Bachelor's Thesis (TUAS)
Degree Program In Information Technology
Internet Technology
2013

Kayode Fadairo

INFORMATION SYSTEM FOR A SECONDARY SCHOOL IN NIGERIA: A CASE STUDY ON CHSA
Abstract

Information system for a secondary school in Nigeria: A case study on CHSA

School Management System (SMS) is a Computer-Based Information System (CBIS) that attempts to consolidate all departments and operations of Comprehensive High School, Ayetoro (CHSA) into a single computer system that services each department’s specific needs. SMS is a convergence of people, hardware and software into an efficient and service delivery system that increases the productivity of the school.

Based on the author's own preliminary research among major schools (primary and secondary) in Nigeria in June 2012, it was discovered that various problems are associated with manual processing, administration and management of schools.

During the author’s own preliminary research, it was found out that the existing system employed a file system of storing data and manual retrieval of information. However, the existing mode of operation was inefficient and less productive.

The developed information system for school management automates all operations performed by the administrators (Bursar, Registrar, Vice-principal and Principal) of CHSA, thereby improving productivity and workflow.
ACKNOWLEDGEMENT

My gratitude goes to Skarli Poppy, and all the lecturers who have imparted knowledge to me all these years. I also want to thank my Supervisor Ferm Tiina for being with me through my thesis.
CONTENTS

Abstract
Acknowledgment
Table of Contents
Notation

CHAPTER ONE: INTRODUCTION

1.1 Introduction..................................................................................................................................................1
1.2 Methodology................................................................................................................................................2
1.3 Arrangement of Work...................................................................................................................................3

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction................................................................................................................................................4
2.2 Case Study: Comprehensive High School, Ayetoro (CHSA).................................................................4
2.3 Review of Existing System..........................................................................................................................6
2.4 Review of Related Findings.........................................................................................................................7
2.5 Proposed System Architecture...................................................................................................................8

CHAPTER THREE: SYSTEM ANALYSIS AND DESIGN

3.1 Analysis of the Proposed System................................................................................................................10
3.1.1 Entity-Relationship Model of the Proposed System........................................................................11
3.1.2 Entity- Life Histories............................................................................................................................15
3.1.3 Use Case Modeling..................................................................................................................................18
3.1.4 System Use Cases....................................................................................................................................19
3.2 System Design .............................................................................................................................................22
3.2.1 Database Design......................................................................................................................................22
3.2.2 Architectural Design.................................................................................................................................26
3.3 Interface Design (Input/Output)..................................................................................................................28
3.3.1 Input Design........................................................................................................................................28
3.3.2 Output Design.......................................................................................................................................28
CHAPTER FOUR: SYSTEM IMPLEMENTATION AND TESTING

4.1 System Implementation ................................................................. 29
4.1.1 Step Requirements Elicitation ................................................ 29
4.1.2 Project Specification ................................................................. 29
4.1.3 System Implementation ............................................................. 29
4.1.4 System Testing ......................................................................... 29
4.1.5 Deployment & Training ............................................................. 30
4.1.6 The Structured System Analysis and Design Methodology (SSADM Approach) ................................................................. 30

4.2 System Requirements .................................................................. 30

4.3 System Testing ............................................................................. 31

4.3.1 Administrator: Registrar .......................................................... 31
4.3.2 Administrator: Bursar ............................................................... 33
4.3.3 Administrator: Principal ........................................................... 34

CHAPTER FIVE: SUMMARY AND CONCLUSION

5.1 Conclusion and Summary ............................................................. 35
5.2 Recommendations ...................................................................... 36

References ....................................................................................... 37
Appendix ......................................................................................... 38

FIGURES

Figure 2.1 Schools Administrative Structure ........................................ 5
Figure 2.2 System Architecture .......................................................... 9
Figure 3.1 Bursar - Student Relationship ............................................. 12
Figure 3.2 Bursar - Bursar Relationship .............................................. 12
Figure 3.3 Bursar - School Relationship ............................................. 12
Figure 3.4 Registrar - Student Relationship ....................................... 13
Figure 3.5 Registrar - Registrar Relationship ................................... 13
Figure 3.6 Vice-principal - Bursar Relationship .................................. 13
Figure 3.7 Vice-principal - Registrar Relationship........................................13
Figure 3.8 Principal - Vice-principal Relationship........................................14
Figure 3.9 School- School Relationship........................................................14
Figure 3.10 Student Entity Life History (ELH) diagram...................................15
Figure 3.11 Bursar Entity Life History (ELH) diagram.................................15
Figure 3.12 Registrar Entity Life History (ELH) diagram.............................16
Figure 3.13 Vice-principal Entity Life History (ELH) diagram...........................16
Figure 3.14 Principal Entity Life History (ELH) diagram.............................17
Figure 3.15 School Entity Life History (ELH) diagram.................................17
Figure 3.16 Use cases for each actor............................................................19
Figure 3.17 Use Case Diagram (System Modeling)........................................20
Figure 3.18. Use Case Diagram (Sub-system Modeling)..............................21
Figure 3.19 System Architecture.................................................................27
Figure 4.1 Student registration.................................................................31
Figure 4.2 Report Generation.................................................................32
Figure 4.3 Result Computation.................................................................32
Figure 4.4 Student Fees Payment.............................................................33
Figure 4.5 Management of the School Account.........................................33
Figure 4.6 Generate School Fees Payment History..................................34
Figure 4.7 Update Login Information........................................................34
TABLES

