DESIGN DOCUMENTATION FOR A CHARACTER DRAW ASSESSMENT TOOL FOR KINDERCARE

Github Link

http://github.com/jordankiggundu/reccess-group-work

Supervisor: Brian Muchake

|  |  |  |
| --- | --- | --- |
| **NAME** | **REGISTRATION NUMBER** | **STUDENT NUMBER** |
| KIGGUNDU JORDAN LINCOLN | 19/U/8820/EVE | 190078820 |
| SSEBWANA IBRAHIM | 19/U/8598/EVE | 190078598 |
| TURYAHEBWA ALEX | 18/U/23405/EVE | 1800723405 |
| KABASUMBI KEINEMBABAZI JOY | 20/U/23151/PS | 2000723151 |

# INTRODUCTION

* 1. Purpose

The software design document describes the architecture and system design of Character Draw Assessment Tool for KinderCare. CDAT will help LS children remember and create the different shapes of characters, and help their teachers to assess them by giving them assignments. The audience of this document is the software developers and supervisor.

* 1. Scope

CDAT shall consist of two interfaces. The web interface is used by the teacher to give assignments to pupils as well as register, deactivate and activate the pupils. The command line interface is used by the pupils to attempt assignments and also send activation requests to the teacher which uses client-server architecture with the database on the server. The CDAT shall of four components namely; web interface, command line interface, grading, reports generation and database.

* Goals/objectives
* To help pupils attempt assignments using the command line interface
* To help pupils to view all assignments when logged into the command line interface
* To help pupils to check the status of their assignments by displaying the status report of the pupil summarizing all assignments
* To help the teacher give assignments to pupils using the web interface
* To automatically award scores to submitted assignments
* To help the teacher register, deactivate and deactivate a pupil using an interface provided
* Benefits
* Helps LS children to remember and create different shapes from A-Z (All in capital) through assignments given.
* Eases decision- making for the teacher after viewing necessary reports.
* Helps pupils improve their working speed since they are expected to submit their assignment within the required time.
  1. Overview

This software design document is organized into chapters 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 CDAT.

Chapter 3: describes the system architecture, which comprises of 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.

Chapter 7: contains the requirement matrix for tracing the requirements defined in the software requirements specification.

* 1. Referential Material
  2. Definitions and acronyms

LS children – Lower School children

CDAT – Character Draw Assessment Tool

KC – KinderCare

# SYSTEM OVERVIEW

CDAT shall be designed to assess pupils of KinderCare in Character draw; including registering pupils, pupils attempting assignments, automatically awarding scores to submitted assignments. The teacher can leave a comment against the pupil’s score and the pupil views this comment once logged in. CDAT will also generate reports for example the status report for the pupil and all necessary reports to help the teacher in decision making.

# SYSTEM ARCHITECTURE

## 3.1 Architectural design

The system uses a 3-Tier client/server model which consists of two client machines i.e. teacher and pupil, application components and a database server.

Architectural design of the Character Draw Assessment Tool

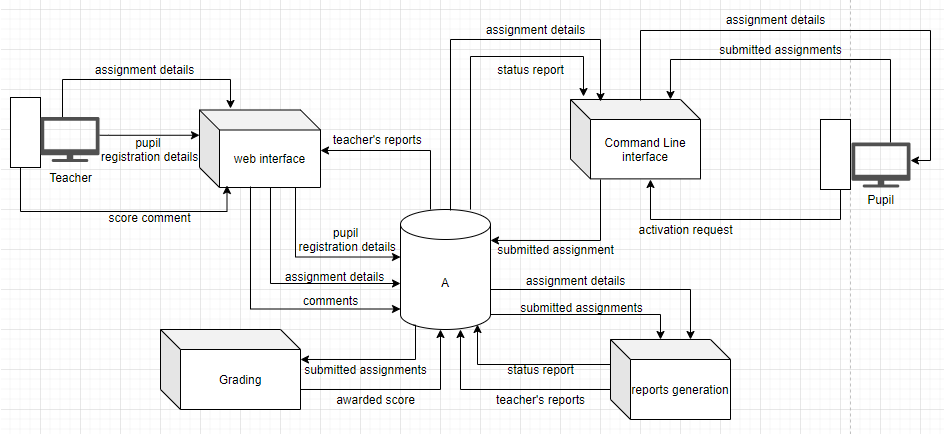


Figure 3.1 architectural design for a character draw assessment tool

**Web interface**

The web interface shall be used to register pupils after which these details will be stored in the database, A. In addition, the web interface will be used by the teacher to give assignments to the pupils.

