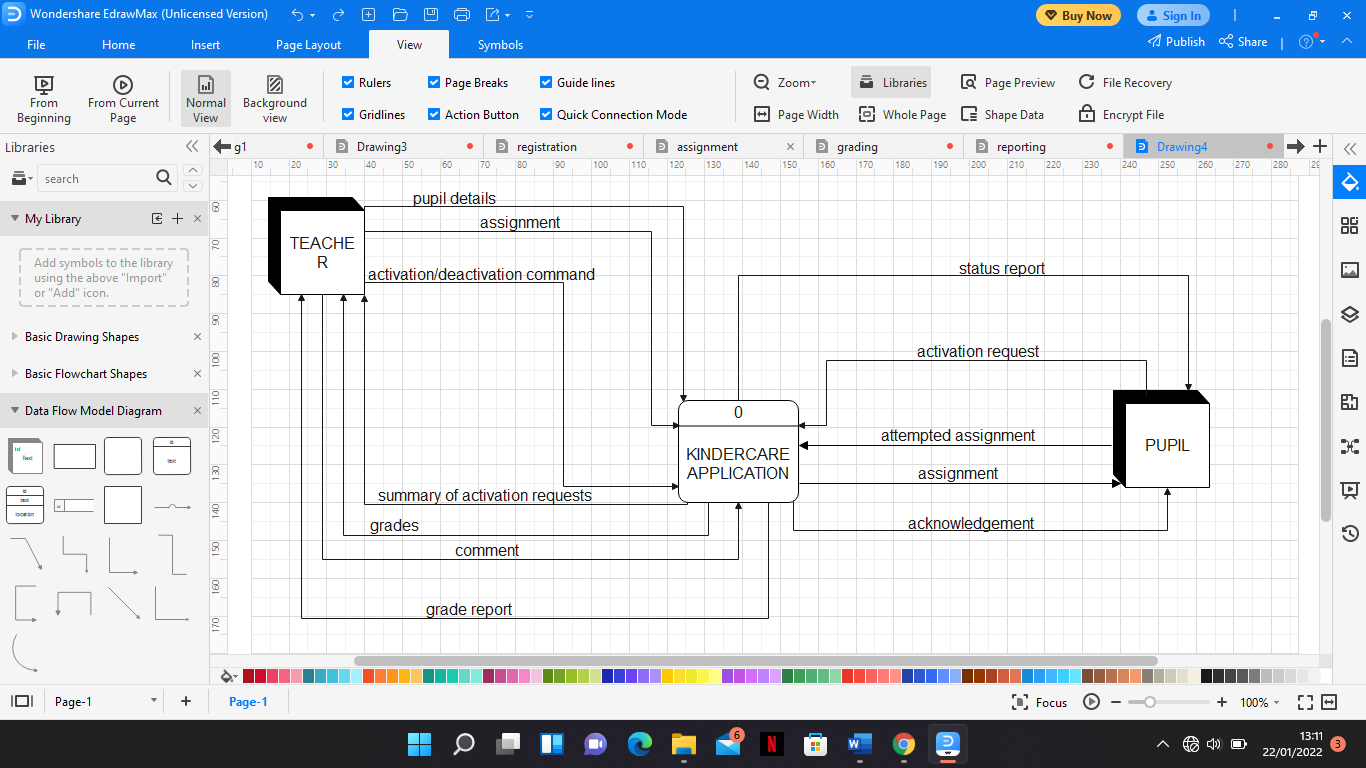


Figure 3.2. 2: Context diagram for the KinderCare application

The context diagram for the lower school learning system is shown in Figure 3.2.2. The KINDERCARE APPLICATION process is at the center of the diagram. The two entities (Pupil and Teacher) are placed around the central process. Interaction among the central process and the entities involves thirteen different data flows. The TEACHER entity has four outgoing data flows that is pupil details, assignment, activation/deactivation command and comment. The TEACHER entity also has four incoming data flows namely pupil roster, summary of activation requests, grades and the grade report.

The PUPIL entity has three incoming data flows that is status report, assignment and acknowledgement. The PUPIL entity also has two outgoing data flows namely activation request and attempted assignment.

