SOFTWARE DESIGN DOCUMENT FOR

KinderCare Character Draw

Github link: https://github.com/Babwat/G-10

G-10

Supervisor : BRIAN MUCHAKE

|  |  |  |
| --- | --- | --- |
| Name | Registration number | Student number |
| BABIRYE PATRICIA NAKYEJWE | 20/U/2017/EVE | 2000702017 |
| NANTANDA JAMILAH | 20/U/7807/PS | 2000707807 |
| DONALD KAIJUKA | 13/U/5658/PS | 1300703528 |
| AGABA ABEL | 19/U/0215 | 1900700215 |

Table of Contents

[1. INTRODUCTION 1](#_Toc93862070)

[1.1 Purpose 1](#_Toc93862071)

[1.2 Scope 1](#_Toc93862072)

[1.3 Overview 2](#_Toc93862073)

[1.4 Reference Material 2](#_Toc93862074)

[1.5 Definitions and Acronyms 2](#_Toc93862075)

[2. SYSTEM OVERVIEW 3](#_Toc93862076)

[3. SYSTEM ARCHITECTURE 4](#_Toc93862077)

[3.1 Architectural Design 4](#_Toc93862078)

[3.2 Decomposition Description 5](#_Toc93862079)

[3.3 Design Rationale 7](#_Toc93862080)

[4. DATA DESIGN 9](#_Toc93862081)

[4.1 Data Description 9](#_Toc93862082)

[4.2 Data Dictionary 9](#_Toc93862083)

[5. COMPONENT DESIGN 12](#_Toc93862084)

[6. HUMAN INTERFACE DESIGN 17](#_Toc93862085)

[6.1 Overview of User Interface 17](#_Toc93862086)

[6.2 Screen Images 18](#_Toc93862087)

TABLE OF FIGURES

[Figure 6. 1 Navigation flow of the system 17](#_Toc93862154)

[Figure 6. 2 Pupils page 18](#_Toc93862155)

[Figure 6. 3 Assignment Page 18](#_Toc93862156)

[Figure 6. 4 Add Pupil page 19](#_Toc93862157)

[Figure 6. 5 Adding Time page 19](#_Toc93862158)

[Figure 6. 6 Results window 20](#_Toc93862159)

[Figure 6. 7 Reports page 20](#_Toc93862160)

[Figure 6. 8 Command line Interface 21](#_Toc93862161)

[Figure 6. 9 Screen Objects and Actions 21](#_Toc93862162)

LIST OF TABLES

[Table 4. 1 Teacher 9](#_Toc93862359)

[Table 4. 2 Pupil 10](#_Toc93862360)

[Table 4. 3 Assignment 10](#_Toc93862361)

[Table 4. 4 Status Report 11](#_Toc93862362)

# INTRODUCTION

## 1.1 Purpose

This software design document describes the architecture and system design of character draw, software that helps lower school children remember and create the different shapes of characters, ranging from A to Z (All in capital).

## 1.2 Scope

The software is going to be used to ease teaching of the alphabet to lower level students at KinderCare. It is going to run on two interfaces, the command line and web interface. The web interface is going to be used by a teacher at KinderCare to register pupils in his/her class, upload assignments and monitor the pupils’ progress by viewing their status reports. The command line interface is going to be used by a pupil to log into their account, attempt an assignment, check their status report and send a request for activation to their teacher.

Goal;

To build a learning system for KinderCare that is both cheap and efficient.

Objectives;

* Efficient monitoring of pupils’ progress by a teacher by being able to view each of their status reports and commenting as necessary.
* To ensure the learning system meets its expectations.
* To use tools that are both efficient and cost effective when building the system software.

Benefits of the project;

* The learner's will be monitored and individual success and progress followed up easily.
* It will ease the process of registering pupils.
* Teachers will have access to necessary reports and make required decisions.

## 1.3 Overview

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

Chapter 1: describes the product scope, the purpose of this document, the reference material and abbreviations used with their meanings.

Chapter 2: describes the system overview i.e. the general description of the functionality, context and design of the KinderCare Learning System.

Chapter 3: describes the system architecture, which comprises the system architecture design, the decomposition of the system and justification for the choice of the design.

Chapter 4: describes the data design of the system, which shows how the information domain of the system is transformed into data structures.