**Command Line Interface**

This component will be used by the pupil to attempt assignments that are not yet expired and are still open. The submitted assignments will be stored in the database, A. The interface will also provide the total time taken to complete the assignment as well as an acknowledgment for submitting the assignment, for the pupil to see.

**Grading**

This component will automatically award scores for the submitted assignments and the awarded score stored in the database, A.

**Reports generation**

A status report of the pupil summarizing all assignments i.e. how many attempted, average score, percentage missed, percentage attempted.

The teacher also views all necessary reports that help in decision-making.

3.2 Decomposition Description

The entire system will follow a structured design approach. The command line interface will run as a background process on the server, and shall be developed on using C language which is structured language. The web interface, grading and reports generation will also follow structured design approach. We shall use raw php and not laravel.

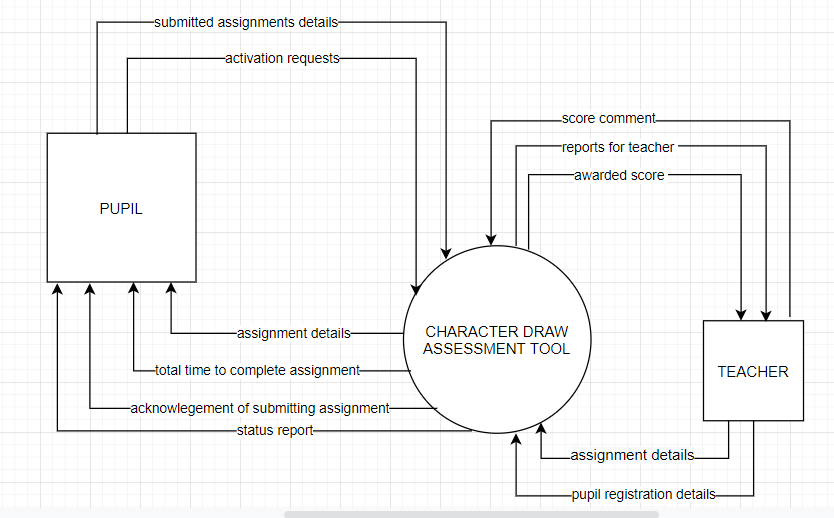


Figure 3.2 context diagram for character draw assessment tool

**Description of the context diagram**

A teacher enters the assignment details, and enters the pupils’ registration details into the system. The teacher who is able to view the awarded score comments on it, and also uses the necessary reports in decision making.

The pupils attempt and submit assignments, see all assignment details and view the status report summarizing all assignments.

**Structural decomposition diagrams for each process of the system**

Approaches used here include functional decomposition diagrams

Level 1 data flow diagram for the character draw assessment tool

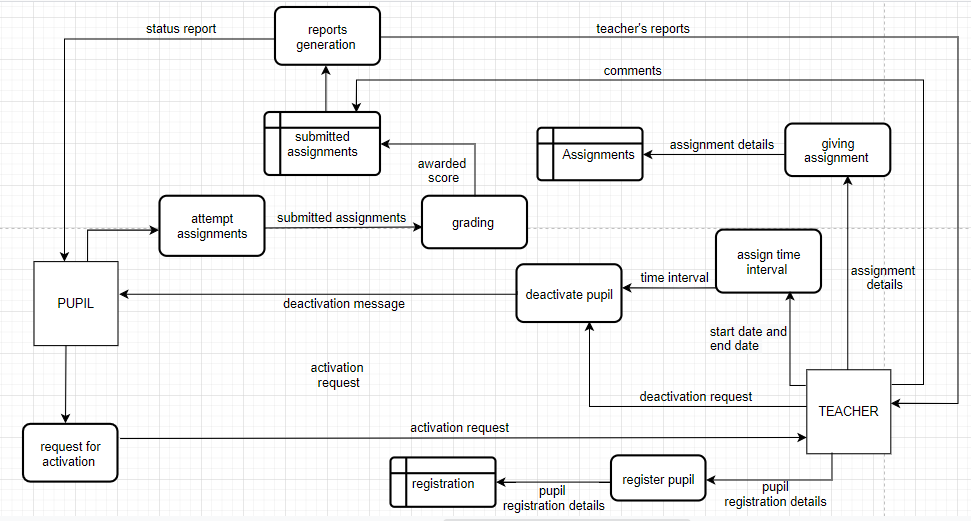
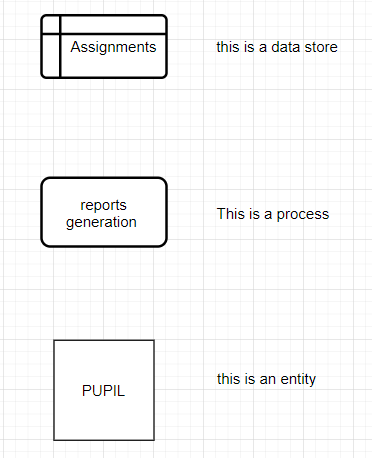


