

DESIGN DOCUMENT FOR KINDERCARE APPLICATION

Group Name: G-13

Project Title: KinderCare Application

Github Link: <https://github.com/KigwanaAugustine/G-13-KINDER-CARE-APPLICATION.git>

Supervisor: Mary Nsabagwa

Name	Registration Number	Student Number
Zijja Aggrey Marvin	20/U/0167	2000700167
Kigwana Augustine	20/U/0168	2000700168
Duku Allan Juma	20/U/7800/PS	2000707800
Achuka Allan Simon	20/U/19779/PS	2000719779

TABLE OF CONTENTS

1. INTRODUCTION	1
1.1 Purpose	1
1.2 Scope	1
1.2.1 Goals and Objectives	2
1.2.2 Benefits	2
1.3 Document Overview	2
1.3.1 Introduction:	2
1.3.2 System Overview:.....	3
1.3.3 System Architecture:	3
1.3.4 Data Design:	3
1.3.5 Component Design:	3
1.3.6 Human Interface Design:.....	3
1.4 Reference Material	4
Bibliography	4
2. SYSTEM OVERVIEW	4
3. SYSTEM ARCHITECTURE	5
3.1 Architectural Design	6
3.1.1 The Client Layer	6
3.1.2 Application Server Layer.....	7

3.1.3 Database Layer	7
3.2 Decomposition Description.....	8
3.3 Design Rationale	14
4. DATA DESIGN.....	15
4.1 Data Description.....	15
4.2 Data Dictionary	18
5. COMPONENT DESIGN	23
5.1 Teacher Registration	23
5.2 Teacher Login	24
5.3 Pupil Registration.....	24
5.4 Create Assignment	26
5.5 Pupil login	26
5.6 Attempt Assignment.....	27
5.7 Activate Pupil.....	28
5.8 Deactivate Pupil	29
6. HUMAN INTERFACE DESIGN.....	30
6.1 Overview of User Interface	30
6.2 Screen Images	31
6.3 Screen Objects and Actions.....	38

Table of Figures:

Figure 3. 1: The figure above represents a 3-client-server tier.	6
Figure 3. 2: Functional decomposition diagram.	8
Figure 3. 3: Context Level 0 diagram	10
Figure 3. 4: Level 0 Data Flow Diagram for the Kinder Care School System	12
Figure 4. 1: The entity-relationship diagram for the database.	17
Figure 5. 1: Pseudocode for teacher registration.	23
Figure 5. 2: Pseudocode for teacher login	24
Figure 5. 3: Pseudocode for registering a pupil	25
Figure 5. 4: Pseudocode for assignment creation.	26
Figure 5. 5: Pseudocode for pupil login.....	27
Figure 5. 6: Pseudocode for attempting assignment	28
Figure 5. 7: <i>Pseudocode for teacher send character function</i>	28
Figure 5. 8: Pseudocode for activating a pupil	29
Figure 5. 9: Pseudocode for pupil deactivation	29
Figure 6. 1: Graphical representation of the user interface.....	30
Figure 6. 2: Login page for the KinderCare application	31
Figure 6. 3: Signup page: Displayed on clicking the register here link on the login page.	32
Figure 6. 4: Register Pupil page: Displayed on teacher logging in to the system.	33
Figure 6. 5: Create assignment page: displayed on clicking create assignment link on the side menu.....	34

Figure 6. 6: Pupil Results page: displayed on clicking the pupil result link on the side menu.	35
Figure 6. 7: Reports; displayed on clicking the Reports link on the side menu.	36
Figure 6. 8: Pupil Management page: displayed on clicking the pupil management link on the side menu.	37
Figure 6. 9: Command line interface for the KinderCare assignment system	37

Table of Tables

Table 4. 1: Shows the data elements that will be in the system.....	16
Table 4. 2: Teacher	18
Table 4. 3: Pupil.....	19
Table 4. 4: Assignment	20
Table 4. 5: Activation request.	21
Table 4. 6: Character.....	21
Table 4. 7: Marks	22
Table 4. 8: Assignment Character.....	22
 Table 6. 1: Screen objects and their actions.....	 39

1. INTRODUCTION

1.1 Purpose

This software design document describes the architecture and system design of the KinderCare School Application. The KinderCare application will be used by teachers of KinderCare to help lower school children remember all capital letters. The system will enable teachers to submit assignments through a web interface to the pupils who will then attempt those assignments through a Command Line interface. The system will then automatically grade the pupil's work and provide the results to the teachers.

The audience of this document are the KinderCare School Management, the software engineers, and developers who will be working on this system, and supervisors of this project.

1.2 Scope

This application has two main parts. The first part is a web interface accessed by the teachers to post assignments for the pupils and generate appropriate performance reports. The other part is a command line interface used by the pupils to attempt the posted assignments which are automatically graded by the system.

The system will use a three-tier client-server architecture where the clients will be the web interface and the command line interface. Then we will have an application server which will interpret the commands from the clients to a form understandable by the database, which is also the final part of the system.

This system will be deployed as a stand-alone system i.e. it will not interact with any other external systems.

1.2.1 Goals and Objectives

- Provide a cheap system to help lower-school children remember and create different shapes of characters.
- Provide readily accessible data in the form of reports about children's academic progress for better decision-making.
- Provide a system that is to ensure that pupils are assessed diligently and properly.

1.2.2 Benefits

- Reduced time of grading the pupil's assignments.
- Efficient assistance of teachers in tracking student performance and progress.
- Better and quick feedback for the pupils and the teachers as well.
- Improved decision-making through analysis of provided pupil reports.

1.3 Document Overview

This software design document has been organized into chapters as described below.

1.3.1 Introduction:

Describes the purpose of the Software Design Document, the scope of the project, goals and objectives of the project, and finally the benefits of undertaking the project. It also introduces the various chapters of the Design Document through the Document overview section.

1.3.2 System Overview:

This chapter will provide a high-level description of how the system will carry out its work, and what the system is generally able to do.

1.3.3 System Architecture:

This chapter talks about the architectural design of the system, the functional decomposition of the system, and justification for the choices of the design.

1.3.4 Data Design:

This chapter shows what data will be gathered from the users of the system as they interact with it through the two interfaces. This chapter also shows how the data gathered will be organized so as to be stored in the database.

1.3.5 Component Design:

This chapter gives a functional description of each major component of the KinderCare application. It also includes algorithms of how the system is to execute these major tasks.

1.3.6 Human Interface Design:

This chapter shows the way users are to interact with the system as they use it to perform their task. This chapter also provides the screen images for each of the components of the user interface.

1.4 Reference Material

Bibliography

Thomas Conoly, Carolyn Begg. (n.d.). *Database systems: A practical guide to design, impementation and management*.
Tiley RosenBlatt. (2016). *Systems Analysis and Design*. Shelly Cashman Series.

2. SYSTEM OVERVIEW

The KinderCare application is to help lower school children remember and create the shapes of all Capital letters (A-Z). (Thomas Conoly, Carolyn Begg)So, it will provide an interface for the teachers to submit assignments to evaluate the pupil's learning and another interface for the pupils to attempt these assignments.

The two interfaces are the command line interface for the pupil use and the web interface for the teacher use.

