PRESCHOOL MANAGEMENT SYSTEM

by Kill Switch Development Team

Prepared by Addo Kwame Okai

COE 356, Introduction to Software Engineering

10 May, 2019

**Table of Content**

1.INTRODUCTION………………………………………………………………………………………….3

[1.1 PURPOSE 3](#__RefHeading___Toc1236_3774037178)

[1.2 PROJECT SCOPE 3](#__RefHeading___Toc1238_3774037178)

[1.3 DOCUMENT CONVENTIONS 3](#__RefHeading___Toc1240_3774037178)

[1.4 INTENDED AUDIENCE AND READING SUGGESTION 3](#__RefHeading___Toc1242_3774037178)

[1.5 REFERENCES 4](#__RefHeading___Toc1246_3774037178)

[1.6 TECHNOLOGIES TO BE USED 4](#__RefHeading___Toc1248_3774037178)

[1.7 OVERVIEW 4](#__RefHeading___Toc1250_3774037178)

[2. Overall Description 5](#__RefHeading___Toc1252_3774037178)

[2.1 PRODUCT PERSPECTIVE 5](#__RefHeading___Toc1254_3774037178)

[2.2 PRODUCT FEATURES 5](#__RefHeading___Toc1581_3774037178)

[2.3 Use-Case Model Survey 7](#__RefHeading___Toc1256_3774037178)

[2.4 Data Dictionary 9](#__RefHeading___Toc1258_3774037178)

[2.5 OPERATING ENVIRONMENT 12](#__RefHeading___Toc1583_3774037178)

[2.6 DESIGN and IMPLEMENTATION CONSTRAINTS 13](#__RefHeading___Toc1585_3774037178)

[2.7 ASSUMPTIONS and DEPENDENCIES 13](#__RefHeading___Toc1264_3774037178)

3.SYSTEM FEATURES……………………………………………………………………………………………...13

4.EXTERNAL INTERFACE REQUIREMENT……………………………………………………………………..

[4.1 USER INTERFACES 16](#__RefHeading___Toc1587_3774037178)

4.2 HARDWARE INTERFACES…………………………………………………………………16

4.3 SOFTWARE INTERFACES………………………………………………………………………17

4.4 COMMUNICATION INTERFACES…………………………………………………………17

5.NON-FUNCTIONAL REQUIREMENTS…………………………………….18

5.1PERFORFANCE REQUIREMENTS…………………………………………………………19

[5.2 SAFETY REQUIREMENTS 20](#__RefHeading___Toc1268_3774037178)

[5.3 SECURITY REQUIREMENT 20](#__RefHeading___Toc1589_3774037178)

[5.4 SOFTWARE QUALITY ATTRIBUTES 20](#__RefHeading___Toc1591_3774037178)

# 1. Introduction

## PURPOSE

To develop a fully functional and user interactive online software which can be used to manage preschools.

## PROJECT SCOPE

* Create user who serves as admins for managing the affairs of the preschool.
* Confirm each admin by providing distinct login details.
* Manage all the affairs of the preschool directly from the web app.
* Add teachers and students to the schools database.
* Access all information related to the management of the preschool.
* To ease management of preschools.
* Provide a comfortable user experience.

## DOCUMENT CONVENTIONS

|  |  |
| --- | --- |
| ADMIN | Administrator |
| DB | Database |
| ER | Entity Relationship |

## INTENDED AUDIENCE AND READING SUGGESTION

This project is a prototype for the preschool management system and is to be used specifically by individuals who would like to manage their preschools. Thus has been implements under the guidance of school directors. This project is useful for the directors as well as head teachers of preschools

## REFERENCES

* + IEEE SRS Format.
  + <https://krazytech.com/project>
  + Fundamentals of database systems by Ramakrishnan.
  + www.researchgate.net

## TECHNOLOGIES TO BE USED

|  |  |
| --- | --- |
| APPLICATION ARCHITECTURE | DRAW.IO |
| DATABASE APPLICATION | MYSQL |
| DEVELOPMENT TOOL DEVELOPER | GIT, VISUAL STUDIO CODE,POSTMAN,SUBLIME, GITHUB,ATOM,GOOGLE |
| DESIGNING TOOL | DRAW.IO |

## OVERVIEW

This project is a software to help in managing preschools. It is more useful in current situations where an organization has no management software and has to do most of its management on paper. In this case querying data becomes easier and also makes the management process a lot more stress free.

# Overall Description

## PRODUCT PERSPECTIVE

The preschool database system stores the following information:

* **School details:**

It includes the name of the school, the location, along with the classrooms and programs offered.