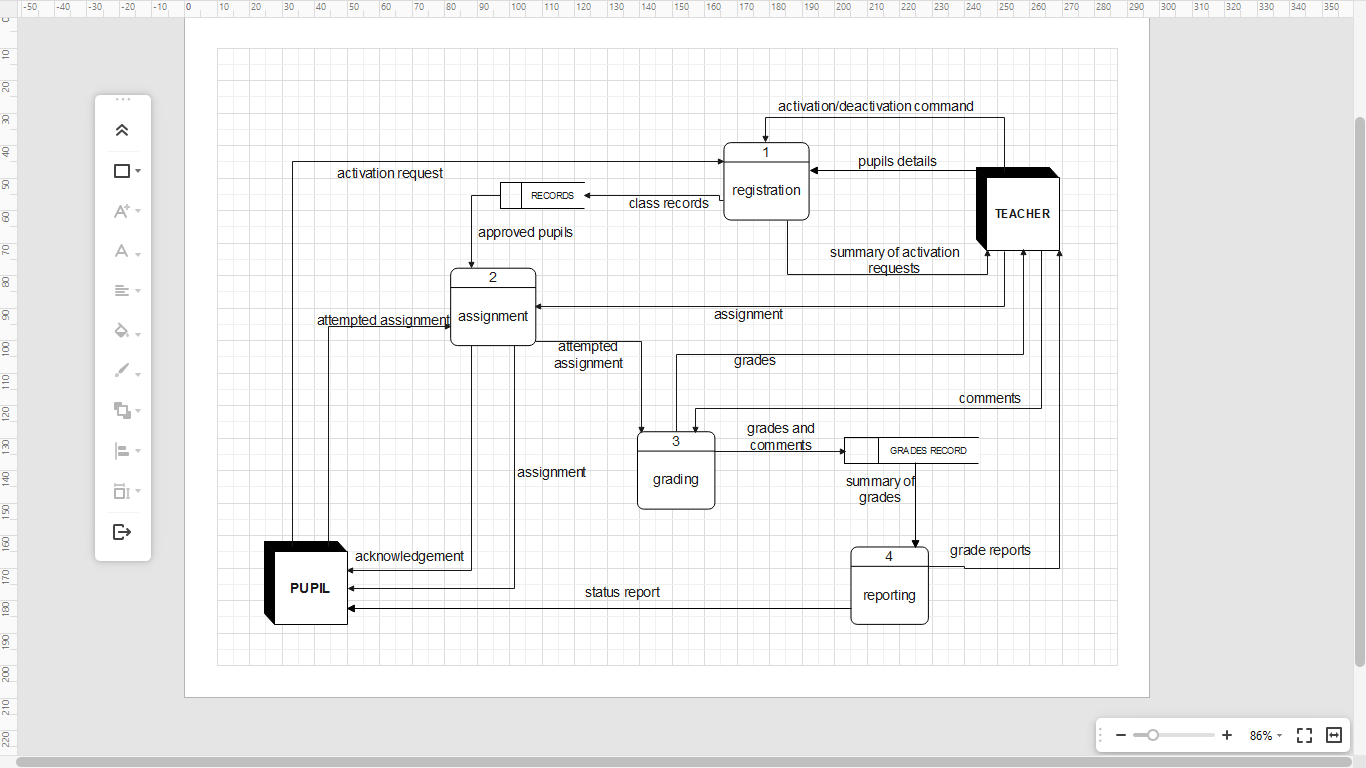


Figure 3.2. 3 Level 0 Data Flow Diagram for the KinderCare application

The level 0 data flow diagram is shown in figure 3.2.3. Process 0 on the KinderCare Application’s context diagram is exploded to reveal four processes (Registration System, Assignment System, Grading System and Reporting system), two data stores (RECORDS and GRADES RECORD) and two additional data flows (class records and approved pupils).

The KinderCare level 0 data flow diagram is then levelled and balanced to generate a level 1 data flow diagram in which breaks down the four processes (Registration System, Assignment System, Grading System and Reporting system) further into distinct processes and data stores respectively.

Levelling will enable all functional primitives to be identified and balancing maintains consistency among the set of Data Flow Diagrams. (Tilley & Rosenblatt, 2017)

.

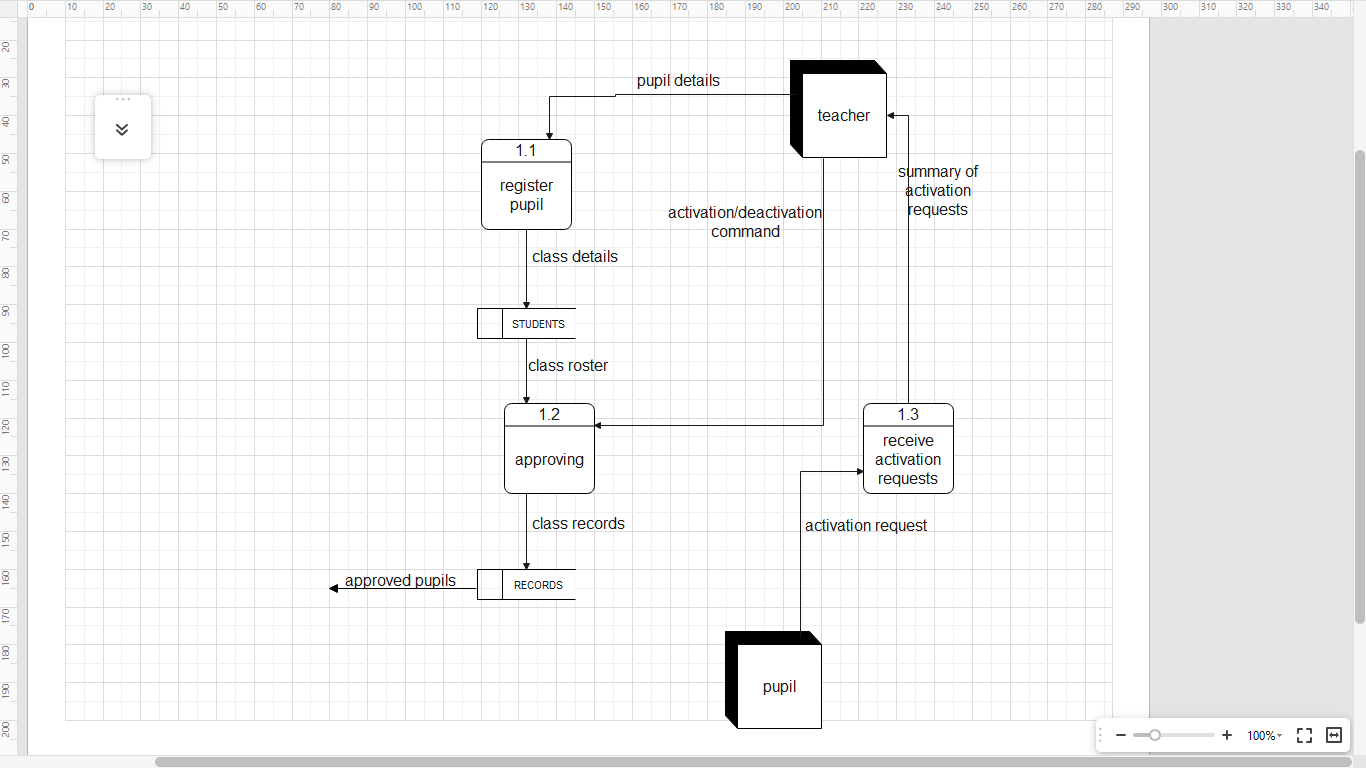


Figure 3.2. 4: Registration system for the KinderCare application

The registration system explodes into three processes (register pupil, approving and receive activation requests) and a data store (PUPILS) as shown in Figure 3.2.4 above.