Chapter 5: describes the component design of the system. It gives a functional description of each component in detail.

Chapter 6: describes the human interface design of the system, the way users interact with the system and provides screen images for each of the components.

## 1.4 Reference Material

# Bibliography

Tilley, Scott and Rosenblatt, Harry J. *Systems analysis and design.* Cengage Learning, 2016.

Liu, Jinfeng, David Muñoz de la Peña, Benjamin J. Ohran, Panagiotis D. Christofides, and James F. Davis. "A two-tier architecture for networked process control." *Chemical Engineering Science* 63, no. 22 (2008): 5394-5409.

## 1.5 Definitions and Acronyms

KCCD - KinderCare Character Draw

DFD – Data Flow Diagram

# SYSTEM OVERVIEW

The KCCD system shall be designed to help lower school children to remember and create the different shapes of characters ranging from A to Z and improve the efficiency in learning by providing teachers and pupils registration, Administration of assignments to pupils, activation and deactivation of pupils, provision of a platform for pupils to attempt assignments, automatic grading of assignments and provision of reports for different pupils.

It shall comprise of three modules which include: the web interface, command line interface and the assignment check. The KCCD is made up of two applications, that is the web interface and command line program. The two applications communicate to each other via the database.

The Teacher via the web application registers the pupils and teachers details after which he/she sets the assignment which the pupil attempts and the system automatically generates marks to the assignment. The teacher then comments on the marks awarded on each pupil’s attempt. The teacher can then view the reports of different individuals.

The pupil via the command line application logs in and consequently attempts the assignment after which he/she submits the assignment to the system. The system then automatically generates marks for the assignment submitted.

The details such as how many attempted, average score, percentage missed, percentage attempted, time taken on each character and total time taken for the assignment are calculated, generated and recorded by the system which stores them in the Assignment file. The assignments and their details are stored and read on the database. The marks awarded for the assignments and time recorded is stored on the database.

This information stored on the database can be viewed both on the Web interface through the reports and on the command line interface via the checkstatus. The pupil's and teachers' details, assignments, and comments on the assignments are also stored on the database server.

# 3. SYSTEM ARCHITECTURE

## 3.1 Architectural Design

The system uses a 3-Tier client/server model which consists of two client machines, that is to say, the teacher, the pupil, the applications components and a database server.

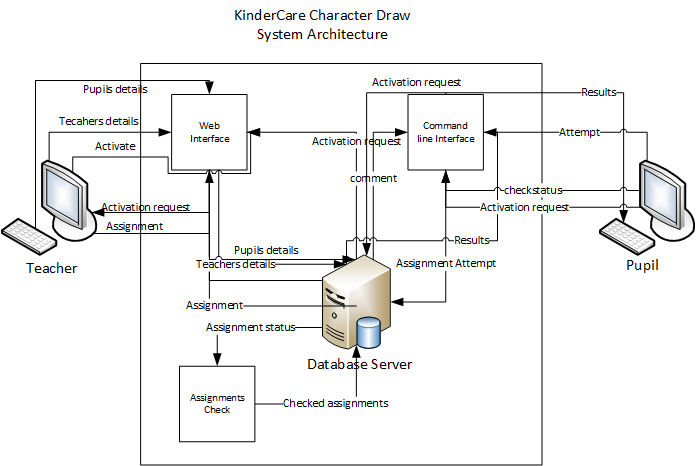


Figure 3. 1 System Architecture for KCCD

Web interface

This will provide a web application where the teacher registers the pupils details and gives them assignments which will be stored on the database. In addition, the interface will be used by the teacher to deactivate a student at any given time and also comment on the pupil's score.

Command line interface

This will be used by the pupil to login, get assignments, submit assignments, amount taken to complete the assignment, get acknowledgement of submission and get instant result scores. Furthermore, a pupil once deactivated can request for activation and view the comments from their teacher on their score on their next login.

Assignments Check

This checks if a pupil attempted the given assignments, the number of attempted assignments, the average score, stores the date from and date to of the assignment and stores them in the database.