Table 3.1 the Entity-Relationship Matrix overleaf...............................................11
Table 3.2 Actor Glossary..........................................................................................18
Table 3.3 Account Database Table.............................................................................23
Table 3.4 Payment Database Table.............................................................................23
Table 3.5 Result Database Table................................................................................24
Table 3.6 Staff Database Table...................................................................................24
Table 3.7 Student Database Table.............................................................................25
Table 3.8 Store/ Procedures Table.............................................................................26
## ACRONYMS, ABBREVIATIONS AND SYMBOLS

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMS</td>
<td>School Management System</td>
</tr>
<tr>
<td>CBIS</td>
<td>Computer-Based Information System</td>
</tr>
<tr>
<td>CHSA</td>
<td>Comprehensive High School, Ayetoro</td>
</tr>
<tr>
<td>SSADM</td>
<td>Structured System Analysis and Design Methodology</td>
</tr>
<tr>
<td>MIS</td>
<td>Management Information System</td>
</tr>
<tr>
<td>DSS</td>
<td>Decision Support System</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>HRDB</td>
<td>Human Resource Development Board</td>
</tr>
<tr>
<td>OAS</td>
<td>Office Automation Systems</td>
</tr>
<tr>
<td>GUIs</td>
<td>Graphical User Interfaces</td>
</tr>
<tr>
<td>LAN</td>
<td>Local area Network</td>
</tr>
<tr>
<td>CAN</td>
<td>Campus Area Network</td>
</tr>
<tr>
<td>MAN</td>
<td>Metropolitan Area Network</td>
</tr>
<tr>
<td>WAN</td>
<td>Wide Area Network</td>
</tr>
<tr>
<td>WIFI</td>
<td>Wireless Fidelity</td>
</tr>
<tr>
<td>ADO</td>
<td>Active Data Objects</td>
</tr>
<tr>
<td>OOP</td>
<td>Object Oriented Programming</td>
</tr>
<tr>
<td>ELH</td>
<td>Entity Life History</td>
</tr>
<tr>
<td>RDBMS</td>
<td>Relational Database Management System</td>
</tr>
<tr>
<td>T-SQL</td>
<td>Transact - Structured Query Language</td>
</tr>
<tr>
<td>MS SQL</td>
<td>Microsoft Structured Query Language</td>
</tr>
<tr>
<td>XML</td>
<td>Extended Mark-up Language</td>
</tr>
</tbody>
</table>
CHAPTER ONE

1.1 INTRODUCTION

Ever since the advent of Computers in our society, a lot of criticisms have arisen on the danger it will pose on the society. Critics of this new technology express their fear on how this system will be displacing and replacing all human skills, therefore resulting to mass unemployment. However, further developments continue to prove the critics wrong as the invention continues to create additional jobs for those who identify themselves with the technology. Thus, computers are partners to human beings in any field of human endeavor.

In the last 15 years, the school administration at Comprehensive High School, Ayetoro (CHSA) has changed from one that is highly administrative to one that is recognized as highly strategic and imperative to the overall success of the school. In this fast-moving world, the computer has served as an aid to decision-making, mostly because of its efficiency in terms of speed, accuracy, reliability, cost and security, among others. In the last 15 years, awareness has existed in people as it concerns the use of computers in planning areas of administrative activities. Schools have adopted the use of Management Information System (MIS) and Decision Support System (DSS) in their decision process which has advanced to Relational Database Management Systems.

School Management System (SMS) attempts to consolidate all department’s operations and functions of a school into a single computer system to increase efficiency and productivity.

Based on the author’s own preliminary research among major schools (primary and secondary) in Nigeria in June 2012, it was discovered that various problems are associated with manual processing, administration and management of schools. These problems are:

1. Inefficient account monitoring system which gives room to fraudulent activities.
2. Inconsistent records due to manual file management system.
3. Delay in information retrieval as data are manually stored and processed.
4. Manual processing of student registration and results which constitute a monumental waste of time; this time could be used productively.
5. Low optimization of time, since virtually all operations are executed manually.
6. Overall down time in compiling student results due to manual processes.
7. Lack of information flow across the entire system administrators (bursar, registrar, vice-principals and principal)

The aim of this thesis is to design and implement an information system that will assist CHSA administration.

The objectives are to:

1. Automate student registration process.
2. Implement an embedded online transaction system for school fees and other income.
3. Implement result automation and analysis system.
4. Integrate an effective security and role management.
5. Increase the productivity of the school through improved efficiency and low maintenance cost.

This thesis focuses on the management and administration of schools using Comprehensive High School, Ayetoro (CHSA) as a case study.

1.2 METHODOLOGY

The methodology employed in this thesis is the Structured System Analysis and Design Methodology (SSADM).

The technology/software to be used is:

1. C#.NET Programming Language
2. Microsoft Visual Studio 2010
4. NET Framework 4.0
5. Windows Installer 4.1
1.3 ARRANGEMENT OF WORK

This thesis is divided into five chapters as described below:

CHAPTER ONE – Introduction: This chapter illustrates the general overview of this study. It consists of Introduction, Methodology, and Arrangement of Work.