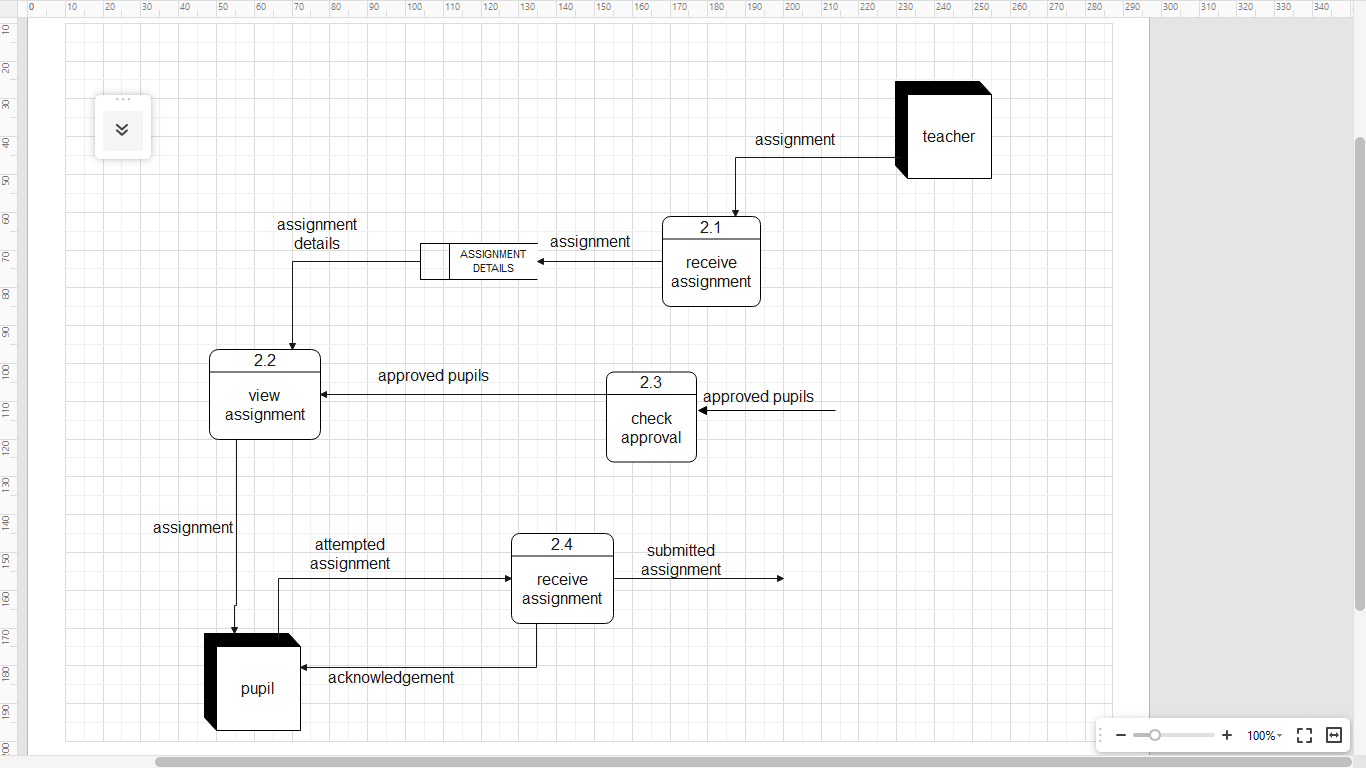


Figure 3.2. 5: Assignment system for the KinderCare application

The Assignment System explodes into four processes (receive assignment, view assignment, check approval and receive assignment) and a data store (ASSIGNMENT DETAILS) as shown in Figure 3.2.5 above.