## 3.2 Decomposition Description

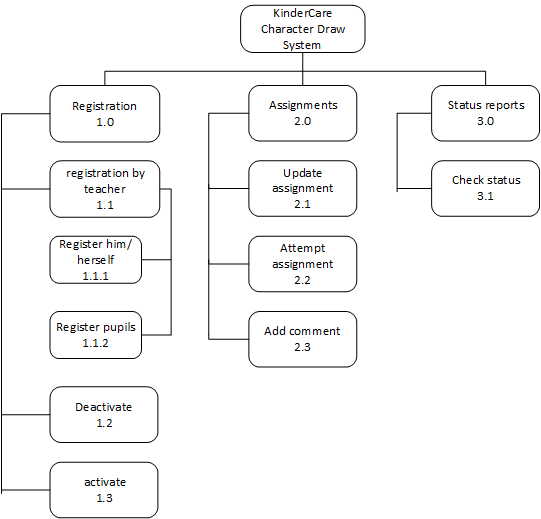


Figure 3. 2 Functional decomposition diagram for KCCD

Functional decomposition description

Process 1.0. Registration. The Teacher enters information details of both the pupil and themselves if they are not registered in the system.

Process 1.1. Registration by teacher. The teacher enters the relevant information.

Process 1.1.1. Register him/herself. The teacher who is not in the system enters their information details. These include first name, last name, username and phone number.

Process 1.1.2. Register pupil. The teacher then enters the information details of the pupil. These include the first name, last name, User code and the Phone number of the pupil.

Process 1.2. Deactivate. The teacher can deactivate a pupil who is inactive or taken long before attempting an assignment.

Process 1.3. Activate. The teacher can activate or reactivate a pupil that has been deactivated if they send an activation request.

Process 2.0. Assignments. The teacher then sets assignments for the pupils.

Process 2.1. Update assignment. The teacher can update or edit an assignment that they set.

Process 2.2. Attempt assignment. The pupil can then attempt the available assignment after they have logged in and are activated.

Process 2.3. Add comment. The teacher attaches a comment to the pupils scores.

Process 3.0. Status Reports. These are generated by the system which may include the marks, time taken for the assignment, total time taken, the comments.

Process 3.1. Check status. The pupil then requests to check on their status that is the percentage marks, time taken, comments.

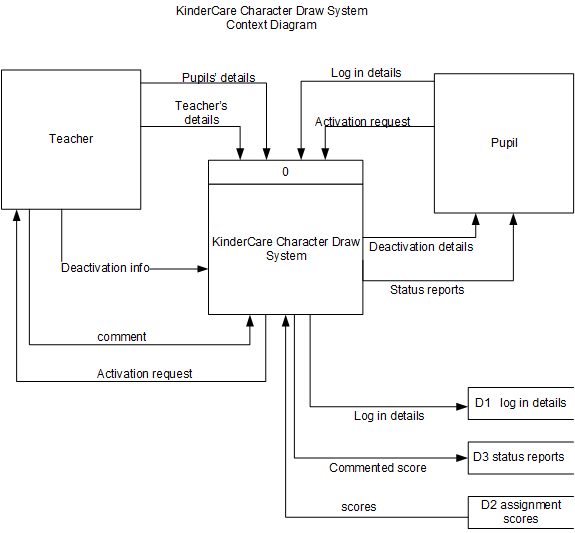


Figure 3. 3 Context diagram for KCCD

Context diagram description

A teacher registers himself, pupils' information, sets an assignment and comments on pupils’ scores. He also deactivates a child or re-activates them upon request. A pupil logs in, can see the assignment details, attempt it and also view a status report or send an activation request to the teacher.