The teacher should register the pupils using a web interface and submit assignments that are stored in the database till the time they can be attempted. The teacher is allowed to add only 8 characters to an assignment. The teacher also specifies the start and end time of the assignment. The teacher can deactivate a pupil.

A registered pupil who is activated should use a command line interface to attempt all assignments that have been submitted by the teacher when the start time specified by the teacher has reached.

If a pupil is deactivated, he/she can send a request through the command-line interface to be activated by the teacher.

The pupil is informed of how many characters they have to attempt in their assignment. The characters are submitted one by one enabling the pupil to attempt and submit. The time taken to attempt each is recorded. The pupil is informed of the time left to close the assignment as he/she is attempting the assignment.

An acknowledgment of submission showing the total time taken to attempt all characters is seen by the pupil after attempting the last character. All characters attempted are drawn on the screen in the form of stars.

The teacher should see the awarded scores against each submitted assignment and attach a comment to the score. The pupil can see the commented scores when they next log in. The system should produce necessary reports to help teachers in decision making.

The student is required to login and use the command line interface to enter the following commands to use the system;

Viewall to display assignment number and date, showing if attempted or not.

Checkstatus to display the status report of the pupil summarizing all assignments (how many attempted, average score, percentage missed, percentage attempted etc.)

Viewassignment assignmentid to see details of a specified assignment.

Checkdates datefrom dateto to show if there is an assignment within a specified date range.

RequestActivation used for pupil to request a teacher to activate him or her.

3. SYSTEM ARCHITECTURE

3.1 Architectural Design

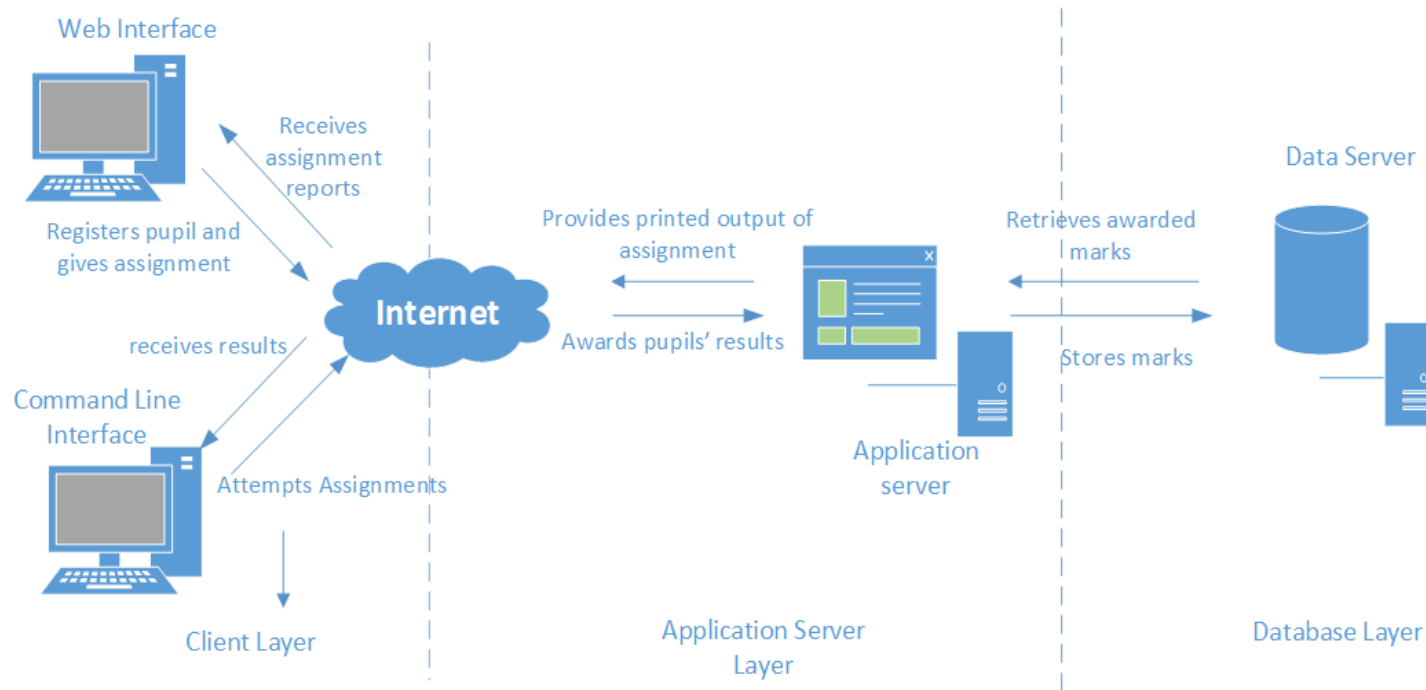


Figure 3. 1: The figure above represents a 3-client-server tier.

3.1.1 The Client Layer

It consists of the command line interface and the web interface as clients with the following functions.

Web Interface: Enables a teacher to register a pupil and also submit assignments for the pupil to attempt. It should also enable a teacher to comment on a pupil's results, and also to activate or deactivate a pupil.

Command Line Interface: It is to be used by a pupil to attempt assignments, request activation, check on their marks, view all assignments they have to do, and check for assignments that may be within a specified range of dates.

3.1.2 Application Server Layer

It processes the clients' requests and translates them into data access commands that can be understood and carried out by the server.

3.1.3 Database Layer

It stores data about the clients and processes that occur in the system.