* **Student details:**

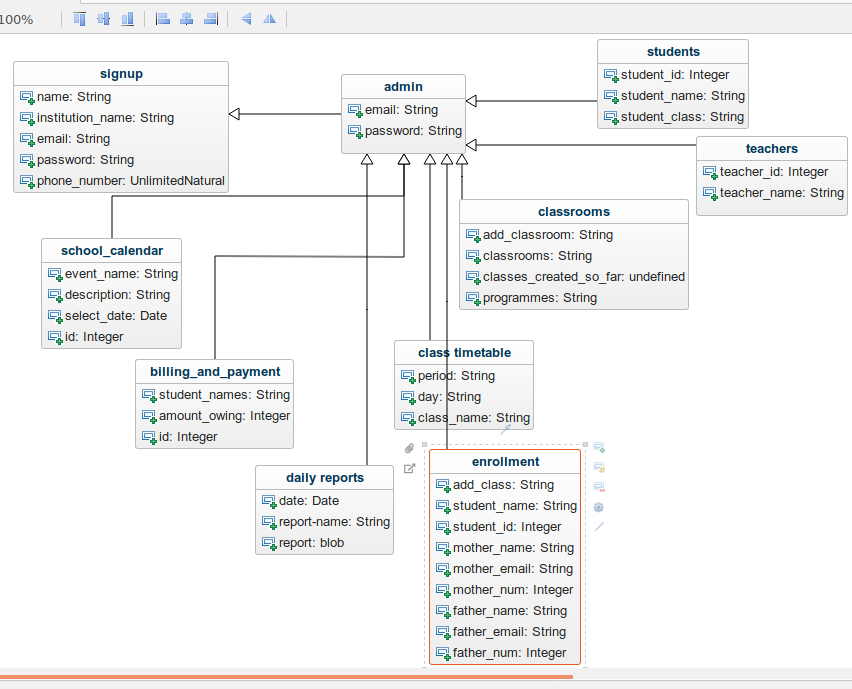
It includes the student’s name, class and contact information of parents. The details may be used for keeping the records of the student for any emergency.

* **Teacher details:**

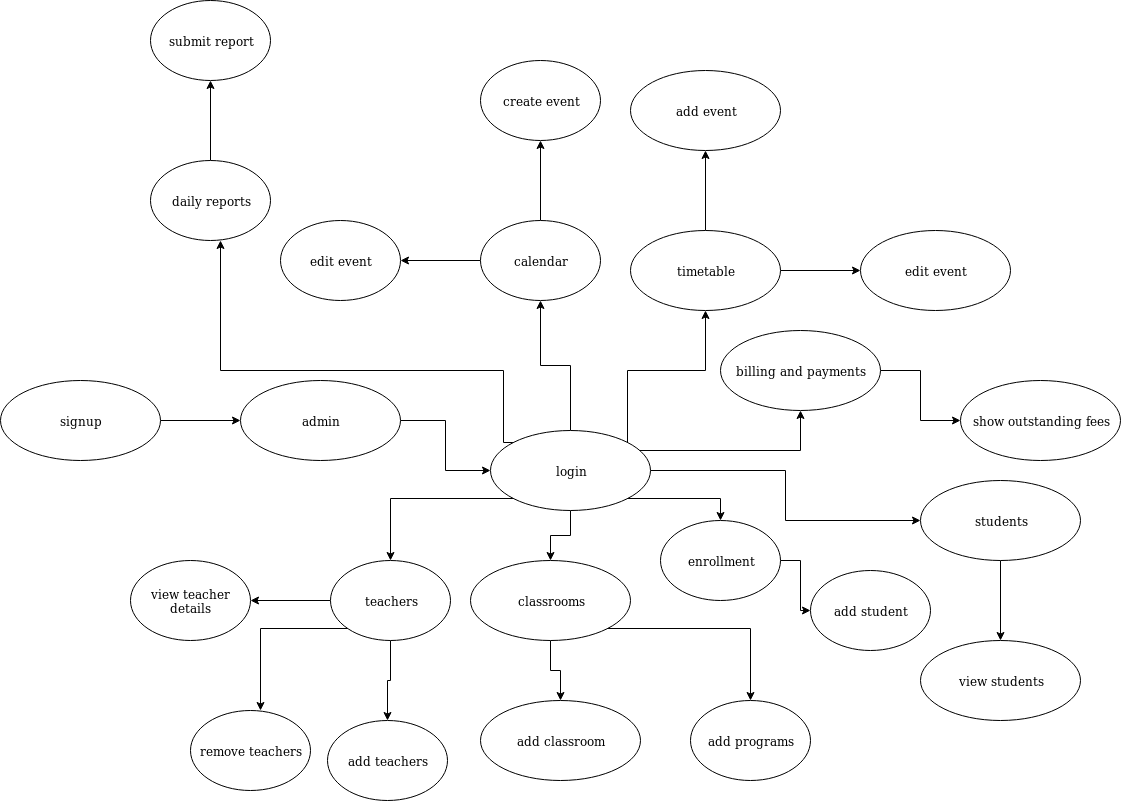
It includes the teacher’s name, class taught, mobile number and email.