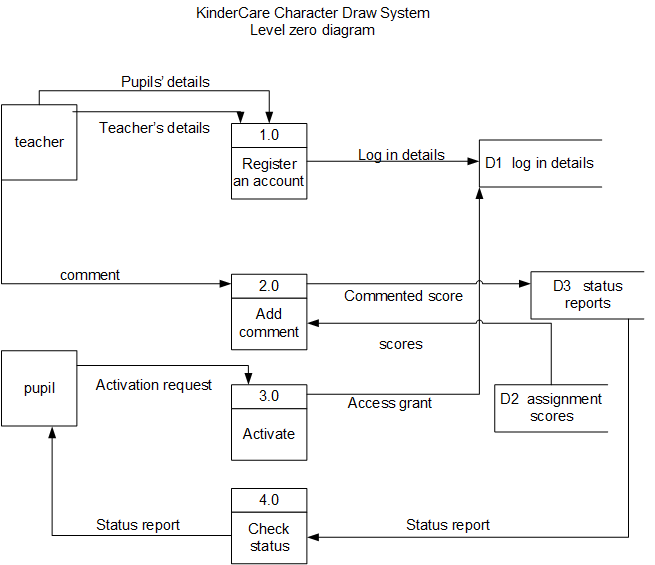


Figure 3. 4 Data Flow Diagram for KCCD

DFD Description

Process 1. Register an account. The Teacher enters the information details of the pupil. These include first name, last name, User code and Phone Number. The teacher also registers his/her details in order to be able to use the system. These are stored in the Login details.

Process 2. Add Comment. After the pupil attempts an assignment, the system automatically awards marks which are stored in the assignment scores data store. The teacher then attaches a comment to the assignment scores of different pupils. The comment is stored in the status reports.

Process 3. Activate. The pupil sends an activation request to the teacher in case they have been deactivated in order to qualify to attempt the assignment.

Process 4. Check Status. Upon completion of the assignment, the pupil requests for a status report which highlights the performance of the pupil among other details.

## https://lh4.googleusercontent.com/0ysY5Dllvr1s8nOzxWzv-Sb8FOHxwPVyFXsl5ZPUTDelIgFkQfjx0oCEiH1l41ydWvZwJo1sCZ-b2tUSRnWWJnkjXFlDnssX8AFDSQffdET96vuXPLQmPYSy7OOYYg

Figure 3. 5 Entity Relationship Diagram for KCCD.

## 3.3 Design Rationale

The purpose of this architecture in 3.1 is to allow data exchange and divide the application processing across the components whereby the critical functions are registering students, updating assignments, checking student progress, deactivation and activation of students, attempting of assignment within the specified time before expiration, submitting of assignments and getting of status report.

The 3-tier increases the performance of the system and helps with scalability plus eliminating many kinds of problems with confusion which can be caused by multi-user access in a two-tier architecture though the advanced complexity of the three-tier architecture means more cost and effort.

# 4. DATA DESIGN

## 4.1 Data Description

The KCCD system uses data of the teachers and pupils. The information of the teachers and pupils registered is recorded into the system by the teacher using a web application. The assignment details and information as well is set up by the teacher using the web interface. The assignment attempt by the pupil is and time taken to attempt is recorded into the system using the command-line interface.

## 4.2 Data Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Constraint** | **Description** |
| Username | varchar | Not null | This is the name that a teacher uses when creating an account. |
| Teacher ID | integer | Primary key | This is an ID number that uniquely identifies the teacher. |

Table 4. 1 Teacher

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Constraint** | **Description** |
| Usercode | Integer | Primary key | This code uniquely identifies a pupil on their account. |
| First name | varchar | Not null | This is the first name of the pupil. |
| Last name | varchar | Not null | This is the last name of the pupil. |
| Phone number | Long integer | Not null | This is the contact detail of the pupil. |
| Teacher ID | Integer | Foreign key | This references the teacher table in the pupil table. |

Table 4. 2 Pupil

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Data type | Constraint | Description |
| Assignment ID | integer | Primary key | This number uniquely identifies the assignment given. |
| Date\_of\_assignment | date | Not null | This describes a date when the assignment is open for attempting. |
| Time | float | Not null | This is the time frame for which the assignment is open for attempting. |
| Score | integer | Not null | This is a mark got by a pupil after attempting the assignment. |
| Teacher ID | integer | Foreign key | This references the teacher table in the assignment table. |
| Usercode | integer | Foreign key | This references the pupil table in the assignment table. |

Table 4. 3 Assignment

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Data type | Constraint | Description |
| Number attempted | integer | Not null | This is the number of assignments a pupil has attempted. |
| Average score | float | Not null | This is a ratio of total scores to number of assignments attempted. |
| Percentage missed | float | Not null | This is a ratio of assignments missed to assignments given multiplied by a hundred. |
| Percentage attempted | float | Not null | This is a ratio of assignments attempted to assignments given multiplied by a hundred. |
| comment | varchar | Not null | This is a comment given by the teacher against a pupil’s score. |
| Usercode | integer | Foreign key | This references the pupil table in the status report table. |
| Teacher ID | integer | Foreign key | This references the teacher table in the status report table. |