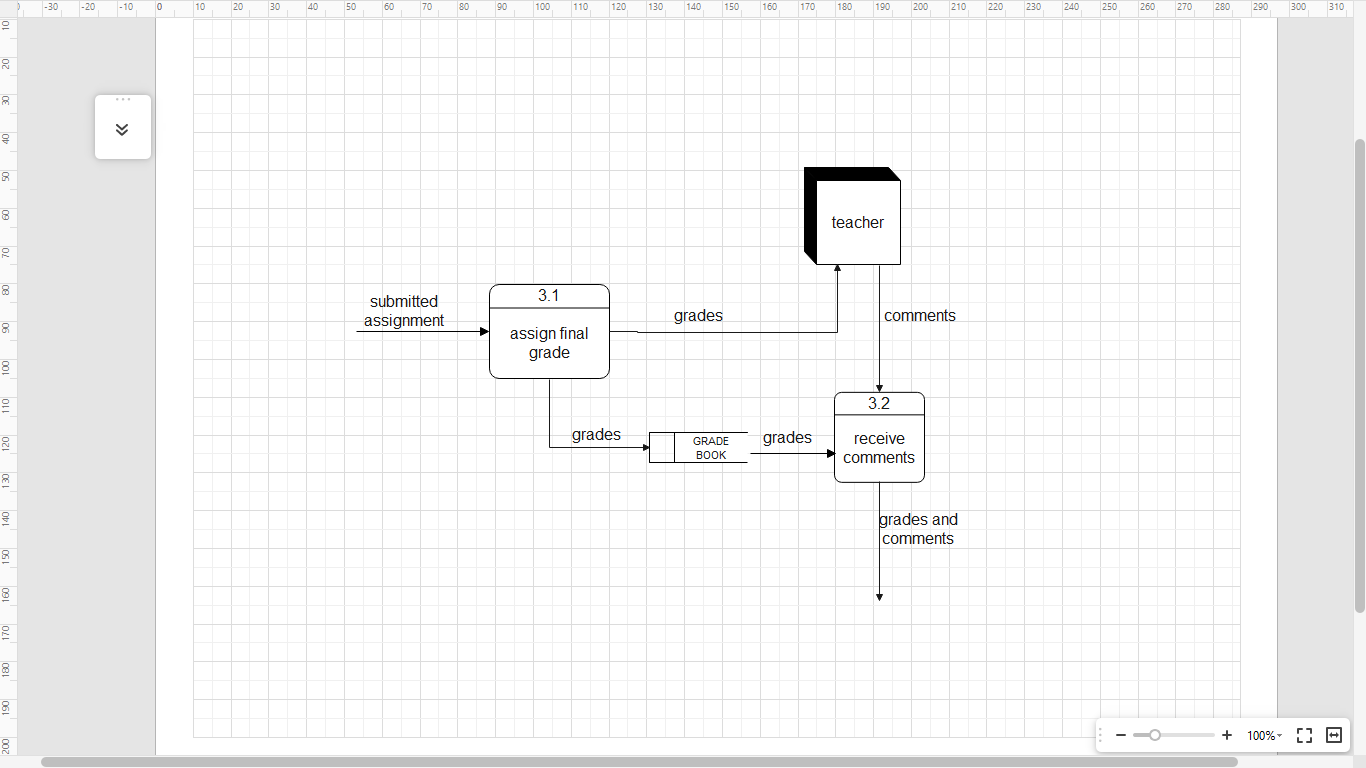


Figure 3.2.6: Grading system for the KinderCare application

The Grading System explodes into two processes (assign final grade and receive comments) and a data store (GRADE BOOK) as shown in Figure 3.2.6.

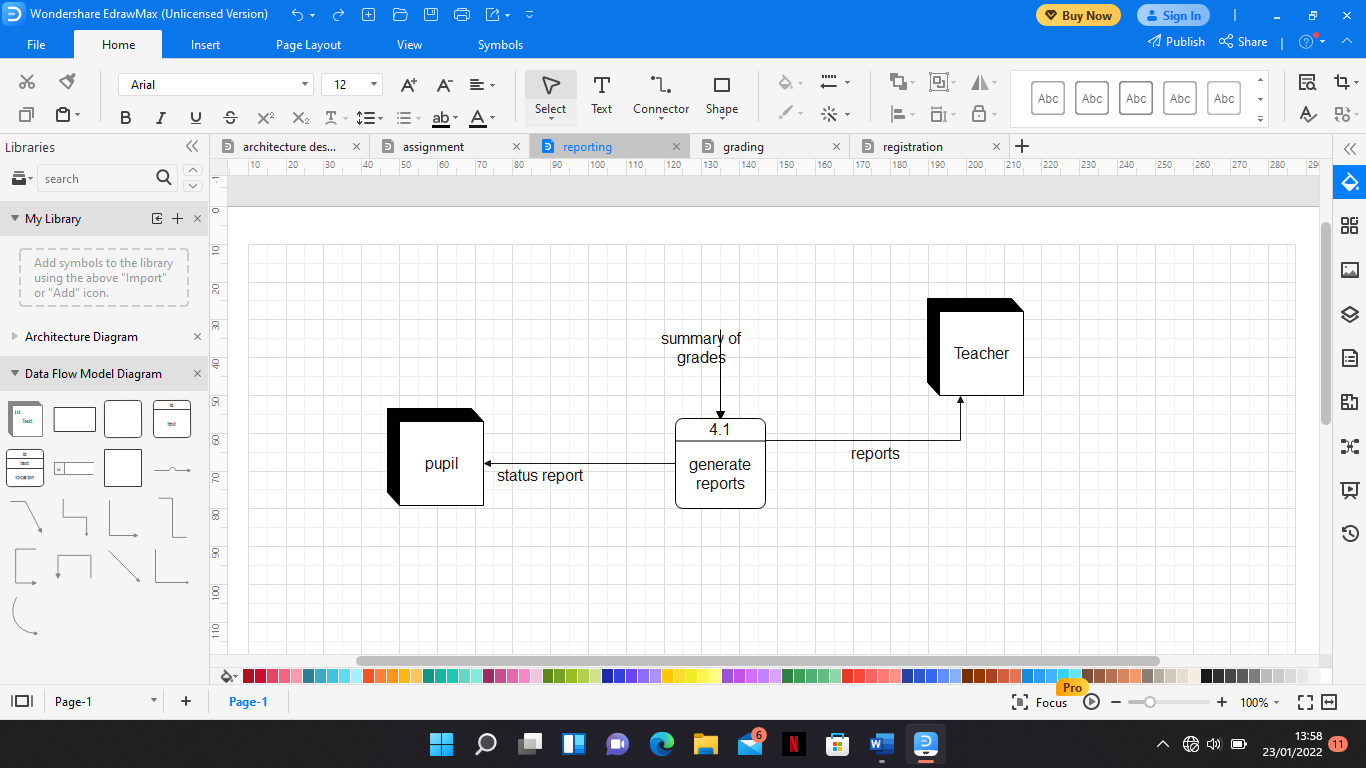


Figure 3.2.7: Reporting system for the KinderCare application

The Reporting System explodes into one process (generate reports) as shown in Figure 3.2.7 above.

## **Design Rationale**

Two approaches were considered during the design of this application that is Structured Analysis approach and Object-Oriented Analysis approach. The Structured Analysis approach was selected over the Object-Oriented Analysis approach basing on the following critical issue:

* Easier comprehension
* Functionality restrictions
* Easier Analysis

This architecture follows a top-down approach which can be more easily comprehended compared to the Object-Oriented approach.