3.2 Decomposition Description

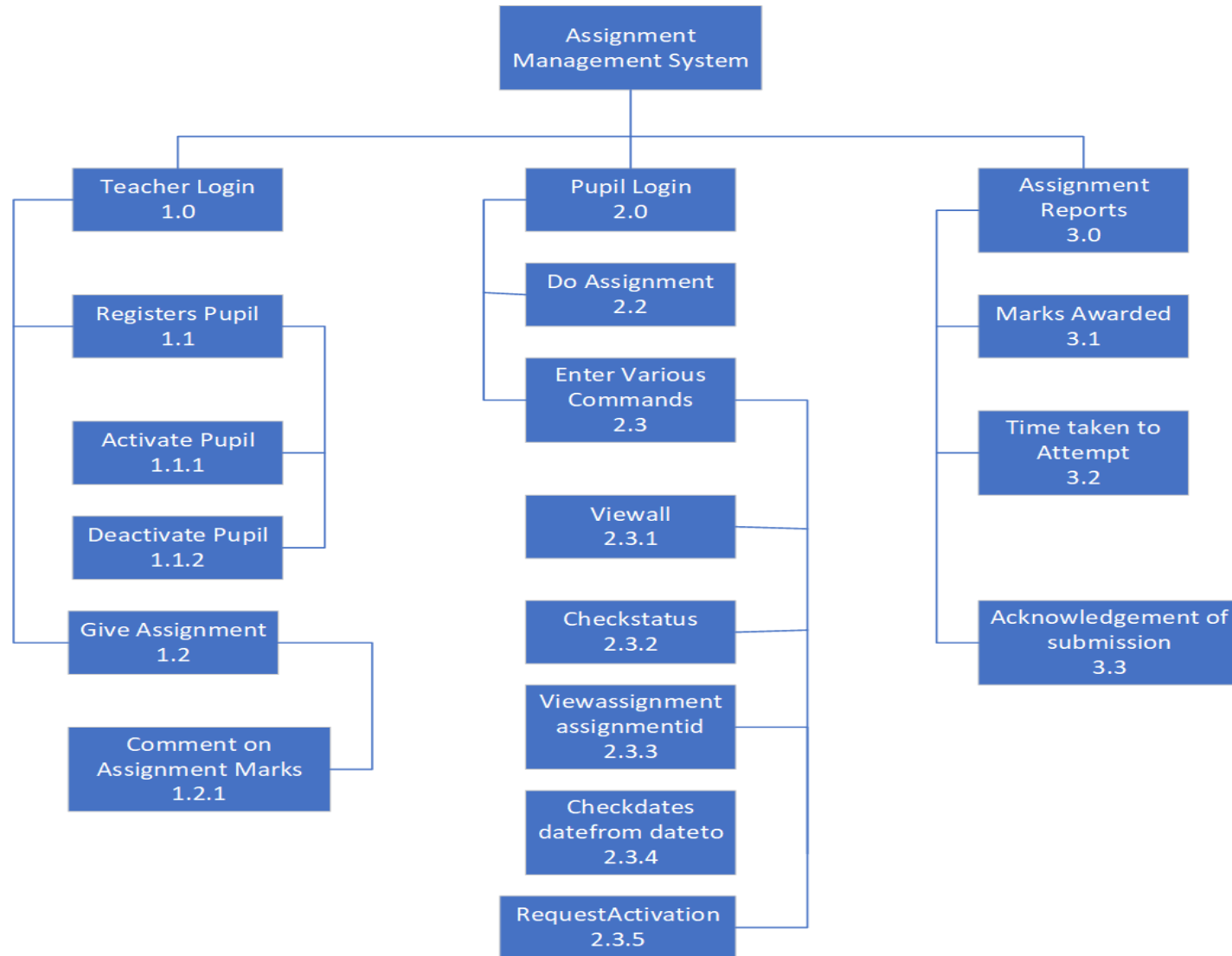


Figure 3. 2: Functional decomposition diagram.

Figure 3.2 is a functional decomposition diagram that helps show the breakdown of processes in our project and their dependencies with each other.

The teacher uses a web interface to login and register pupils into the system. The teacher can activate and deactivate registered pupils. The teacher can give assignments and details about them such as start and end time. A teacher can comment besides the assignment upon receiving assignment marks.

The pupil uses the command line interface to attempt assignments. The pupil can further use the command line interface to type various commands as follows;

View All to display assignment number and date, showing if attempted or not.

Checkstatus to display the status report of the pupil summarizing all assignments (how many attempted, average score, percentage missed, percentage attempted etc.)

Viewassignment assignmentid to see details of a specified assignment.

Checkdates datefrom dateto to show if there is an assignment within a specified date range.

RequestActivation is used for a pupil to request a teacher to activate him or her.

The KinderCare system makes reports about the assignment attempted by the pupil. It also awards marks for each assignment attempted, the acknowledgement of the assignment submission and time taken to attempt an assignment.

Context Diagram

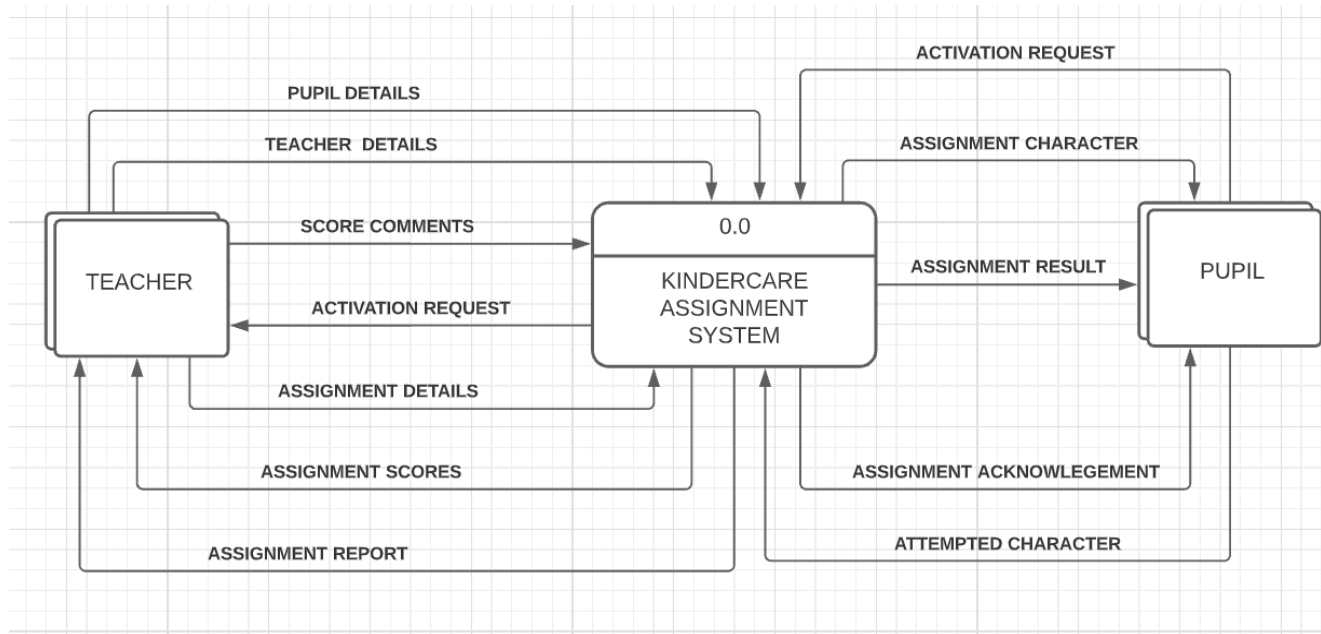
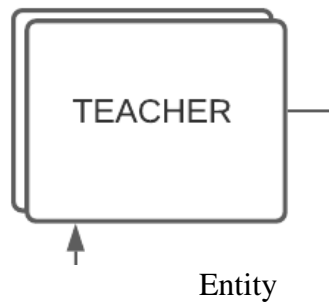


Figure 3. 3: Context Level 0 diagram

Key

→ Data flow



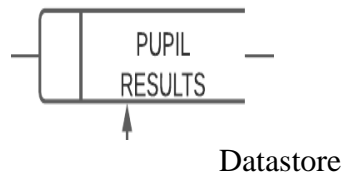


Figure 3.3 shows the context diagram of the KinderCare system. This is a top-level representation of how data will be processed in the system. It has two external entities i.e., the teacher and the pupil. The diagram has one process which is the system itself. The figure also has several data flows which shows the data that is being put into the system and the information that comes out of the system.