Figure 3.3 level 1 data flow diagram for character draw assessment tool

Description of the diagram



**Process 1 Register pupils**.

The teacher registers pupils that will use CDAT. The pupils’ registration details which include first name, last name, phone number and user code, are then stored in the registration table in the database, A.

**Process 2 Give assignment.**

The teacher adds only eight characters to the assignment list and specifies the start time and end time of the assignment. The assignment details are then stored in the assignments table in the database, A.

**Process 3 Attempt assignment.**

A registered pupil can attempt all assignments that are still within required time. The pupil attempts one character at a time and submits it.The time taken to attempt that character is recorded and total time taken to attempt the assignment calculated and displayed. An acknowledgement for submitting assignment is also provided. All characters attempted are drawn on the screen in form of stars. These submitted assignments are stored in assignments file in the database, A.

**Process 4 Grading.**

The submitted assignments are automatically awarded scores. The awarded scores are then stored in the assignments file in the database, A.

**Process 5 Deactivate pupil.**

The teacher at a given time can deactivate the pupil. The deactivated pupil cannot attempt assignments even if they are registered.

**Process 6 Request for activation**

To be activated, the pupil sends a request to the teacher.

**Process 7 Reports generation**.

This process checks the assignments details and reports to the pupil if there is an assignment within a specified date range, details of a specified assignment like assignment number, date, how much time is left to close the assignment and how many characters they have in their assignment. The process also checks details of submitted assignments and report about the assignment status (whether attempted or not), generates a status report of the pupil summarizing all assignments (how many attempted, average score, percentage missed, percentage attempted. The teacher should be able to view the necessary reports that can help him or her in decision making.

Structural decomposition diagram

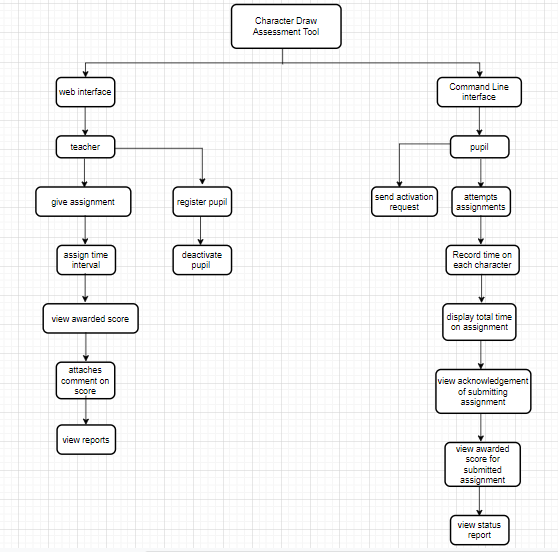


Figure 3.4 structural decomposition diagram

**Description of the structural decomposition diagram**

The process decomposition diagram explains the breakdown of processes within our project. The purpose is to show all the processes and identify relationships and dependencies among them. The command-line interface is used by the pupil to attempt assignments, send activation request to the teacher, check status of the assignments and then after the status report can be generated using the details above. The web application will be used by the teacher to register pupils into the system, give assignments to pupils. The system will record time a pupil spends on each character and after attempting the last character, the total amount of time taken to attempt the assignment is provided as well as an acknowledgement of submitting the assignment. The system will also use the assignment details from the teacher and display for him the awarded score against the assignment and the pupil and then he can attach a comment to the score.

3.3Design 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 such as giving assignments, attempting assignments, automatically awarding scores for the submitted assignments, generating reports and storing pupils’ registration details processed on the server.

The client machines will process functions such as generating reports from the database, giving assignments and registering pupils via the web application, attempting assignments, sending activation requests and viewing status report will be done using the command-line interface. The central shared data repository is to enable the components to store and easily access data items within the system.

DATA DESIGN

4.1 Data Description

The CDAT uses the data about teachers and pupils. Information gathered from the teachers is recorded into the system by the system administrator using the web application whereas the information obtained from the pupil is recorded into the system using the command line interface.