The Object-Oriented approach has functionality restricted to only objects unlike the Structured Analysis approach which is based upon functionality.

The specifications in Structured Analysis approach are written in simple English language hence can be easily more analyzed by non-tech savvy people unlike the Object-Oriented approach.

# **Data Design**

## **Data Description**

The KinderCare Application comprises two interfaces namely web interface and command-line interface. The web interface is used by the teacher while the command-line interface is used by the pupil.

Table 3: Data description for the web interface

|  |  |
| --- | --- |
| Data | Description |
| Character combination | This enables combining of different characters in the attributes of the Teacher entity |
| Password | This holds the security key for the teacher |
| Integer | This holds the score for the pupil |

Table 4: Data description for the command-line interface

|  |  |
| --- | --- |
| Data | Description |
| Character combination | This enables combining of different characters in the attributes of the Pupil entity |
| Integer | This holds the whole numbers of the phone number for the pupil |
| Time | This holds the start and end time of the assignment |
| Date | This holds the current date for the assignment |

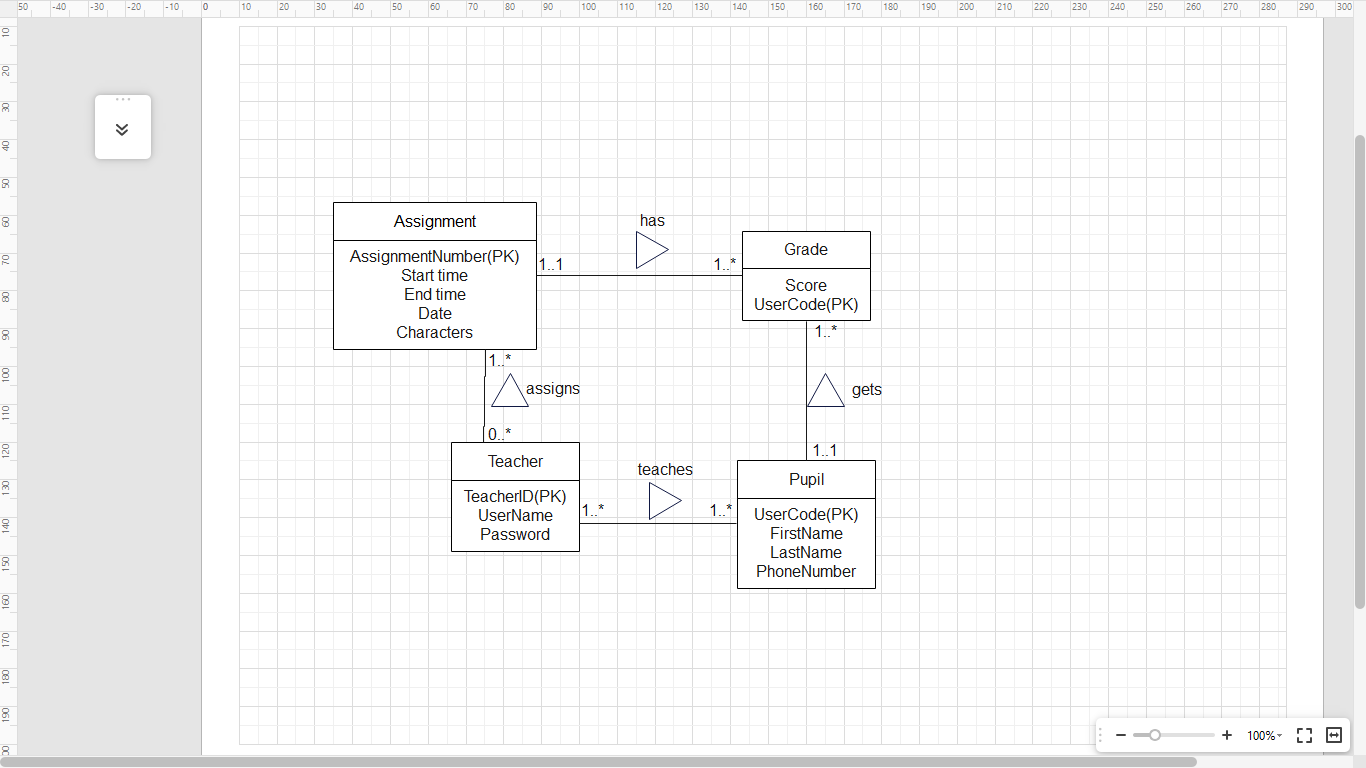


Figure 4.1. Entity Relationship Diagram for the KinderCare application

## **Data Dictionary**

Table 5: Assignment

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Datatype | Constraint | Description |
| Assignment Number | varchar | Primary Key | Uniquely identifies each assignment |
| Start time | time | Not Null | Shows the starting time for the assignment |
| End time | time | Not Null | Shows the ending time for the assignment |
| Date | date | Not null | Shows the date for the assignment |
| Characters | varchar | Not Null | It holds either a 1 or 0 for the pupil to attempt the assignment. |

Table 6: Grade

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Datatype | Constraint | Description |
| Score | Integer | Not Null | Shows the score for each pupil |
| Assignment Number | Varchar | Foreign key | References the assignment that is set by the teacher |
| User Code | Varchar | Primary Key and Foreign key | Uniquely identifies each pupil and references a pupil that is being registered by a teacher |

Table 7: Pupil

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Datatype | Constraint | Description |
| First Name | Varchar | Not Null | Identifies the first name of the pupil |
| Last Name | Varchar | Not Null | Identifies the last name of the pupil |
| Phone Number | Integer | Not Null | Identifies the phone number of the pupil |
| User Code | Varchar | Primary Key | Uniquely identifies each pupil |

Table 8: Teacher

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Datatype | Constraint | Description |
| Teacher ID | Varchar | Primary Key | Uniquely identifies each teacher |
| User Name | Varchar | Not Null | Identifies the name of the teacher |
| User Code | Varchar | Foreign key | References a pupil that is being registered by a teacher |
| Password | Varchar | Not Null | Protects a teacher’s account and any information it contains |
| Assignment Number | Varchar | Foreign key | References the assignment that is set by the teacher |

# **Component Design**

Grading algorithm

**Start**

Get the submitted assignment

The assignment is auto marked by the system

Determine the ratio of the number of correct characters to total number of characters in the assignment.