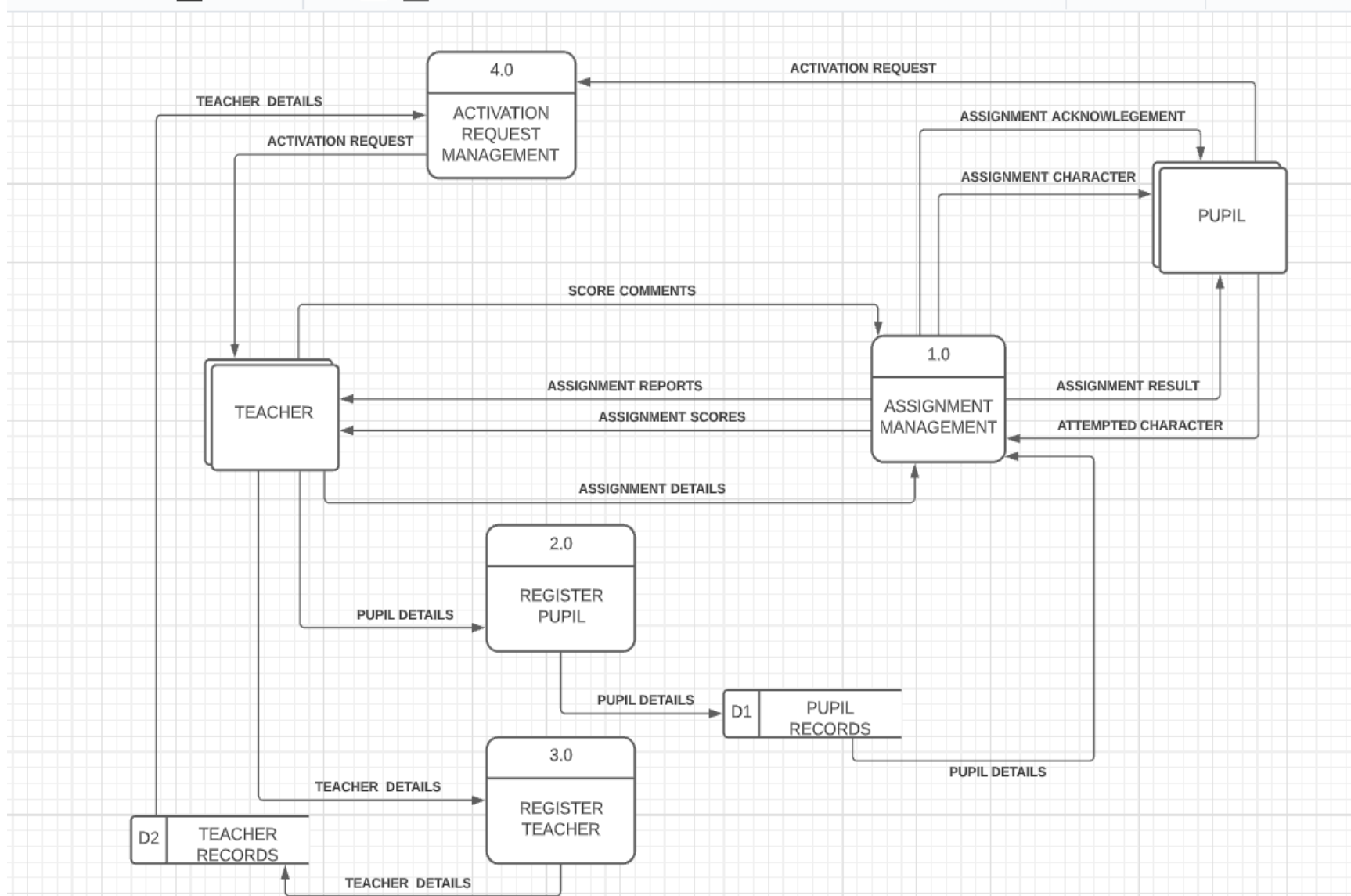


Figure 3. 4: *Level 0 Data Flow Diagram for the Kinder Care*

Figure 3.4 is a level 0 data flow diagram that is a further expansion of the context diagram. It shows the major processes that are within the system and these include, the Register pupil process, Register teacher process, Activation request management and the Assignment Management process.

The image also includes several data stores. These show how data will be stored in the system and what processes will access what data.

The Register pupils process takes in information about the pupils and produces organized information that is stored in the Pupil records data store.

The register teacher process takes in teacher details from the teacher entity and then produces teacher records that are organized so that they can be stored in the Teacher records data store.

The activation request management process gets an activation request from the pupil entity. It then accesses information from the Teacher records data store and then produces and then forwards the activation request to the right teacher.

The assignment management process gets assignment details from the teacher entity and then produces assignment characters for the student to attempt one by one. The assignment management process also gets attempted characters from the Pupil process and then grades them. It then produces an Assignment acknowledgement to the Pupil.

The Assignment management process should access information from the Pupil records data store so as to produce Assignment scores which are then sent to the teacher entity. The Assignment management process receives Score comments from the Teacher entity and produces reports for both the teacher entity and the Pupil entity.

NOTE: Processes contain the business logic, also called business rules, which transform the data and produce the required results. (Tiley RosenBlatt, 2016)

3.3 Design Rationale

We choose a 3-tier client-server diagram due to the following reasons;

- It supports horizontal scalability of the system to accommodate growing traffic, data and storage needs.
- It is also easy to maintain because there isn't a deep dependency of one tier on the other. This makes it easy to repair faults and bugs.
- The architecture makes the system easy to develop because each level of the system has specific deliverables which act as milestones during development rather than building a block program.
- The architecture is also a little complex compared to other architectures. We could have used 2-tier architecture but it is difficult to develop because it combines the server and database layers.

We also chose to use structured analysis and design since we are going to use the C programming language, and also structured PHP while developing the system.

It would have been hard if we had used Object oriented analysis yet we are going to write the code in procedural based programming languages.

4. DATA DESIGN

This section is made up of two sections i.e. data description and data design. Data description helps inform us of what data will be stored in the database by organizing the data into entity types that have relationships between them. Data design helps describe how the data will be stored in the database in form of tables.

4.1 Data Description

The system will have two sources of data, that is, data from the command line interface and also data from the web application interface. All this data will be stored in one central database.

The command line interface will be designed using C programming language while the web application will be designed using PHP. This means that we will have different variable declarations on all sides though the data will be on the same database. Below are tables showing the data in the two programs.

Some data elements are unique to the PHP program while the others are common to both the programs. The information about the teacher will be the information about the teachers. The variable declarations on both sides will be different though corresponding variables have to use the same element in the database.

Table with data for both the programs.

<u>Data elements</u>	<u>Description</u>
User code	Unique code that identifies a pupil
Pupil's first name	A pupil's first name

Pupil's last name	A pupil's first name
Phone number	A pupil's phone number
Start and end dates	The start and end dates of the assignments
AssignmentID	Uniquely identifies each assignment
Assignment name	The name of each assignment
Character name	The unique name of a character.
Character arrays	Multi-dimensional arrays to hold the character mapping positions
Marks	Marks scored by the students in a particular assignment
User name	Unique name to identify a teacher
Teacher's first name	A teacher's first name
Teacher's last name	A teacher's last name
Password	The teacher's password
Status	Shows whether a student is active or not.

Table 4. 1: Shows the data elements that will be in the system.