## PRODUCT FEATURES

The major features of the preschool management database system is shown in the class diagram below



## Use-Case Model Survey



1. Administrator: - is responsible for registering with the software and creating a profile with the school’s name. He is also manages the management system.
   1. **Sign-up: -** Admin creates an account and registers his/her school.
   2. **Login:-** Admin logs in to access the management system and all the services provided.
   3. **Calendar**: - Admin is able to create and edit events and time periods.
   4. **Timetable:-** Admin is able to create and edit class timetables for all the available classrooms.
   5. **Billings and Payments:-** Admin has access to a billings and payment page that displays student and their outstanding fees if there’s any.
   6. **Enrollment:-** Admin is able to add students into the school’s database providing details like the student’s name, class, name of guardian, etc.
   7. **Classrooms:-** Admin is able to add classrooms providing the various programs available class.
   8. **Students**:- Admin has access to the student database. He/she therefore is able to view the details of every student from the students tab.
   9. **Teachers**:- Admin is able to add teachers, view the details of every teacher and remove teachers from the school’s database.
   10. **Daily reports:-**Admin is able to provide a report in pdf format that entails whatever happens on a school day. Reports must Include any special details worthy of note. Reports are submitted at the end of each day by individual teachers to be added by the admin.
   11. **Log out:-** Admin is signed out and is returned it the homepage.

## Data Dictionary

All tables used in project

a. signup

|  |  |  |
| --- | --- | --- |
| Name | Datatype | nullable |
| name | VARCHAR | no |
| institution\_name | VARCHAR | no |
| password | VARCHAR | no |
| Phone\_number | int | no |

# b. admin

|  |  |  |
| --- | --- | --- |
| Name | Datatype | Nullable |
| email | VARCHAR | no |
| password | VARHAR | no |

c. school\_calendar

|  |  |  |
| --- | --- | --- |
| Name | Datatype | Nullable |
| event\_name | VARCHAR | no |
| description | VARCHAR | no |
| select\_date | VARCHAR | no |
| id | int | no |

d. class\_timetable

|  |  |  |
| --- | --- | --- |
| Name | Datatype | nullable |
| period | VARCHAR | no |
| day | VARCHAR | no |
| class\_name | VARCHAR | no |

e. billings\_and\_payment

|  |  |  |
| --- | --- | --- |
| Name | Datatype | nullable |
| student\_names | VARCHAR | no |
| amount | INT | no |
| id | INT | no |

f. enrollment

|  |  |  |
| --- | --- | --- |
| Name | Datatype | nullable |
| add\_class | VARCHAR | no |
| student\_name | VARCHAR | no |
| student\_id | INT | no |
| mother\_name | VARCHAR | no |
| mother\_email | VARCHAR | yes |
| mother\_num | INT | no |
| father\_name | VARCHAR | no |
| father\_email | VARCHAR | yes |
| father\_num | INT | no |

g. classrooms

|  |  |  |
| --- | --- | --- |
| Name | Datatype | nullable |
| add\_classroom | VARCHAR | no |
| classrooms | VARCHAR | no |
| classes\_created\_so\_far | INT | no |
| programmes | VARCHAR | no |

h. students

|  |  |  |
| --- | --- | --- |
| Name | Datatype | nullable |
| student\_id | INT | no |
| student\_name | VARCHAR | no |
| student\_class | VARCHAR | no |

i. teachers

|  |  |  |
| --- | --- | --- |
| Name | Datatype | nullable |
| teacher\_id | INT | no |
| teacher\_name | VARCHAR | no |

J. daily\_reports

|  |  |  |
| --- | --- | --- |
| Name | Datatype | nullable |
| date | DATE | no |
| report\_name | VARCHAR | no |
| report | BLOB | no |

## OPERATING ENVIRONMENT

Operating environment for the preschool management system is as listed below.

* Centralized database
* Client/server system
* Operating system:Windows
* Database:MySQl
* Platform:HTML, CSS, Javascript, Nodejs

## DESIGN and IMPLEMENTATION CONSTRAINTS

1. The global schema, fragmentation schema and allocation schema.
2. SQL commands for above queries /applications.
3. How the response for application 1 and 2 will be generated. Assuming there are global queries. Explain how various fragments will be combined to do so.
4. Implement the database at least using a centralized database system.

## ASSUMPTIONS and DEPENDENCIES

* The Admin from one school is not being assigned to two or more schools.
* Only the Admin has access to the software.