Convert the ratio to percentage and store as the grade

Send the grade to the teacher’s dashboard

**If** teacher comments on the grade

Store the comment on the grade in the database

**Else**

Break

**Stop**

Figure 5.1: Grading algorithm for the KinderCare application

Assignment algorithm

**Start**

Receive assignment from the teacher

Store the assignment details in the database

Check the pupils’ approval status

**If** pupil’s status is activated

Send assignment to the pupils’ dashboard

Receive attempted assignment from pupil

**Else**

Break

**Stop**

Figure 5. 2: Assignment algorithm for the KinderCare application

Reporting algorithm

**Start**

Get summary of pupils’ grade from the grading system

Generate the necessary reports i.e., progress graphs

Send a status report to the pupil on request

Send the grade reports to the teacher

**Stop**

Figure 5. 3 Reporting algorithm for the KinderCare application

Registering algorithm

**Start**

Enter pupils’ details

Store the details in the database

The teacher sends a deactivation command at any time to deactivate a student

**If** pupil is deactivated

Send an activation request

**Else**

Continue

Teacher receives a summary of the activation requests

Sends activation commands to activate a particular student at any time

Update the pupil’s records in the database

**Stop**

Figure 5.4: Registering algorithm for the KinderCare application

# **Human Interface Design**

## **Overview of user interface**

The system will be used by two users;

* The teacher
* Pupils

The system has two interfaces that the users will interact with i.e., the web interface and the command line interface.

**Web interface**

The web interface will be primarily used by the teacher. The teacher will be required to login into the system. When the teacher is done with the logging in, the web interface will then display the teacher’s dashboard which has five tabs;

Registration: this tab will open up a form that will enable the teacher to fill in the pupils’ details such as first name, last name, and phone number and user code

Assignment: this tab will have a form which will be used by the teacher to make an assignment that the pupils will be attempted. The teacher shall input the assignment number, start and ending time of a particular assignment, and also be able to select characters to be used be in a particular assignment.

Grades: this tab will display the table that summarizes the pupils’ details showing clearly the names of pupils, the assignment number, and the grade that has been scored and then gives the teacher the liberty to add a comment to the grades that have scored by the pupil

Report: this tab will show a table that will display a summary of pupils’ performance which the teacher can use for decision making. The table will show the names, number of assignments attempted, average score, percentage missed, percentage attempted.

Activation: this tab will show return the registered pupils’ details which will be displayed in a table with an option to deactivate pupil. It will also display the activation requests from the pupils

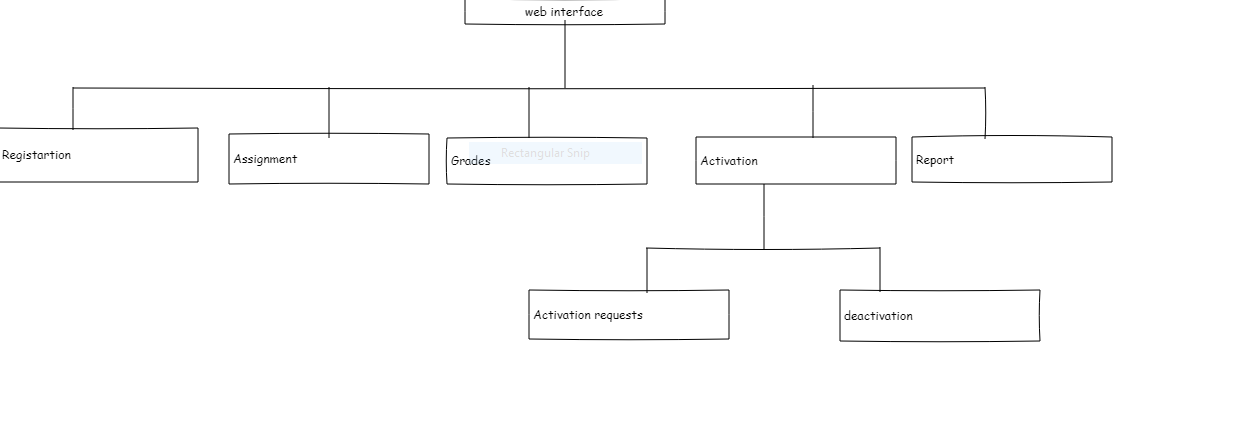


Figure 6.1.1: a summary of web interface

**Command line interface**

The command line interface will be used by the pupil. A pupil will be required to input their details

Then a menu will be displayed which has four options;

Viewall: this will display the assignment number and date, showing if an assignment is attempted or not

CheckStatus: this will display the status report summarizing all the assignments (how many attempted, average score, percentage missed, percentage attempted, etc.)

Viewassignment assignmentid: this will show details of a specified assignment

Checkdates datefrom dateto: this will show if there is an assignment within a specified date range

RequestActivation: this will be used by the pupil to request the teacher to activate him or her

## **Screen images**

****

Figure 6.2.1: This shows the log in page

Figure 6.2.1 shows the login page that the teacher will interact with. When the teacher fills in their username and password, he or she clicks on the log in button. On clicking it the username and password are checked from the database to confirm if they are correct.

In case a teacher does not yet have an account, he or she can click on the sign-up text and follow instructions on the sign-up page.