ENTITY RELATIONSHIP DIAGRAM FOR KINDERCARE SYSTEM

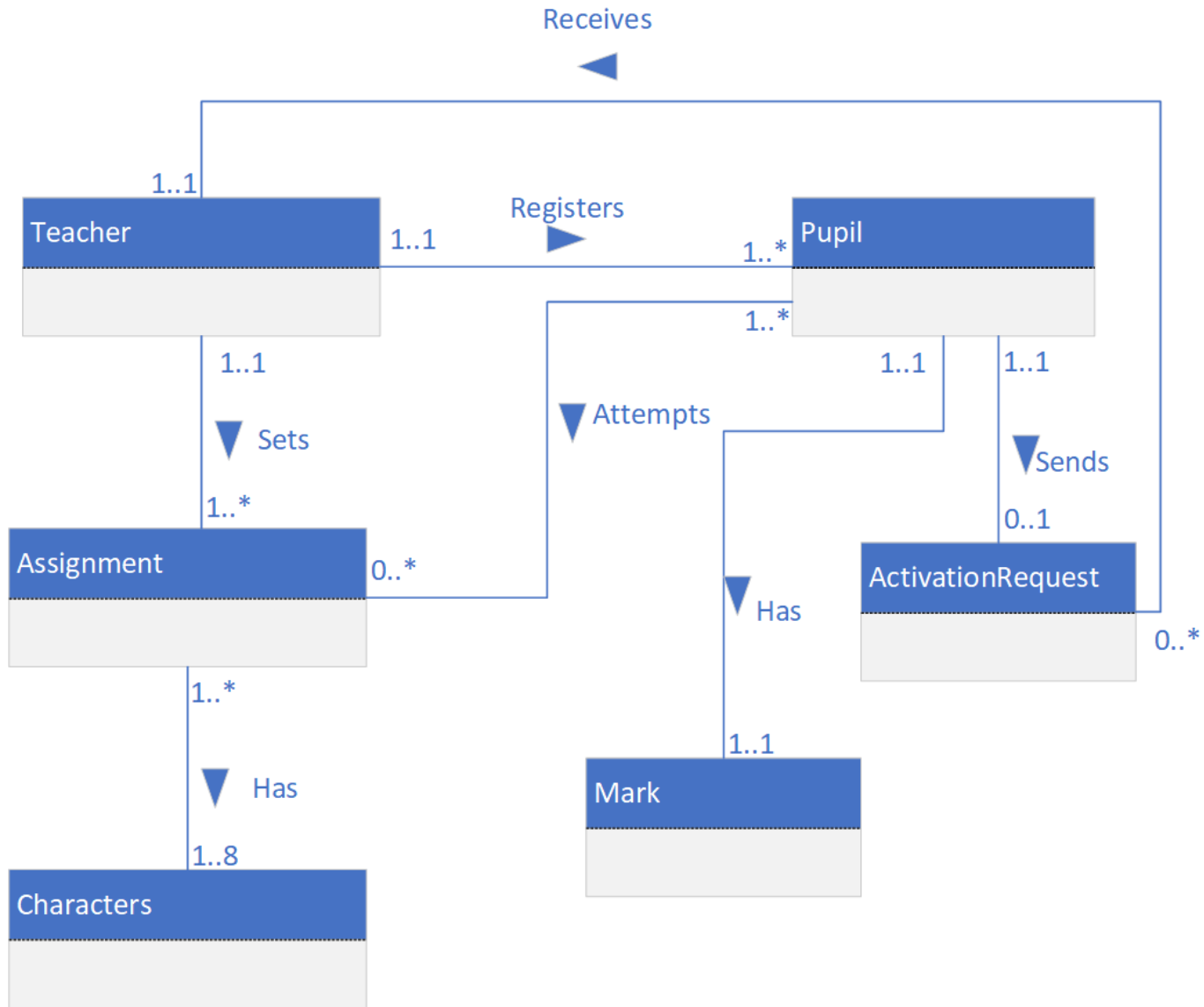


Figure 4. 1: The entity-relationship diagram for the database.

This is an entity relationship diagram showing how the database elements of the system interact. It shows the relationships between the major entity types that will be in the system.

An entity type is a group of objects with the same properties, which are identified by the enterprise as having an independent existence. (Thomas Conoly, Carolyn Begg)

The diagram has several relationships between the entity types and relationships which are represented by a line, an arrow and then a label for that relationship.

The diagram also has multiplicities such as 1..1, 1..*, 0..*, and so on which show how instances of an entity type relate to other entity types. For example, 1..8 means that an assignment can have up to eight characters.

4.2 Data Dictionary

The data dictionary shows the actual tables as they will be stored in the database. These tables are derived from the Entity relationship diagram above. They mapped from the ERD so as to be used to design a database that fulfills the information requirements of the KinderCare system.

Field Name	Data type	Length	Constraint	Description
userName	varchar	20	PRIMARY KEY	Uniquely identifies a teacher
firstName	varchar	15	NOT NULL	Identifies the teacher's first name
lastName	varchar	15	NOT NULL	Identifies the teacher's last name
passWord	varchar	20	NOT NULL	Identifies the teacher's password

Table 4. 2: Teacher Table 4.2 will store data about a teacher and will have fields as shown above.

Table 2: Pupil Table

Field Name	Data type	Length	Constraint	Description
userCode	varchar	20	PRIMARY KEY	The pupil's unique identifier
firstName	varchar	15	NOT NULL	The pupil's first name
lastName	varchar	15	NOT NULL	The pupil's last name
phoneNo	varchar	15	UNIQUE	The pupils telephone number
status	Boolean		NOT NULL	Determines whether a pupil is active or not.
userName	varchar	20	FOREIGN KEY	It is to reference the teacher table (i.e. show that there is a relationship between the two tables)
assignmentID	integer		FOREIGN KEY	It references the assignment table

Table 4. 3: Pupil Table 4.3 will store information about a pupil and will also include fields that enable it to communicate to other tables.

Field Name	Data type	Length	Constraint	Description
assignmentID	integer		AUTO INCREMENT	Uniquely identifies the assignment

assignmentName	varchar	30	NOT NULL	Identifies the name of the assignment
startTime	date/time	15	NOT NULL	Identifies when the assignment will start
endTime	date/time	15	NOT NULL	Identifies when the assignment will expire
userName	varchar	20	FOREIGN KEY	This references the teacher who set a particular assignment.

Table 4. 4: Assignment

Table 4.4 is to store information about the submitted assignments as show but it will also include some fields from other tables that will reference those tables.

Field Name	Data type	Length	Constraint	Description
requestNo	integer		AUTO INCREMENT	Uniquely identifies an activation request.
userName	varchar	20	FOREIGN KEY	References the teacher to whom the activation request is being sent.
userCode	varchar		FOREIGN KEY	References the pupil who is sending the activation request

Table 4. 5: Activation request.

Table 4.5 is to store information about activation requests sent by the pupils to the teachers. It also includes foreign keys from the teacher and pupil tables to ensure that the right activation request from a certain pupil is sent to the right teacher.

Character table

Field Name	Data type	Length	Constraint	Description
characterName	varchar	15	PRIMARY KEY	Uniquely identifies a letter
characterSequence	varchar		NOT NULL	Identifies the correct character sequence that will be used by the system while marking the pupil.

Table 4. 6: Character

Table 4.6 stores information about the characters that are to be put in assignments. It also includes fields that reference other tables.

Field Name	Data type	Length	Constraint	Description
userCode	varchar	20	PRIMARY KEY, FOREIGN KEY	References the pupil to whom the marks belong

assignmentID	integer		PRIMARY KEY, FOREIGN KEY	References the assignment to which the mark belongs
score	integer			Identifies the mark scored by a pupil.
comment	varchar	100		Identifies the comment a teacher attaches to the pupil's score

Table 4. 7: Marks

Table 4.7 is to store information about a student's scores.