Table 4. 4 Status Report

# 5. COMPONENT DESIGN

Represents the algorithm at a level of detail that can be reviewed for quality of the project.

|  |
| --- |
| Start:  Enter  Teacher  Registers teacher’s details and creates an account on the system.  Teacher username, teacher ID  If no information on teacher is added,  Create account details on the database.  Else  Keep editing |

Figure 5. 1 Register Teachers details algorithm

Register pupils’ details on the database

This shows how pupils’ are registered onto the system by the teacher

|  |
| --- |
| Start:  Enter :  Teacher  Registers students details  Pupil’s firstname, lastname, phone number and usercode auto generated by the system.  If no more information on pupil is added,  Add pupil’s details on the database  Else  Keep editing details. |

Figure 5. 2 Register Pupils details algorithms

Uploading assignments

This shows how teachers upload assignments on the system

|  |
| --- |
| Start:  Teacher logins to the system  Adds a new assignment  If current number of assignments <= 8  Add new assignment  Else  Edit current assignments  Submit  Stop. |

Figure 5. 3 Updating Assignments algorithms

Attempting assignments

This shows how a pupil attempts assignments.

|  |
| --- |
| Start:  pupil login through the command line interface and enter commands.  If assignment is still valid  If all characters are attempted  Submit assignment.  Else if  Keep editing assignments.  Else teacher  Adds a comment on previous attempted assignment scores.  Stop . |

Figure 5. 4 Attempting Assignment Algorithm

Status report

This shows the report details of a pupil

|  |
| --- |
| Start:  Enter  Pupil  Pupil writes command  If view all  Displays assignment number and date, showing if attempted or not; check status  Else if check status  Displays the status report of the pupil summarizing all assignments, how many attempted, average score, percentage missed and attempted.  Else if view assignment    Enter,  assignment id  To see details of a specified assignment.  Else if check dates date from date to  Shows if there is an assignment within a specified date range  Else request activation.  Used for a pupil to request teacher to activate him or her.  Stop. |

Figure 5. 5 Status Report Algorithm

# 6. HUMAN INTERFACE DESIGN

## 6.1 Overview of User Interface

Navigation flow of the web system

|  |
| --- |
|  |

Figure 6. 1 Navigation flow of the system

**Users**

* Teacher
* Pupils

**Functionality**

The teacher will log into the web interface and navigate using the dashboard to register pupils, submit assignments, view scores, add comments and view reports.

The pupils will log into a command line interface and use commands to view details of assignments and attempt them as well as request activation from the teacher if deactivated.