CHAPTER TWO – Literature Review: The Literature Review gives a detailed description of the area of the work (domain) of this thesis and a review of related publications by researchers pertaining to the work of study.

CHAPTER THREE – System Analysis and Design: This chapter analyzes the proposed system using Logical Data Modeling, Data Flow Modeling and Entity Relationship Modeling. Further, a top down design method will be employed and a clear description of the conceptual or logical model will be obtained from the requirement specifications of the proposed system.

CHAPTER FOUR – Implementation and Documentation: This chapter presents the system implementation and documentation.

CHAPTER FIVE – Conclusion and Summary: This chapter consists of the thesis summary, recommendations and conclusion. Appendix, references and sample source codes are also appended.
CHAPTER TWO

2.1 INTRODUCTION

School Management System (SMS) is a Computer-Based Information System (CBIS) that attempts to consolidate all of the school's departments and operations into a single computer system that services each department's specific needs. SMS is a convergence of people, hardware and software into an efficient and service delivery system that increases the productivity of the school.

School Management Systems provide essential information that assist management with the smooth running of the school. The information is considered to be relevant, timely, accurate, complete and concise and economically feasible. SMSs also have embedded decision support systems that provide the type of information that may not be predictable. They are designed to acknowledge to a wide range of requests and they also assist high administrative officials in making decisions that are not structured or semi-structured.

2.2 CASE STUDY: COMPREHENSIVE HIGH SCHOOL, AYETORO (CHSA)

The need for secondary school system arose in 1960 just after Nigeria gained her independence. The authorities started to consider the need for manpower and the closest solution was to start the secondary school system.

The authorities began by sourcing advice from both international community and by using direct initiatives. Following this, they decided to put science together with the introduction of a technical stream at the school-certificate level on their priority list.

USAID from Harvard University and Ford Foundation assisted the then Western Region Government with the establishment of Comprehensive High School Aiyetoro in February 1963 with a total enrolment of 70 students and lectures from Harvard University, renowned institutions in United State, and education officers in Nigeria. Students population has since risen to over 2000. Ford Foundation provided the funding of the school activities and management until 1973 [1].
CHSA is situated on a 171-hectare land, 37 kilometers west of Abeokuta in Ogun State. The school was founded based on a philosophy and the philosophy was to achieve three goals which are: Serving the potentialities and the educational needs of every child; providing education which is relevant to the technical, economic, social and technical needs of the Nation; and building democratically minded people who would be conscious of their country’s economic, social and political problems in every situation.

Comprehensive High School Aiyetoro had benefited from the services of the best experts in the educational ministry. Going by the list, a large number of renowned educationists served the school. Moreover, the school had benefited international supports following the funding, personnel and material which has resulted in the great achievement of the school.

- **OFFICE OF THE PRINCIPAL**: The principal is responsible for overseeing the functions and operations performed by the vice-principal academic and the vice-principal admin. The principal also ensures that each member of staff performs their individual task and when due by periodic monitoring of the staff. He also makes decisions based on the reports and feedbacks generated by both the vice-principal academic and vice-principal admin.
• **OFFICE OF THE VICE-PRINCIPAL ACADEMIC:** The vice-principal academic is mainly obligated to manage all teaching activities which include drafting time tables, resolving teaching issues, coordinating teachers and lab instructors, and also upholding high academic standard of the school. She also plays a mentoring role for all students of the school by setting aside a scheduled time for counseling and advisory services.

• **OFFICE OF THE VICE-PRINCIPAL ADMIN:** The vice-principal admin manages the administrative aspect of the school. These activities include overseeing the operations of the school bursar, registrar and all other non-academic staff. He also ensures that each expenditure the school is about to make is well scrutinized and then forwards an approval to the bursar for cash payment.

• **OFFICE OF THE BURSAR:** The school bursar is solely responsible for the school finance as regards school fees payment and management of the school account. The bursar manages the school income (school fees and special levies) and issues out all expenditure (salaries and maintenance fees).

• **OFFICE OF THE REGISTRAR:** The school registrar’s office comprises of a number of non-academic staff highly skilled in computer usage. They are mainly responsible for record management; that is keying-in of data into spreadsheet packages as well as retrieval of information.

### 2.3 REVIEW OF EXISTING SYSTEM

The existing system methods and principle mode of operation were examined in order to know how to improve on the system.

Based on the author’s own preliminary research in June 2012, it was discovered that the existing system of Comprehensive High School, Ayetoro (CHSA) has its activities grouped into the following:

1. Registration
2. Account
3. Result computation
REGISTRATION: The registration exercise of CHSA is preceded by the admission process. The Human Resource Development Board (HRDB) of the School manages the admission and/or enrollment of new students by conducting the entrance examination and forwards the names of successful candidates to the school registrar. The school teachers also assist HRDB in setting examination questions, screening exercise and supervision during examinations. The office of the registrar then create a file for each successful student and input their records in Microsoft Excel spreadsheet package for storage and further updates.