Field Name	Data type	length	Constraint	Description
assignmentID	integer		PRIMARY KEY, FOREIGN KEY	References to a certain assignment.
characterName	varchar	15	PRIMARY KEY, FOREIGN KEY	References to a certain character.

Table 4. 8: Assignment Character

Table 4.8 is to provide a reference between the assignment and character tables.

5. COMPONENT DESIGN

This section shows the actual pseudocode behind all the functional components of the kinder care school application.

5.1 Teacher Registration

Here, we show how the system will create an account for a new teacher on their first login.

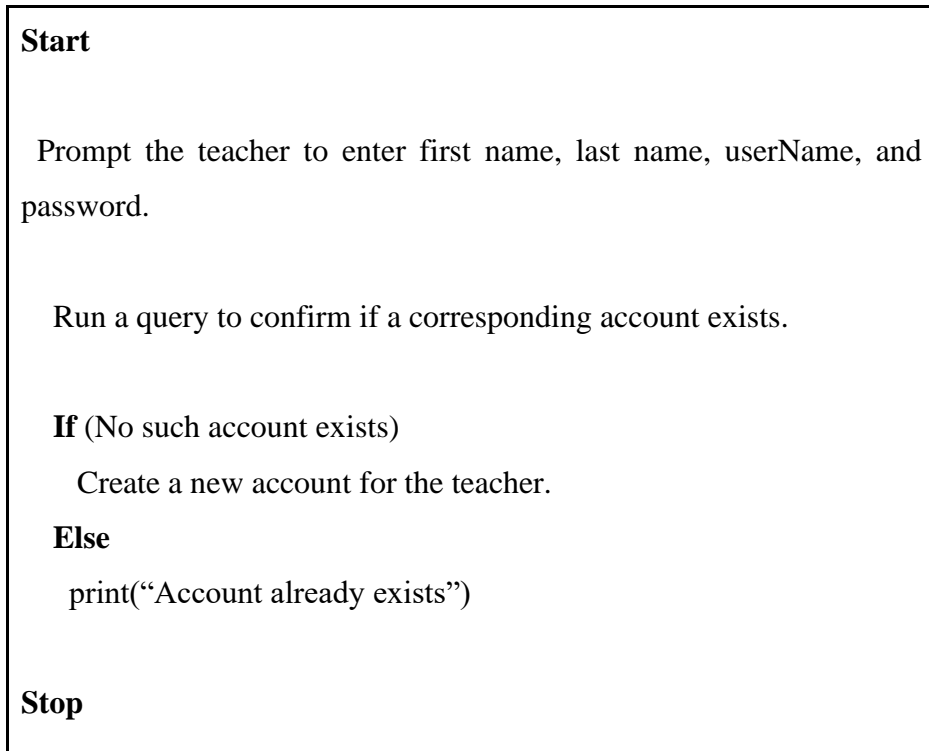


Figure 5. 1: *Pseudocode for teacher registration.*

5.2 Teacher Login

This pseudocode details the underlying steps executed by the system when a teacher is logging in.

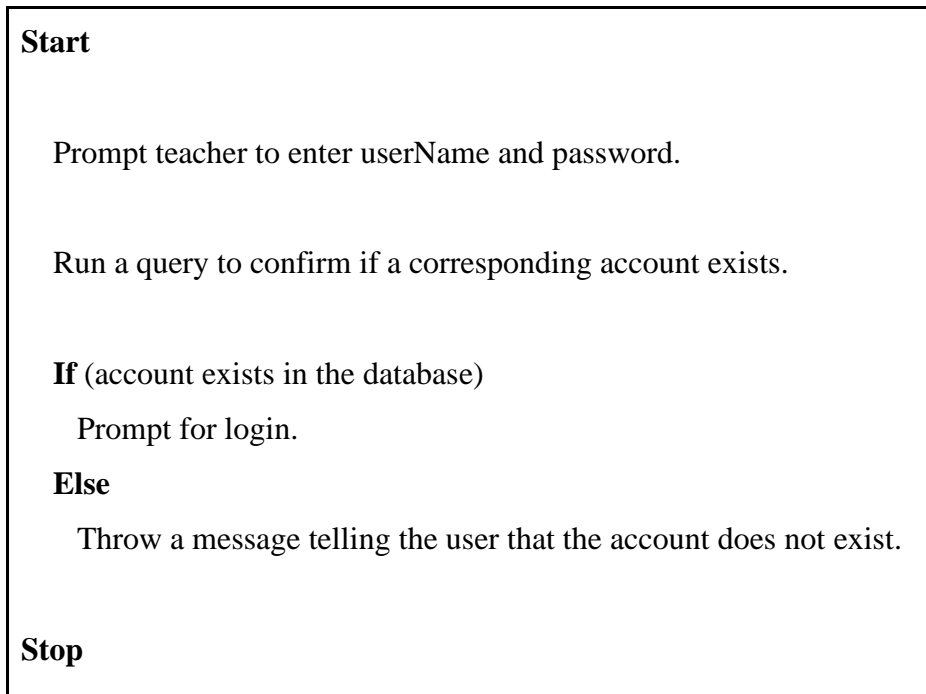


Figure 5. 2: Pseudocode for teacher login

5.3 Pupil Registration

This pseudocode shows the steps executed by the system in registering a pupil.

Start

Prompt a user to enter pupil's first name, last name, phone number and user code.

Run a query to the database to check if no such entry exists.

If (entry exists in the database)

 print ("This pupil is already registered")

Else

 Create a new database entry for the pupil.

 Activate pupil.

Stop

Figure 5. 3: Pseudocode for registering a pupil

5.4 Create Assignment

This pseudocode shows the steps to be followed by the system to help the teacher create an assignment.

```
Start
Enter number of characters(n) for the assignment.
while (n <= 8)
    Enter a character for the assignment.

    Enter start time for assignment.

    Enter end time for assignment.
    Run a query to determine whether the characters entered are in
the database.
    if (characters exist)
        Store assignment details in assignment records.
    else
        print ("Please enter a capital letter")

else
    print ("You can only enter 8 characters at most")

Stop
```

Figure 5. 4: Pseudocode for assignment creation.

5.5 Pupil login

This shows the steps that the system will follow before granting a pupil access to their accounts through the command line.

```
Start  
Prompt pupil to enter user code and phone number  
  
Check if the entered details correspond to any entry in the database  
  
If (entry exists)  
    Grant pupil access to the system and all privileges they need.  
else  
    print ("Enter the correct login details")  
Stop
```

Figure 5. 5: Pseudocode for pupil login

5.6 Attempt Assignment

This pseudocode shows the steps executed by the system when a pupil is attempting an assignment.

```
Start  
    If pupil activated  
    {  
        While (time >= start time && time < end time)  
        {  
            Call sends character function  
            Store the students attempt in an array  
            Capture time taken to attempt current character in an array  
character.  
            Total time = total time used so far
```



```

        If (character == last character)
            Break out of loop
        }
        Display total time of attempting the assignment.
        Print attempt acknowledgment.
        Print attempted characters.
    }
    Else
        Print “You are deactivated. Please request activation.”
    Stop

```

Figure 5. 6: Pseudocode for attempting assignment

5.6.1 Send character function

This is the pseudocode of the send character function.

```

Start
    sendCharacter (assignmentID, characterNo)
        Get assignment array using the using assignmentID
        return character (assignmentarray[characterNo])
Stop