## 6.2 Screen Images

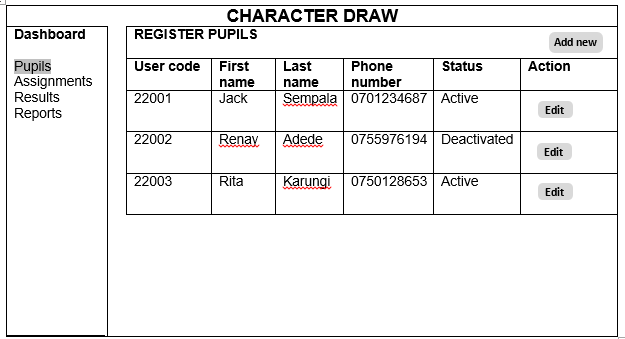
****

Figure 6. 2 Pupils page

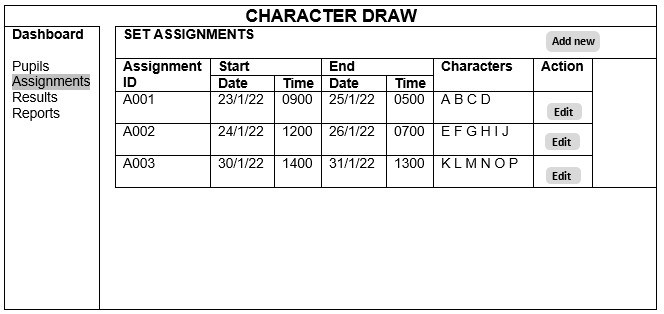
****

Figure 6. 3 Assignment Page

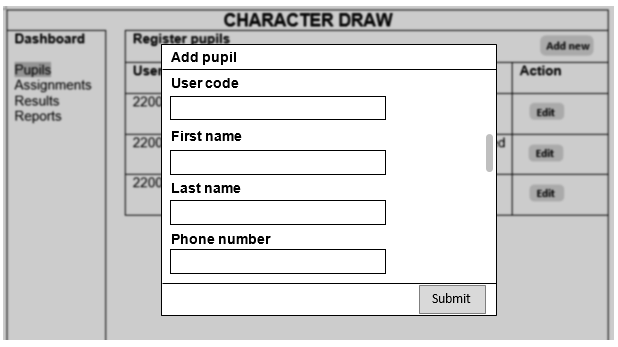
****

Figure 6. 4 Add Pupil page

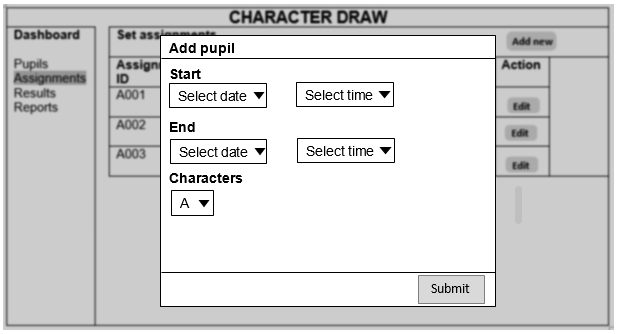
****

Figure 6. 5 Adding Time page

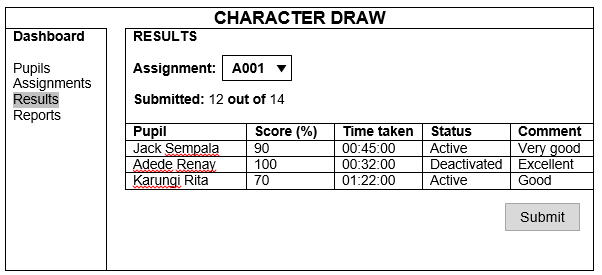
****

Figure 6. 6 Results window

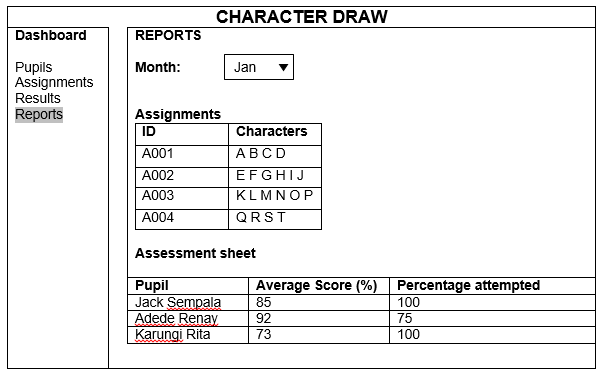
****

Figure 6. 7 Reports page

**Command line interface**

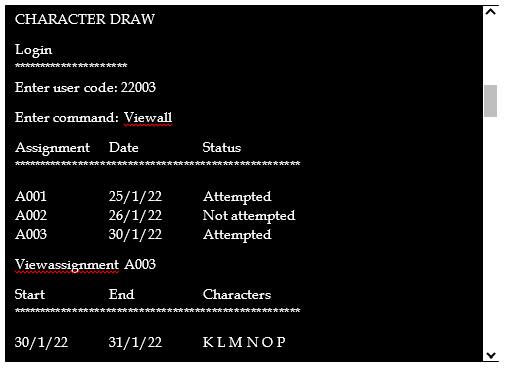
****

Figure 6. 8 Command line Interface

**6.3 Screen Objects and Actions**

|  |  |  |
| --- | --- | --- |
| **Object** | **Name** | **Action** |
|  | Select time | Allows user to select an option |
|  | Button | Submits form data. Displays dialog box |
|  | Input field | Used to enter data |

Figure 6. 9 Screen Objects and Actions

* The Select time Object. This allows a user to select a time when or for how long an assignment is due.
* The Add new Button. This allows a teacher to register a new pupil into the system.
* The Edit Button. This allows a user to edit or update any entry.
* The Input field. This allows a user to enter any information into the system.