# 3. SYSTEM FEATURES

* **DESCRIPTION and PRIORITY**

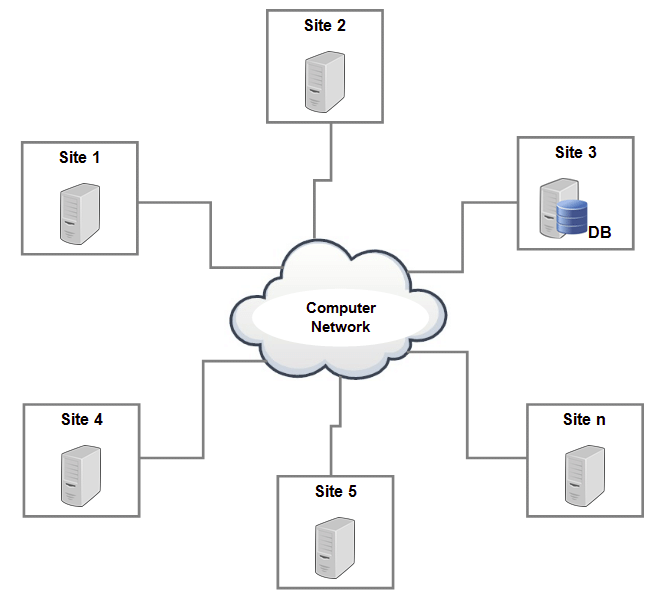
The preschool management system maintains information on students,teachers, outstanding payments and classrooms. Of course this project has a high priority because it is very difficult to manage affairs of an educational institution without having the appropriate tools.

* **STIMULUS/RESPONSE SEQUENCES**
  + Register students
  + Display a detailed list of students and their respective classes.
  + Delete a student
* **FUNCTIONAL REQUIREMENTS**

Other system features include

**CENTRALIZED DATABASE**

A centralized database is a database that is located, stored and maintained in a single location. This location is often a central computer or database system.



**CLIENT/SERVER SYSTEM**

The term client/server refers primarily to an architecture or logical division of responsibilities, the is the application(also known as the front-end), and the server is the DBMS(also known as the back-end).

A client/server system is a distributed system in which,

* + - Some sites are client sites and others are server sites.
    - All the data resides at the server sites.
    - All applications execute at the server sites.

# 4. EXTERNAL INTERFACE REQUIREMENTS

## 4.1 USER INTERFACES

* + Front-end software: HTML/CSS/JS
  + back-end software:MySQL/NodeJS

**4.2 HARDWARE INTERFACES**

* + Windows
  + A browser which supports CGI, HTML& JS

**4.3 SOFTWARE INTERFACES**

Following are the software used for the preschool management system.

|  |  |
| --- | --- |
| Software used | Description |
| Operating system | We have chosen Windows operating system for its best support and user-friendliness. |
| Database | To save students,teachers and class records we have chosen MySQL database. |
| HTML/JS | To implement the project we have chosen HTML/JS language for its more interactive support. |

**4.4 COMMUNICATION INTERFACES**

The project supports all forms of web interfaces. We are using simple electronic forms for the student registration, teacher registration, etc.

# 5. NON-FUNCTIONAL REQUIREMENTS

**5.1 PERFORMANCE REQUIREMENTS**

The steps involved to perform the implementation of the preschool database are listed below

1. **E-R DIAGRAM**

The E-R diagram constitutes a technique for representing the logical structure of a database in a pictorial manner. This analysis is then used to organize data as a relation, normalizing relation and finally obtaining a relation database.

* + - * + **ENTITIES**: Which specify distinct real world items in an application.
        + **PROPERTIES/ATTRIBUTES**: Which specify properties of an entity and relationships.
        + **RELATIONSHIPS**:Which connect entities and represent meaningful dependencies between them.

Admin

Confirm

Enquire

Student/teacher details

Records

Availability

Table 1: the diagram shows the E-Rdiagram of preschool database

1. **NORMALIZATION**

The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored.

If a database is not properly designed it can give rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table. Similarly, in traditional databases as well as improperly designed relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database.

Normalization is the process of breaking down a table into smaller tables. So that each table deals with a single theme. There are three different kinds of modifications of anomalies and formulated the first, second and third normal forms (3NF) is considered sufficient for most practical purposes. It should be considered only after a thorough analysis and complete understanding of its implications.

## 5.2 SAFETY REQUIREMENTS

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

## 5.3 SECURITY REQUIREMENT

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

## 5.4 SOFTWARE QUALITY ATTRIBUTES

* **USABILITY:** The management system should satisfy a maximum number of customer needs.
* **MAINTAINABILITY:** The administrators charger should maintain correct student and teacher records at all times.