ACCOUNT: The school account is managed by the office of the bursar. The bursar receives bank tellers of school fees payment and issues a receipt for each payment. The bursar then forwards the schools fees records to include names of debtors to the office of the registrar for inputting into the individual student file in order to capture a complete information when each student’s records is been retrieved. The bursar also keeps account of the school’s income (school fees payment and special levies), expenditure (salaries, maintenance fees and other miscellaneous expenses) and computes a total balance of the school account. The details of the school account are then made available to the principal and vice-principal admin on request.

RESULT COMPUTATION: The school result computation process is preceded by the each subject teacher grading of student performance which is then collated by each class teacher. The class teacher submits the result grades of every student in his/her class to the vice-principal academic for cross examination and approval. If the vice-principal is not satisfied with a particular subject grading, such result is returned to that subject teacher through the class teacher for remarking; else the result is approved and forwarded to the office of the registrar for recording. The registrar manually inserts the results into the result table with the aid of spreadsheet packages such as Microsoft Excel.

2.4 REVIEW OF RELATED FINDINGS

After review of the existing system at CHSA, the following problems or limitations were identified with the existing system mode of operation:
I. **Gradual transition from manual file management to computerization:** The existing mode of operation partially adopted computerization which involves the use of Office Automation Systems (OAS) such as Microsoft Excel while still retaining the traditional file system of data management. The consequence of this approach is that information cannot be retrieved as a complete whole and this results in data inconsistency due to the fact that data are stored in different locations partially in files and computer systems.

II. **Poor data security:** The existing system currently in use grossly lacks good security features. This is because almost all operations performed on the computer system are carried out using spreadsheet packages which has no provision for security features such as user authentication. In addition, due to the large number of staff using limited systems, anybody, either a staff or an intruder can easily perform an illicit transaction such as fraud, altering of information, deleting relevant files and so on.

III. **Low computer literacy level:** Many school staff members (teaching and non-teaching) have limited knowledge of the use of computer since the existing file system of management encourages computer illiteracy as it requires no computer skills to manage.

2.5 **PROPOSED SYSTEM ARCHITECTURE**

The proposed system architecture (Fig 2.2) illustrates the system architecture and employs the n-tier model which comprises of three Levels. The levels include:

1. **The External Level:** The external level consists of client systems or Graphical User Interface (GUIs) that receives input from the user, informs of data, and displays results to the user as processed information. These client systems are connected through a network such as: Local area Network (LAN), Campus Area Network (CAN), Metropolitan Area Network (MAN), Wide Area Network (WAN), Wireless Fidelity (WiFi), Intranets or the Internet.

2. **The Conceptual Level:** This level is regarded as the logical level because it is neither external nor physical. The level comprises of Active Data Objects (ADO) which are mainly responsible for connecting and interacting with the database. These objects are coded in a class and this class is then instantiated on every
page that interacts with the database. This paradigm of programming is also known as Object Oriented Programming (OOP) which is based on three fundamental principles: Encapsulation, Inheritance and Polymorphism.

3. **The Physical Level**: The physical level acts as a repository data store and is generally called a database. It consists of the relations, stored procedures, transaction logs and triggers that implement every operation perform on the client systems.

![System Architecture Diagram]

*Fig 2.2. System Architecture*
CHAPTER THREE
SYSTEM ANALYSIS AND DESIGN

3.1 ANALYSIS OF THE PROPOSED SYSTEM

The structured system analysis and design methodology SSADM [2] is employed in the analysis and design stages of this thesis. SSADM specifies in advance the modules stages and tasks to be carried out. It also specifies the deliverables to be produced and the methods used to produce the deliverables.

SSADM adopts the waterfall model of the system development. This approach is employed in this thesis because it sets up a comfortable dialogue between the user, analyst, designer and software developer. Structured system analysis and design methodology make use of two key techniques. These are:

1. Entity/ Event Modeling
2. Use Case Modeling

Entity/Event Modeling

This technique helps to recognize, identify and document the events of the organization which affect each entity and the order in which the events occur. This model is made up of a set of entity life histories and relevant documentation.

Therefore the proposed system will be modeled using the tools listed below:

- Entity - Relationship Diagram
- Entity - Life History Diagrams
- Use Case System Diagram
- Use Case Subsystem Diagram
- Database Tables
3.1.1 ENTITY-RELATIONSHIP MODEL OF THE PROPOSED SYSTEM

Identified entities in the system are modeled and the relationships that exist among them. The modeling technique was developed by Peter Chen in 1976 [3]. It is a top-down approach to system design. After observing the operations of CHSA, the following entities were identified (Table 3.1):

- Student
- Bursar
- Registrar
- Vice-principal
- Principal
- School

Table 3.1 below shows the identified entities in the system and their consequence relationships.