```

Figure 5. 7: Pseudocode for teacher send character function

5.7 Activate Pupil

This pseudocode details the underlying steps executed by the system in activating a pupil.

```
Start
    Send a query to the database to set pupil status to active.
Stop
```

Figure 5. 8: Pseudocode for activating a pupil

5.8 Deactivate Pupil

This pseudocode details the underlying steps executed by the system in deactivating a pupil.

```
Start
    Send a query to the database to set pupil status to inactive.
Stop
```

Figure 5. 9: Pseudocode for pupil deactivation

6. HUMAN INTERFACE DESIGN

6.1 Overview of User Interface

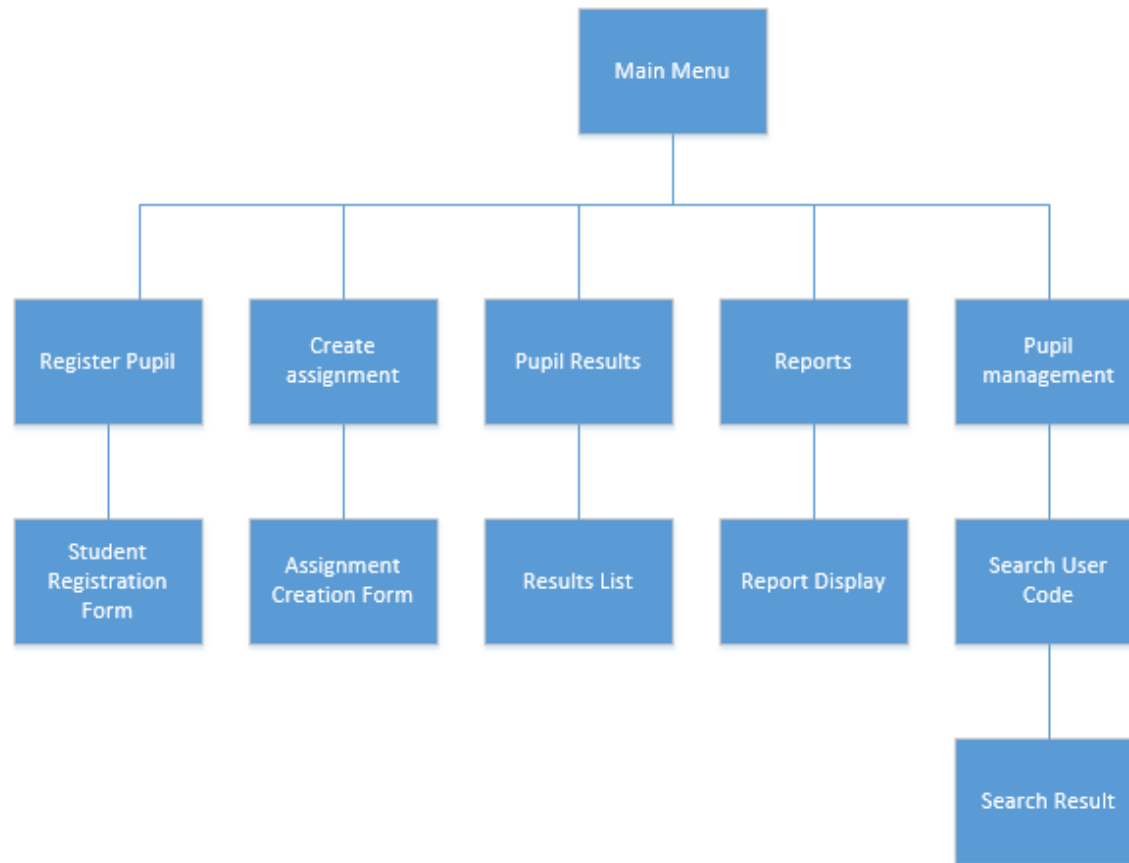
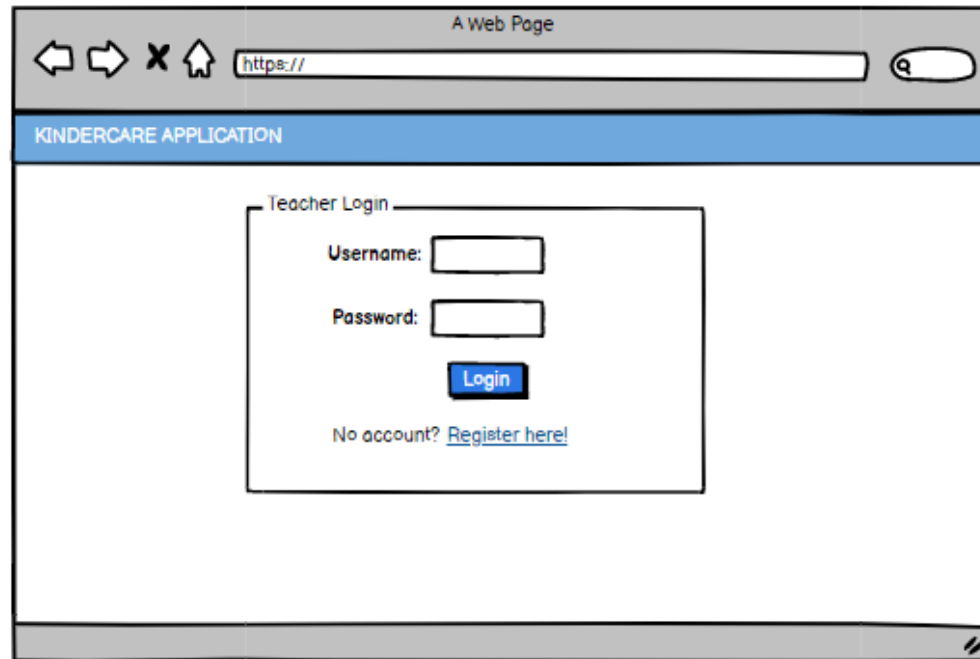


Figure 6. 1: Graphical representation of the user interface

The teacher selects options on the main menu. When he selects the register pupil option, He is directed to the pupil registration page where he can register the pupils who will be attempting his assignment. If he selects the create assignment, he is provided with an option to provide the characters for the assignment with the start and end dates. If he selects pupil results, he is provided with a list of students with their marks and provided with an option to submit a comment against each score. When he selects the reports, the reports on the pupils performance is displayed and lastly when he selects the pupil management, he is provided with a search field where he can enter the user code of the pupil and he can activate or deactivate that pupil.

6.2 Screen Images



The image shows a web browser window titled "A Web Page". The address bar contains "https://". Below the address bar is a blue header bar with the text "KINDERCARE APPLICATION". The main content area features a "Teacher Login" form. The form includes a "Username:" label followed by a text input field, a "Password:" label followed by a text input field, a blue "Login" button, and a link that says "No account? [Register here!](#)".

Figure 6. 2: Login page for the KinderCare application

A screenshot of a web browser window titled "A Web Page". The address bar shows "https://". The page has a blue header bar with the text "KINDERCARE APPLICATION". Below the header, there is a "Teacher signUp" form. The form contains four input fields: "First Name:", "Last Name:", "Username:", and "Password:". Below these fields is a blue "Register" button. At the bottom of the form, there is a link that says "Already have an account? [Login!](#)".