4.2 Data dictionary

This represents a relational database which will be used to store the data in CDAT. Below are the different relations that will be used in the database to store the data;

Table1. Teacher

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraint | Description |
| Teacher ID | Varchar (50) | Primary key | Uniquely identifies each teacher from the rest. |
| User Name | Varchar(50) | Not null | This Identifies the name of a given teacher. |
| User code | Varchar(50) | Foreign key | This references a pupil that is being registered by a teacher. |
| Assignment NO | Varchar(50) | Foreign key | This references the assignment provided by the teacher. |

Table2. Pupil

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraint | Description |
| Pupil ID | Varchar(50) | Primary key | Uniquely identifies each pupil from the rest. |
| FName | Varchar(50) | Not null | This identifies the first name of a given pupil. |
| LName | Varchar(50) | Not null | This identifies the last name of a given pupil. |
| User code | Varchar(40) | unique | This uniquely identifies the passcode of the pupil. |
| Phone Number | Int (10) | unique | This uniquely identifies the phone number of each pupil’s parent/guardian. |
| Assignment NO | Varchar(40) | Foreign key | This references the assignment that is attempted by the pupil. |

Table3. Assignment

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraint | Description |
| Assignment NO | Varchar(50) | Primary key | This uniquely identifies the assignment provided by the teacher. |
| Start Time | date | Not null | This holds the time a pupil begins to attempt the assignment. |
| End Time | date | Not null | This holds the time a pupil ends to attempt the assignment. |
| Time Left | date | Not null | This shows the time left for the assignment to expire. |
| Assignment\_Answers | Varchar(200) | Not null | This holds the answers of a given assignment. |
| Pupil ID | Varchar(50) | Foreign key | This references the pupil who attempts a given assignment |

Table4. Scores

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraint | Description |
| Score ID | Varchar (50) | Primary key | Uniquely identifies the score of a given assignment attempted by the pupil. |
| Assignment NO | Varchar(50) | Foreign key | This references the assignment provided by the teacher. |
| Pupil ID | Varchar(50) | Foreign key | This references the pupil who attempts the assignment provided by the teacher. |

Table5. Activation Requests

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraint | Description |
| Request ID | Varchar (50) | Primary key | Uniquely identifies the request of sent by a given pupil to the teacher. |
| Pupil ID | Varchar(50) | Foreign key | This references the pupil who sends the activation request to the teacher. |
| Teacher ID | Varchar (50) | Foreign key | This references the teacher who responds to the activation requests sent by a given a pupil. |

# COMPONENT DESIGN

Functional algorithmic descriptions for KinderCare CDAT for closer analysis.

*Register user*

*Represents how a teacher registers a pupil*

*Start:*

*Enter* u*ser first name, last name, phone number and user code.*

*Submit registration form*

*Details saved to database.*

*Stop*

*Give assignment*

*Represents how a teacher gives an assignment*

*Start:*

*For each assignment*

*Add only up to only 8 characters to assignment list*

*Record start time and end time*

*End for each*

*Stop*

*Attempt assignment*

*Represents how a pupil attempts an assignment*

*Start:*

*For each character*

*Attempt character and submit*

*Record time taken to attempt that character*

*Draw characters in question and time taken to attempt*

*End for each*

*Calculate and display total time taken attempt*

*Provide acknowledgement for submitting assignment*

*Save details to database*

*Stop*

Grading of a pupil

*Start:*

*If submitted assignment is acknowledged*

*Award score*

*Stop*

Deactivation of a pupil

*Start:*

*Enter pupil user code*

*Pupil deactivated*

*Stop*

Commands used by pupil

View all assignments to user.