**Table 3.1. The Entity-Relationship Matrix overleaf**

<table>
<thead>
<tr>
<th></th>
<th>Student</th>
<th>Bursar</th>
<th>Registrar</th>
<th>Vice-principal</th>
<th>Principal</th>
<th>School</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bursar</td>
<td>Manage</td>
<td>Manage</td>
<td>Manage</td>
<td></td>
<td>Manage</td>
<td></td>
</tr>
<tr>
<td></td>
<td>School</td>
<td>Login</td>
<td>Login</td>
<td></td>
<td>Account</td>
<td></td>
</tr>
<tr>
<td>Registrar</td>
<td>Manage</td>
<td></td>
<td>Manage</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vice-principal</td>
<td>Oversee</td>
<td>Oversee</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Principal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Oversee</td>
<td></td>
</tr>
<tr>
<td>School</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Manage</td>
<td></td>
</tr>
</tbody>
</table>
Participation and cardinality constraints of each of the elements of the matrix are illustrated. This is be done using the Crow Foot notation [5]. The names and the degree of relationships of each entity are shown in Figures 3.1-3.9

Bursar - Student relationship (Fig 3.1) is a one-to-many relationship with mandatory participation for the bursar. This relationship implies that a bursar can manage more than one student school fees.

![Fig 3.1. Bursar - Student Relationship](image)

Bursar - Bursar relationship (Fig 3.2) is a one-to-one relationship which shows that any occurrence of the entity bursar is associated with only one occurrence of the bursar. It illustrates that a bursar can only manage his/her login's information.

![Fig 3.2. Bursar - Bursar Relationship](image)

Bursar - School relationship (Fig 3.3) is a one-to-one relationship which shows that any occurrence of the entity bursar is associated with only one occurrence of the school. This implies that a bursar can only manage account(s) for a single school.

![Fig 3.3. Bursar - School Relationship](image)
Registrar - Student relationship (Fig 3.4) is a one-to-many relationship which has both optional and mandatory participation. The optional participation shows that not all students are managed by a particular registrar. The mandatory participation illustrates that at least a student will be managed by a particular registrar.

**Fig 3.4. Registrar - Student Relationship**

Registrar - Registrar relationship (Fig 3.5) is a one-to-one relationship which shows that any occurrence of the entity registrar is associated with only one occurrence of the registrar. It illustrates that a registrar can only manage his/her login’s information.

**Fig 3.5. Registrar - Registrar Relationship**

Vice-principal - Bursar relationship (Fig 3.6) is a one-to-one relationship which shows that any occurrence of the entity vice-principal is associated with only one occurrence of the bursar. It illustrates that only the vice-principal has supervision over the bursar.

**Fig 3.6. Vice-principal - Bursar Relationship**

Vice-principal – Registrar relationship (Fig 3.7) is a one-to-one relationship which shows that any occurrence of the entity vice-principal is associated with only one occurrence of the registrar. It illustrates that only the vice-principal has supervision over the registrar.

**Fig 3.7. Vice-principal - Registrar Relationship**
Principal – Vice-principal relationship (Fig 3.8) is a one-to-one relationship which shows that any occurrence of the entity principal is associated with only one occurrence of the vice-principal. It illustrates that only the principal has supervision over the vice-principal.

![Fig 3.8. Principal - Vice-principal Relationship](image)

School – School relationship (Fig 3.9) is a recursive one-to-many relationship which requires only one mandatory participation and it illustrates that a school must be under the supervision of one school which might be itself.

![Fig 3.9. School - School Relationship](image)
3.1.2 ENTITY-LIFE HISTORIES

An Entity Life History (ELH) diagram describes the life cycle of entities within the database [2]. It is essentially a diagrammatic technique which provides a picture of all possible outcomes for a particular entity in the system (Figures 3.10-3.15).

![Student Entity Life History (ELH) diagram]

![Bursar Entity Life History (ELH) diagram]
Fig 3.12. Registrar Entity Life History (ELH) diagram

Fig 3.13. Vice-principal Entity Life History (ELH) diagram
Fig 3.14. Principal Entity Life History (ELH) diagram

Fig 3.15. School Entity Life History (ELH) diagram
Summary of Entity-Life History Diagrams

The diagrams are read from left to right. Maintaining records involves creation, deletion and update of related fields of each entity (table). Housekeeping operations are carried out on each entity (table) in order to prevent the database from becoming too difficult to manage.

3.1.3 USE CASE MODELING

The objective of constructing the use-case model is to anatomize enough requirements information to prepare a model that communicates what is required from a user perspective but free of special details about how the system is built and implemented [4].

Use Cases are initiated or triggered by external users called actors. The actor system initiates system activities and a use case for the purpose of finishing some business task that produces something of measurable value. Identifying system actors assist in concentrating on how the system is used and not how is built [4]. Focusing on the actors helps to restructure and further structure the scope and boundaries of the system.

Table 3.2 shows the description of an actor.

Table 3.2. Actor Glossary

<table>
<thead>
<tr>
<th>Actor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student</td>
<td>A person studying for a leaving school certificate.</td>
</tr>
<tr>
<td>Bursar</td>
<td>An employee in charge of school fees and account operations.</td>
</tr>
<tr>
<td>Registrar</td>
<td>An employee in charge of student records and results.</td>
</tr>
<tr>
<td>Vice-principal</td>
<td>An employee that oversees the bursar and registrar.</td>
</tr>
<tr>
<td>Principal</td>
<td>An administrator that oversee the vice-principal and school.</td>
</tr>
</tbody>
</table>
3.1.4 System Use Cases

A use case captures the interactions with the user in a technology and implementation free. Use-cases describe how a real-world actor communicates with the system and how they will use the system.