Figure 6.2.2: This shows the sign-up page.

Figure 6.2.2 shows the sign-up page A teacher using the interface for the first time will enter the information (Teacher ID, Username, password, confirm password) as required and click on the sign-up button. In case the teacher already has an account, he or she can click on the log in text and follow instructions on the log in page.

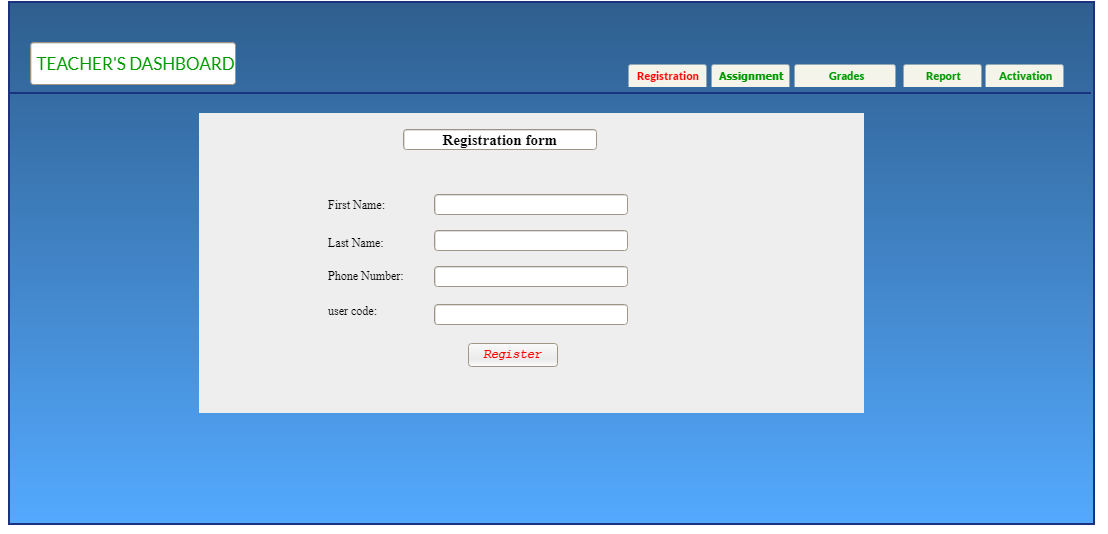


Figure 6.2.3: this shows registration form on the teacher's dashboard.

Figure 6.2.3 shows the contents in the registration form on the teacher’s dashboard. The teacher will input pupils’ details here and click on the Register button. On clicking the button, the details are stored in the database.

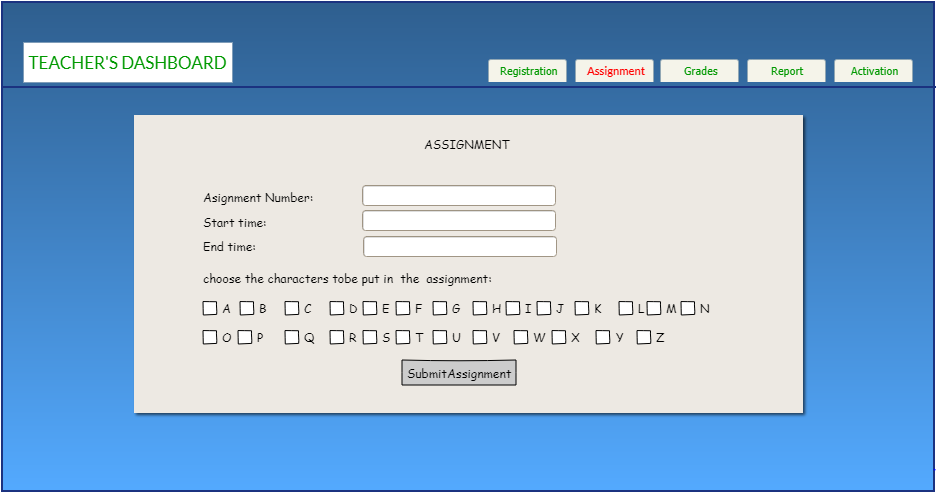


Figure 6.2.4: This shows the assignment form

Figure 6.2.4 shows the how the contents of the assignment form will look like. The teacher is required to input the assignment number, specify the start and end time of a particular assignment and also select characters in a particular assignment.

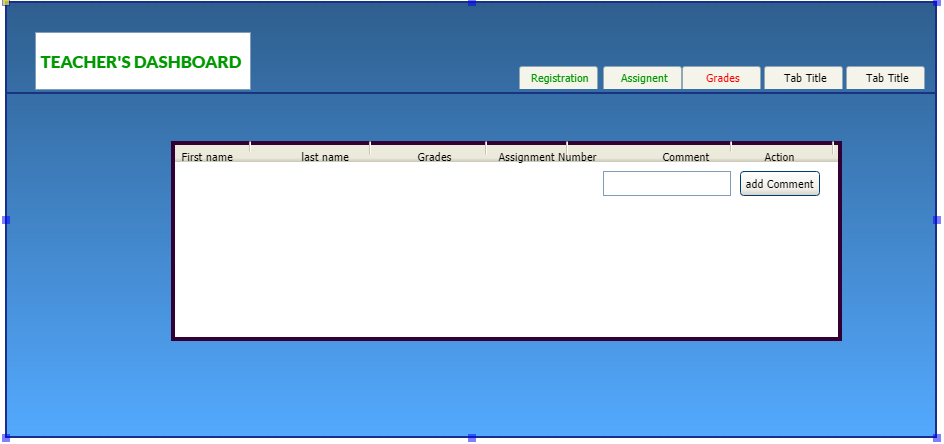


Figure 6.2.5: this shows what the Grades tab.