*Start:*

*Enter*

*If user not deactivated*

*View all assignments*

*If assignment still within required time*

*Else*

*Stop*

View specific assignment

*Start:*

*Enter the assignment identification number*

*If user not deactivated*

*View specific assignment*

*Else*

*Stop*

Check for assignment within specific dates

*Start:*

*Enter the specified date range*

*If assignment within required time*

*View assignment*

*else*

*Stop*

Request for activation

*Start:*

*Enter request for activation*

*Request forwarded*

*Stop*

# 6. HUMAN USER INTERFACE

**6.1 overview of user interface**

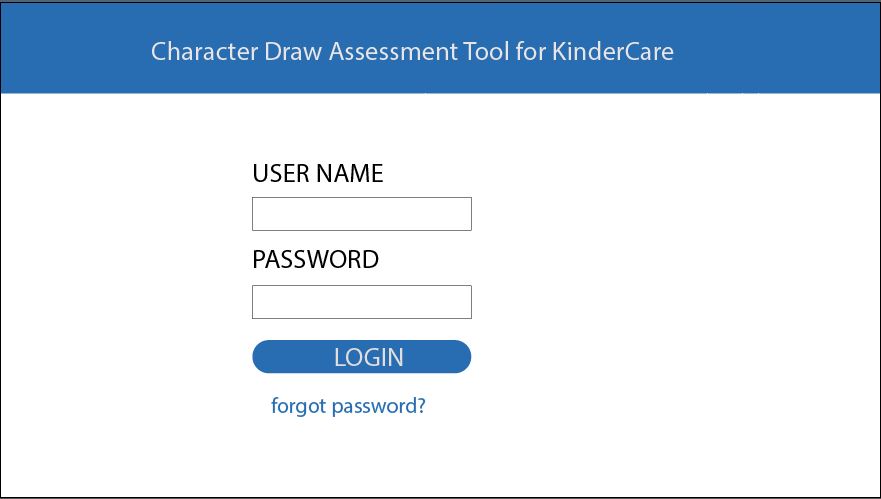
The system will be used by mainly two groups of people

* Teachers
* Pupils

Teachers will have a web interface through their browsers having a form for registering a pupil’s first and last name, phone contact and user code. A teacher can set assignments adding upto 8 characters at most including the start and stop time. He or she will comment on a pupil’s score and view necessary reports.

Pupils will use a command line interface to attempt their assignments having specific commands. On every attempt the character in question will be drawn and time taken to attempt question is displayed.

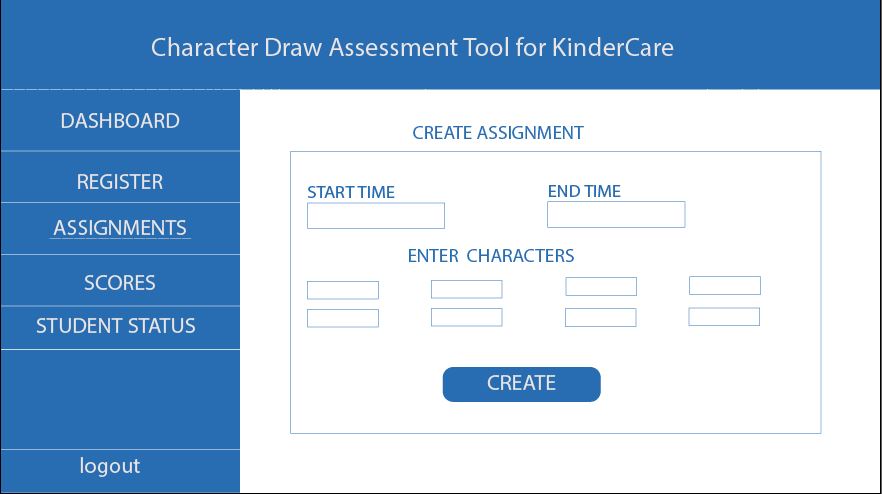
**6.2 Screen images**

****

**Fig 1: shows the login page of a teacher**

****

**Fig 1: shows the registration page used by a teacher to register students**

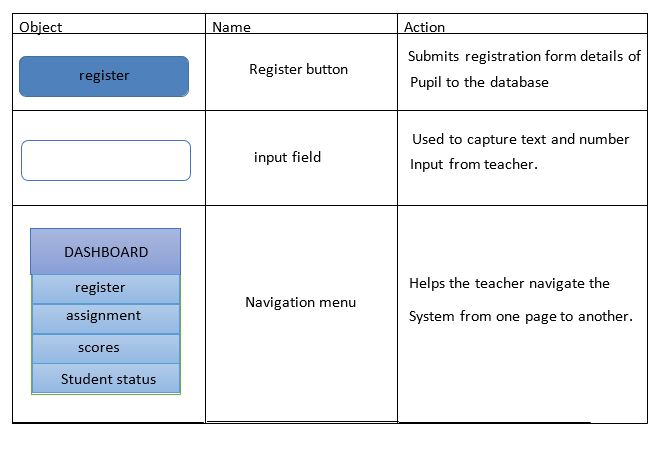
****

**Fig 1: shows the page used by a teacher to create an assignment.**

**Command line interface for pupil**



**6.3 Screen objects and Actions**

****

# 7. REQUIREMENTS MATRIX