**PROJECT TEAM NAME: GROUP -34**

PROJECT MEMBERS

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

**PROJECT TITLE: KINDERCARE CHARACTER DRAW:**

**DATE: 21st January, 2022**

Table of Content

[**1.** INTRODUCTION iv](#_Toc61356659)

[1.1 Purpose iv](#_Toc61356660)

[1.2 Scope iv](#_Toc61356661)

[Goals/objectives of the project iv](#_Toc61356662)

[Benefits of the project iv](#_Toc61356663)

[1.3Overview iv](#_Toc61356664)

[1.4 Reference Material v](#_Toc61356665)

[1.5 Definitions and Acronyms v](#_Toc61356666)

[**2.** SYSTEM OVERVIEWS v](#_Toc61356667)

[**3.** SYSTEM ARCHITECTURE viii](#_Toc61356668)

[**3.1** ARCHITECTURAL DESIGN viii](#_Toc61356669)

[**3.2** DECOMPOSITION DESCRIPTION ix](#_Toc61356670)

[**3.3** DESIGN RATIONALE xii](#_Toc61356671)

[REASONS FOR USING THE LAYERED ARCHITECTURE IN Figure 1 xii](#_Toc61356672)

[DISADVANTAGES OF USING THE LAYERED ARCHITECTURE IN Figure 1 xii](#_Toc61356673)

[OTHER ARCHITECTURE THAT WHERE FOREGONE xii](#_Toc61356674)

[**4.** DATA DESIGN xiv](#_Toc61356675)

[**4.1** Data Description xiv](#_Toc61356676)

[**4.2** Data Dictionary xv](#_Toc61356677)

[**5.** COMPONENT DESIGN xvii](#_Toc61356678)

[**6.** HUMAN INTERFACE DESIGN xix](#_Toc61356679)

[**6.1** Overview of User Interface xix](#_Toc61356680)

[**6.2** Screen Images xx](#_Toc61356681)

Appendices

Table of Figures

Table of Figures

[Figure 1 layered architecture](../../../../C:/Users/Josha/Desktop/work%20desk/GROUP%2011%20software%20design%20document.docx#_Toc61356683) viii

[Figure 2 level 0 Data Flow DiagramError: Reference source not found](#_Toc61356684)

[Figure 3 Level 1 Data Flow Diagram](../../../../C:/Users/Josha/Desktop/work%20desk/GROUP%2011%20software%20design%20document.docx#_Toc61356685) xi

[Figure 4 KinderCare Character Teacher Login and registrationError: Reference source not found](#_Toc61356686)

[Figure 5 web application login and register pageError: Reference source not found](#_Toc61356687)

[Figure 6 web application home pageError: Reference source not found](#_Toc61356688)

[Figure 7](../../../../C:/Users/Josha/Desktop/work%20desk/GROUP%2011%20software%20design%20document.docx#_Toc61356689) KinderCare teacher registration of students xxi

figure 8 Teachers Assignment submission text Area……………………………..……….……xxii

figure 9 teacher’s view of Pupils table with activate and Deactivate buttons………………...xxiii

figure 10 Commandline Representation of students Request to Reactive……………………...xxiv

figure 11 Commandline Representation generated Automark…………………………….……xxv

figure 12 Screen Representation of Drawn Characters………………………………..……….xxvi

# **1.** INTRODUCTION

## 1.1 Purpose

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

## 1.2 Scope

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

Goals/objectives

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

**Overview**

**4.2 Data Dictionary**

Pupil table

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

Assignment table

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Description** |
| AssignmentId | int | Uniquely identifies an assignment and is automatically generated. |
| Start\_time | date | Indicates the time when an assignment commences |
| End\_time | Date | Indicates the time when an assignment ends |
| scores | int | Mark scored by the pupil for the assignment submitted |
| comment | Varchar(50) | Given by the teacher against the pupil’s assignment |
| UserCode | int | Fk referencing the pupil’s table |

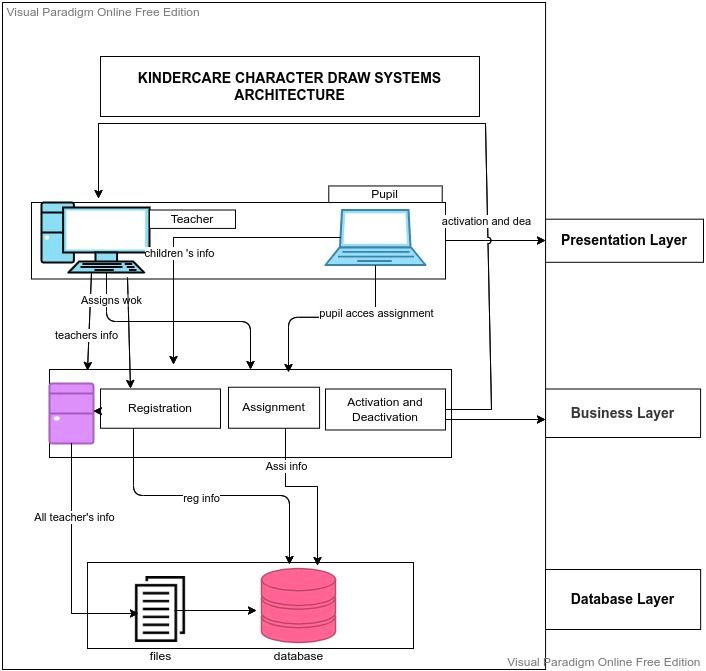
Teacher

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Description** |
| TeacherID | Int | Unique,automatically identifies the teacher |
|  |  |  |
|  |  |  |

# **3.** SYSTEM ARCHITECTURE

The system registers and keeps records of all KinderCare Character Draw Pupils and their teachers. The pupils are registered through the Web interface and the teacher is able to give assignments to pupils through through the same interface.The system also allows Pupils to attempt the Assignments given by the teacher as long as they are still with in the required time and they are registered. Teachers are allowed by the system to deactivate the pupils at a given time and reactivate them.

3.1 ARCHITECTURAL DESIGN

figure1

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

**The presentation layer**: This layer is responsible for handling all user interface and browser communication logic. We have the client computer used by KinderCare Teachers to register new pupils with the aid of a client web interface that sends the data to the socket c program. The information entered by the teachers about the pupils is viewed by the respective pupils who are registered, through socket c program that sends the data to the socket server c program that is located in the business and persistence layer.

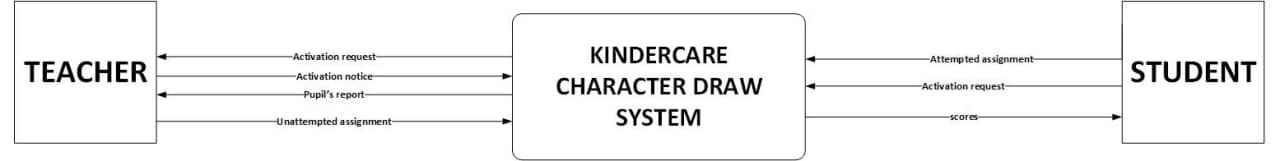
We again stress the point of command line interface which basically provides all the functionality necessary for the the pupils to interract with the system by viewing their registration status, Assignments to attempt, sending claims to the teacher to re-activte a deactvated pupil.

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

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

**KINDERCARE CONTEXT DIAGRAM**

This is our context diagram



**KNDERCARE DATA FLOW DIAGRAM**