The elicited functional requirements from the requirement discovery process is represented as use cases. Thus, the system use cases are: (Figs 3.16)

- Manage School Fees
- Manage Account
- Manage Bursar’s Login Information
- Register Student
- Manage Result
- Manage Staff
- Manage Registrar’s Login Information
- Oversee Bursar
- Oversee Registrar
- Manage Vice-principal’s Login Information
- Oversee Vice-principal
- Generate Reports
- Manage Principal’s Login Information

**Fig 3.16. Use cases for each actor**
Fig 3.17. Use Case Diagram (System Modeling)
Fig 3.18. Use Case Diagram (Subsystem Modeling)
3.2 SYSTEM DESIGN

In the design stage a top-down method is used. A clear description of the logical model was obtained from the requirement specifications. The conceptual model goes through a series of improvements which give the physical model of the system. The implementation of the requirement specifications leads to the physical model.

The implemented design of the system was carried out in three phases:

1. Database design
2. Architecture design
3. Interface Design(Output/Reports)

The design has a front-end Graphical User Interface (GUI) Windows-based application and a back-end Relational Database Management System (RDBMS). The front-end is designed using C#.NET as a programming language - while the back-end is designed with using Microsoft Structured Query Language (SQL) Server 2008. The database is developed based on the Entity-Relationship Model.

3.2.1 DATABASE DESIGN

The physical model of database is constructed based on the ER data model. This essentially means that the logical model is converted to the physical view, i.e., the database. The database makes use of two database objects, which are:

- **Tables**: Store information about each entity of the application.
- **Store/Procedures**: Implement all the functionality of the application with Transact - Structured Query Language (T-SQL).
The tables and the associated functions are described in tables 3.3-3.7

This database table (Table 3.3) is used for recording all account transactions.

**Table 3.3. Account Database Table**

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransactionID</td>
<td>Integer</td>
</tr>
<tr>
<td>SchoolFees</td>
<td>Varchar</td>
</tr>
<tr>
<td>PaymentID</td>
<td>Integer</td>
</tr>
<tr>
<td>Expenditure</td>
<td>Decimal</td>
</tr>
<tr>
<td>ExpenditureDetails</td>
<td>Varchar</td>
</tr>
<tr>
<td>AccountBalance</td>
<td>Decimal</td>
</tr>
<tr>
<td>DateOfTransaction</td>
<td>DateTime</td>
</tr>
</tbody>
</table>

The table (Table 3.4) is the database table through which school fees payment are recorded.

**Table 3.4. Payment Database Table**

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PaymentID</td>
<td>Integer</td>
</tr>
<tr>
<td>RegistrationNo</td>
<td>Integer</td>
</tr>
<tr>
<td>SchoolFees</td>
<td>Decimal</td>
</tr>
<tr>
<td>AmountPaid</td>
<td>Decimal</td>
</tr>
<tr>
<td>Balance</td>
<td>Decimal</td>
</tr>
<tr>
<td>BankPaidTo</td>
<td>Varchar</td>
</tr>
<tr>
<td>PaidInBy</td>
<td>Varchar</td>
</tr>
<tr>
<td>TellerNo</td>
<td>Varchar</td>
</tr>
<tr>
<td>ReceiptNo</td>
<td>Varchar</td>
</tr>
<tr>
<td>DateOfPayment</td>
<td>DateTime</td>
</tr>
<tr>
<td>DateOfBalance</td>
<td>DateTime</td>
</tr>
</tbody>
</table>
The database table (Table 3.5) is used for storing result details.

**Table 3.5. Result Database Table**

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>RegistrationNo</td>
<td>Integer</td>
</tr>
<tr>
<td>SubjectName</td>
<td>Varchar</td>
</tr>
<tr>
<td>SubjectTestScore</td>
<td>Integer</td>
</tr>
<tr>
<td>SubjectExamScore</td>
<td>Integer</td>
</tr>
<tr>
<td>SubjectAverageScore</td>
<td>Integer</td>
</tr>
<tr>
<td>SubjectGrade</td>
<td>Char</td>
</tr>
<tr>
<td>Class</td>
<td>Varchar</td>
</tr>
<tr>
<td>Term</td>
<td>Varchar</td>
</tr>
<tr>
<td>TeacherComment</td>
<td>Varchar</td>
</tr>
<tr>
<td>PrincipalComment</td>
<td>Varchar</td>
</tr>
</tbody>
</table>

The database table (Table 3.6) is used for recording staff details.

**Table 3.6. Staff Database Table**

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>StaffID</td>
<td>Integer</td>
</tr>
<tr>
<td>SurName</td>
<td>Varchar</td>
</tr>
<tr>
<td>FirstName</td>
<td>Varchar</td>
</tr>
<tr>
<td>OtherName</td>
<td>Varchar</td>
</tr>
<tr>
<td>ResidentialAddress</td>
<td>Varchar</td>
</tr>
<tr>
<td>MailingAddress</td>
<td>Varchar</td>
</tr>
<tr>
<td>Email</td>
<td>Varchar</td>
</tr>
<tr>
<td>Telephone</td>
<td>Varchar</td>
</tr>
<tr>
<td>DateOfBirth</td>
<td>DateTime</td>
</tr>
<tr>
<td>DateOfRecruitment</td>
<td>DateTime</td>
</tr>
<tr>
<td>Salary</td>
<td>Decimal</td>
</tr>
<tr>
<td>Position</td>
<td>Varchar</td>
</tr>
<tr>
<td>Gender</td>
<td>Char</td>
</tr>
</tbody>
</table>
The database table (Table 3.7) stores details of students upon registration.