Figure 6. 3: Signup page: Displayed on clicking the register here link on the login page.

The image shows a web browser window titled "A Web Page" with a search bar containing "https://". The main content area is titled "KINDERCARE APPLICATION" and includes a "Logout" link. A sidebar on the left lists navigation options: "Register Pupil", "Create assignment", "Pupil Results", "Reports", and "Pupil management". The main content area displays the "Pupil Registration Form" with four input fields: "First Name:", "Last Name:", "Phone Number:", and "User Code:". A blue "Register" button is located below the "User Code:" field.

Browser: A Web Page
Address: https://

KINDERCARE APPLICATION [Logout](#)

Register Pupil
Create assignment
Pupil Results
Reports
Pupil management

Pupil Registration Form

First Name:
Last Name:
Phone Number:
User Code:
[Register](#)

Figure 6. 4: Register Pupil page: Displayed on teacher logging in to the system.

A Web Page

https://

KINDERCARE APPLICATION [Logout](#)

[Register Pupil](#)
[Create assignment](#)
[Pupil Results](#)
[Reports](#)
[Pupil management](#)

CREATE ASSIGNMENT

Choose characters for the assignment

Character 1: Character 2:

Character 3: Character 4:

Character 5: Character 6:

Character 7: Character 8:

Set assignment duration

Start time: / / End time: / /

[Post Assignment](#)

Figure 6. 5: Create assignment page: displayed on clicking create assignment link on the side menu.

A Web Page

https://

KINDERCARE APPLICATION [Logout](#)

- [Register Pupil](#)
- [Create assignment](#)
- [Pupil Results](#)
- [Reports](#)
- [Pupil management](#)

Student Name	Score	Time taken	Comments
student 1	8	21	<input type="text"/>
student 2	15	35	<input type="text"/>
student 3	11	17	<input type="text"/>

[Submit Comments](#)

Figure 6. 6: Pupil Results page: displayed on clicking the pupil result link on the side menu.

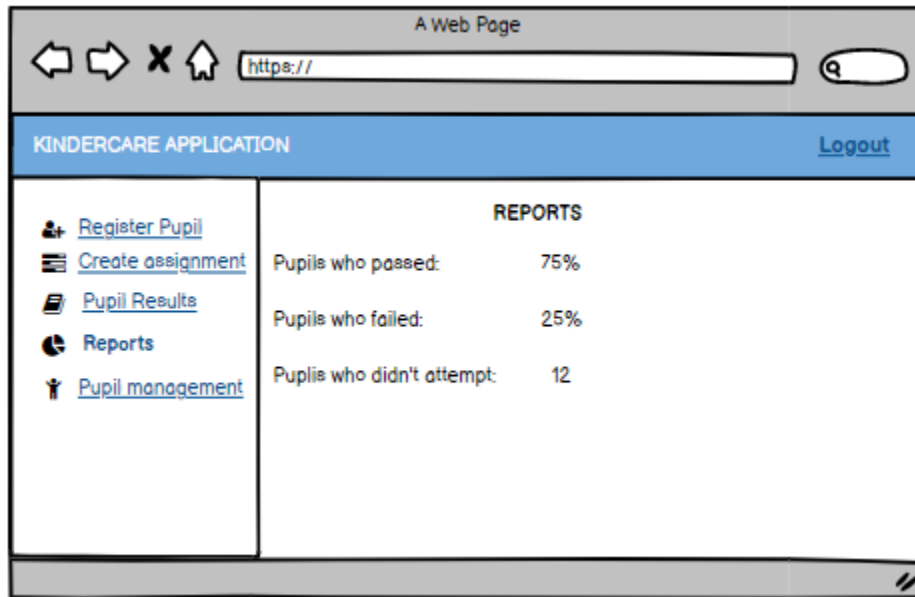


Figure 6. 7: Reports; displayed on clicking the Reports link on the side menu.

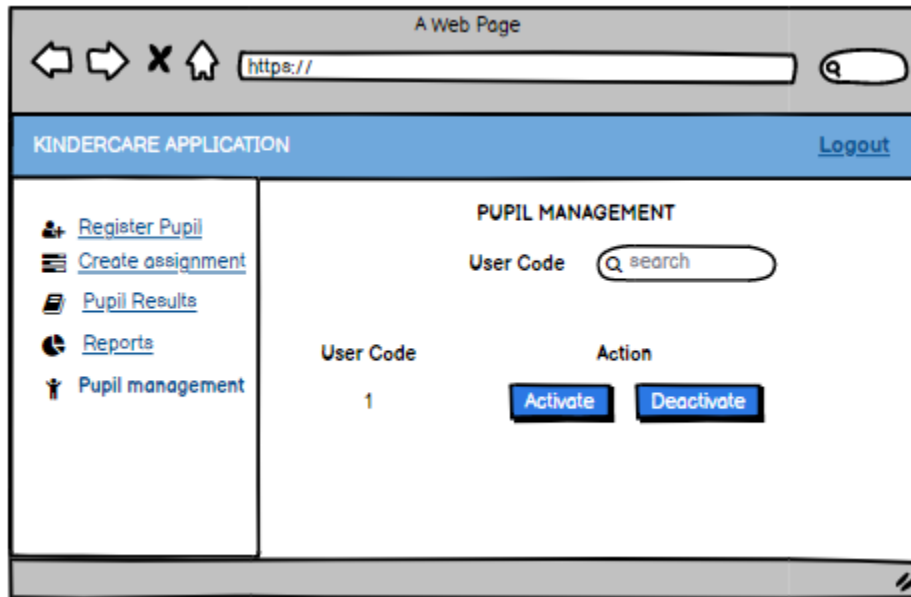


Figure 6. 8: Pupil Management page: displayed on clicking the pupil management link on the side menu.

```

prompt>Enter User Code: P10
prompt>Enter Phone Number: 0753451233

*****KINDERCARE ASSIGNMENT SYSTEM*****
**System Commands*****Action*****
Viewall                      Displays assignment number and date
Checkstatus                  Displays status report and assignments
Viewassignment assignmentid   See details of specified assignment
Checkdates datefrom dateto   See if assignment is within range specified
RequestActivation            Request for activation from the teacher

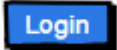






prompt>Enter command:

```

Figure 6. 9: Command line interface for the KinderCare Application

A pupil uses the command line interface by typing commands after “Enter command:”.

6.3 Screen Objects and Actions

Object	Object Name	Action
	Login button	Submits the login details of the teacher to the server for authorization to enable him/her access to the system.
	Register button	Submits details of new -teachers to the database to enable creation of teacher accounts for the system.
MENU  Register Pupil  Create assignment  Pupil Results  Reports  Pupil management	Side menu	List of links for navigating the different pages in the web application.








	Text input field	Captures the text data entered by the teacher.
	Search field	Searches for user code of pupil from database.
	Activate and Deactivate buttons	Teacher clicks to activate and deactivate the pupil.
	Logout button	Clears out the current session and returns the teacher to the Login page.
	Date input field	Sets the start and stop time for the assignment.
	Submit comments button	Submits the Score comments from the teacher to the database.
	Post assignment button	Submits a new assignment to the database.

Table 6. 1: Screen objects and their actions.