Figure 6.2.5 shows the Grades tab. The grade for each pupil is shown where the teacher can add a comment. The comments are then stored in the database together with the grades to generate grade reports.

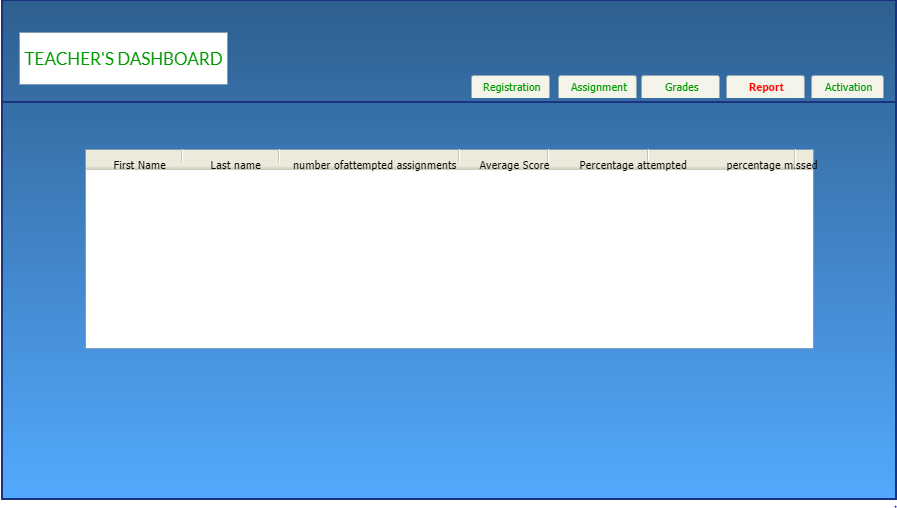


Figure 6.2.6: this shows what is in the report tab.

Figure 6.2.6 shows the contents of the report tab where a teacher will be able to view a summary of pupils’ performance which he/she will use for decision making.

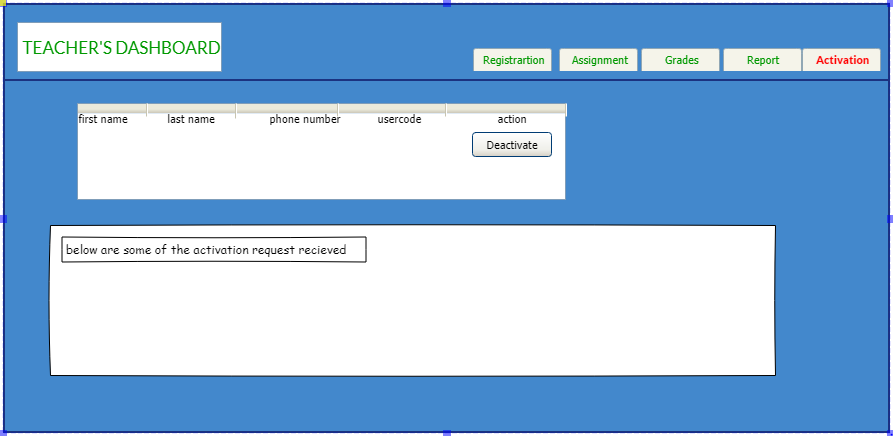


Figure 6.2.7: this shows what is in the activation tab.

Figure 6.2.7 shows a table displaying the details of all the registered pupils is displayed to the teacher, a teacher can be able to deactivate a pupil.

A summary of activation request is also displayed below.

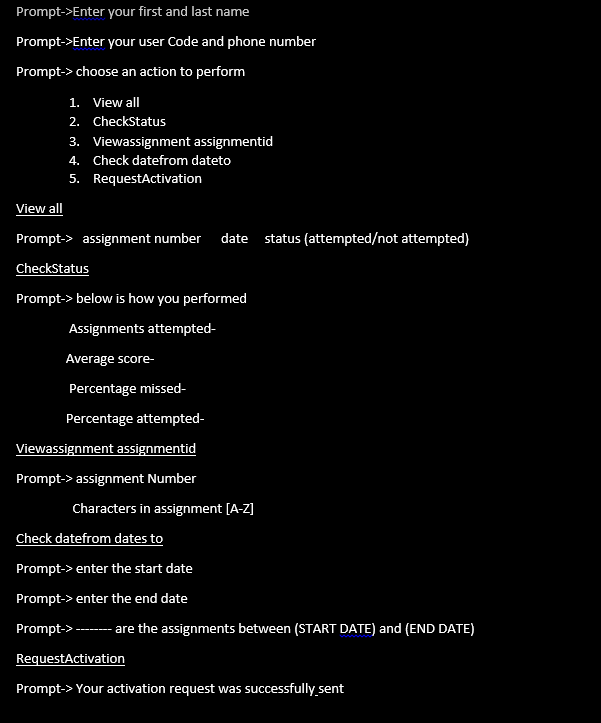


Figure 6.2.8: command line interface

Figure 6.2.8 shows how the command line interface will look like when the pupil logs in.

## **Screen objects and actions**

Table 9: Screen objects and actions

|  |  |  |
| --- | --- | --- |
| **object** | **name** | **Action** |
|  | Log in button | it will submit the log in details to the server through PHP commands to allow a teacher to access the web interface |
|  | Sign up button | It will submit the sign-up details of the teacher who is using the interface for the first time |
|  | Log in link | It will provide a link to the log in page from the sign-up page |
|  | Sign up link | it provides a link to the sign-up page from the log in page |
|  | Register button | It will submit the pupil’s details that a teacher has submitted to be registered |
|  | Submit assignment button | It submits the assignment details that teacher has intended to give to the pupils |
|  | Add comment button | it will add a comment to the grades of a particular pupil that the teacher has attached |
|  | Deactivate button | It is used by teacher to disenable a registered pupil from attempting assignments |