**Table 3.7. Student Database Table**

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>RegistrationNo</td>
<td>Integer</td>
</tr>
<tr>
<td>SurName</td>
<td>Varchar</td>
</tr>
<tr>
<td>FirstName</td>
<td>Varchar</td>
</tr>
<tr>
<td>OtherName</td>
<td>Varchar</td>
</tr>
<tr>
<td>Age</td>
<td>Integer</td>
</tr>
<tr>
<td>DateOfBirth</td>
<td>DateTime</td>
</tr>
<tr>
<td>StateOfOrigin</td>
<td>Varchar</td>
</tr>
<tr>
<td>Religion</td>
<td>Varchar</td>
</tr>
<tr>
<td>ContactAddress</td>
<td>Varchar</td>
</tr>
<tr>
<td>SchoolAttended</td>
<td>Varchar</td>
</tr>
<tr>
<td>Guardian</td>
<td>Varchar</td>
</tr>
<tr>
<td>GuardianAddress</td>
<td>Varchar</td>
</tr>
<tr>
<td>Telephone</td>
<td>Varchar</td>
</tr>
<tr>
<td>Email</td>
<td>Varchar</td>
</tr>
<tr>
<td>BloodGroup</td>
<td>Varchar</td>
</tr>
<tr>
<td>Genotype</td>
<td>Varchar</td>
</tr>
<tr>
<td>Drugs</td>
<td>Varchar</td>
</tr>
<tr>
<td>ClassOfAdmission</td>
<td>Varchar</td>
</tr>
<tr>
<td>CurrentClass</td>
<td>Varchar</td>
</tr>
<tr>
<td>Term</td>
<td>Varchar</td>
</tr>
<tr>
<td>DateAdded</td>
<td>DateTime</td>
</tr>
</tbody>
</table>
Table 3.8 shows the details of the stored procedures

**Table 3.8. Store/Procedures Table**

<table>
<thead>
<tr>
<th>Stored Procedures Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bursar_Login</td>
<td>Authenticates a valid bursar</td>
</tr>
<tr>
<td>Registrar_Login</td>
<td>Authenticates a valid registrar</td>
</tr>
<tr>
<td>VicePrincipal_Login</td>
<td>Authenticates a valid vice-principal</td>
</tr>
<tr>
<td>Principal_Login</td>
<td>Authenticates a valid principal</td>
</tr>
<tr>
<td>Student_Registration</td>
<td>Registers a new student records</td>
</tr>
<tr>
<td>Staff_Registration</td>
<td>Registers a new staff</td>
</tr>
<tr>
<td>SchoolFess_Payment</td>
<td>Insert school fees payment</td>
</tr>
<tr>
<td>Update_Payment</td>
<td>Updates subsequent payment</td>
</tr>
<tr>
<td>Insert_Income</td>
<td>Insert income into account table</td>
</tr>
<tr>
<td>Insert_Expenditure</td>
<td>Insert expenditure into account table</td>
</tr>
<tr>
<td>Compute_Balance</td>
<td>Calculate account balance</td>
</tr>
<tr>
<td>Bursar_Login_Update</td>
<td>Update bursar’s login credentials</td>
</tr>
<tr>
<td>Search_PaymentByName</td>
<td>Searches school fees payment via student name</td>
</tr>
<tr>
<td>Search_PaymentByRegNo</td>
<td>Searches school fees payment via registrationNo</td>
</tr>
<tr>
<td>Search_PaymentByClass</td>
<td>Searches school fees payment via class</td>
</tr>
<tr>
<td>Search_PaymentByTerm</td>
<td>Searches school fees payment via term</td>
</tr>
<tr>
<td>Update_Student</td>
<td>Update Student records</td>
</tr>
<tr>
<td>Update_Staff</td>
<td>Update staff records</td>
</tr>
<tr>
<td>Update_Registrar_Login</td>
<td>Update registrar login credentials</td>
</tr>
<tr>
<td>Insert_Result</td>
<td>Insert result data</td>
</tr>
<tr>
<td>Update_Result</td>
<td>Update changes in result data</td>
</tr>
<tr>
<td>Compute_Result</td>
<td>Begins result computation</td>
</tr>
<tr>
<td>Search_ResultByRegNo</td>
<td>Returns result details via registration number</td>
</tr>
<tr>
<td>Search_ResultByName</td>
<td>Returns result details via student name</td>
</tr>
<tr>
<td>Update_VicePrincipa_Login</td>
<td>Update changes made to vice-principal login info</td>
</tr>
<tr>
<td>Update_Principal_Login</td>
<td>Update changes made to principal login info</td>
</tr>
</tbody>
</table>

### 3.2.2 ARCHITECTURAL DESIGN

The development and implementation of the School Management System application involves having a complete understanding of its design architecture. The design has two phases (Front-end and Back-end). The front-end is the user interface design which consists of windows forms. The back-end mainly contains the database of the application; it contains tables for storing input data and stored procedures for implementing database transactions. The front-end (windows form) and back-end (database) are connected via a middleware program called Active Data Object (ADO.NET) (Fig 3.19)
The middleware layer is implemented using a strongly typed dataset in Microsoft .Net framework. The strongly typed dataset works in such a way that the whole operation that involves the database is performed with the strongly typed dataset component. All the stored procedures and various SQL transactions at the database are mapped to appropriate methods which are created by the dataset component. Strongly typed datasets are implemented as DBAccess class in this application using the concept of Object Oriented Programming (OOP) which supports Encapsulation, Inheritance and Polymorphism.

Fig 3.19. System Architecture
3.3. INTERFACE DESIGN (INPUT/OUTPUT)

The interface design composes of both input and output designs of the School Management System.

3.3.1 Input Design

The system input is designed to receive data from the user or administrators (Bursar, Registrar, Vice-principal and Principal). The first set of input to the system is a pair of Username and Password. After the pair has been entered, the system authenticates the user by validating the username and password. This is done by calling the login stored procedure which in turn checks the database by verifying if both the Username and Password supplied exist in the database. If true, the system authenticates the user else, the user is denied access. The other inputs to the system are only carried out after the user has been granted access to the system. These inputs include: Student registration details, Result data for computation, Staff details, School fees payment data etc.

3.3.2 Output Design

The system output present information to the user either as a single query or in report format. The system output varies from search results, query reports, login authentication message, exception messages, e.t.c. The system output is the most important component of a working system because the interactivity of the system depends on its output. This is the main reason why the output of an information system determines the effectiveness and efficiency of the system.
CHAPTER FOUR

SYSTEM IMPLEMENTATION AND TESTING

4.1 SYSTEM IMPLEMENTATION

The implementation of the proposed system has a front-end (Graphical User interface) GUI-based application and a back-end Relational Database Management System. The front-end is implemented with C#.NET (using Microsoft Visual Studio 2008 platform) while the back-end is designed with Microsoft Structured Query Language (MS SQL) Server 2008. The database implementation is based on the Entity-Relationship Model. The development processes followed for this project are:

4.1.1 Requirements Elicitation

Requirements elicitation remains one of the most important and challenging steps in systems analysis and design [6].

This phase of the implementation documented the strategic and tactical objectives of the system. It determines the workflow and functional requirements, processes and identifies areas where information needs to be shared, tracked and integrated.

4.1.2 Project Specification

A formal system implementation plan is created and finalized during this phase. A system administrator coordinates all stakeholders as well the system integrator or consultant(s) managing the implementation.

4.1.3 System Implementation

During this phase, installation of the software, workflow policies and procedures are implemented. System configuration, privileges, preferences and security are determined.

4.1.4 System Testing

An operational prototype of the system application is up and running at this stage. Operational issues are identified and resolved, and basic modifications are made.
4.1.5 Deployment & Training

During this phase, the application is completely operational. Training begins for all system administrators and teachers. Ongoing support is provided by the system developers through periodic maintenance.

4.1.6 The Structured System Analysis and Design Methodology Approach

SSADM takes a modular approach to the School Management Software initiative using the best practice guide to protect the successful implementation of one part of the project at once [2]; this makes the project more scalable. The Relational Database Management (RDBM) of the application is implemented first and tested to make sure that it conforms with all requirements (specifications, business rules and constraints). After that, the front-end GUI (Graphical User Interface) application is integrated. This methodology allows the project to start and break down the implementation process into modules that are attainable, based on the requirements and resource availability.

Technology/Software Used:

- Microsoft Visual Studio 2010
- C#.NET programming language.
- .NET Framework 4
- Microsoft SQL Server 2008 (Database Engine)
- T-SQL (Transact Structured Query Language) database programming language.
- ADO.NET (middleware program)
- Microsoft Visio 2007 (UML Class Diagram, DFD, ER, Use Case)

4.2 SYSTEM REQUIREMENTS

Hardware Requirements

1. Microprocessor: Pentium III, 337MHz (Minimum)
2. Random Access Memory (RAM) Space: 256MB (Minimum)
3. Hard Disk Storage Capacity: 40GB (Minimum)
Software Requirements
1. Microsoft SQL Server 2008
2. Windows Installer 4.5
4. .NET Framework 4

4.3 SYSTEM TESTING

4.3.1 Administrator: Registrar

The registrar is expected to select the administrator role, supply admin name and password before gaining access to the School Management Software. When authenticated, the registrar is given the privilege to perform the operations described in Figures 4.1-4.3.

The student registration form (Fig 4.1) enables the registrar to register a new student after completing the entrance examination successfully.

Fig 4.1. Student registration
Report generation form (Fig 4.2) enables the registrar to generate reports on students as regards their details and other important information.

![Report Generation](image)

**Fig 4.2. Report Generation**

The result computation subsystem (Fig 4.3) enables the automation and computation of student results which is designed to replace the existing broadsheets currently in use.

![Result Computation](image)

**Fig 4.3. Result Computation**
4.3.2 Administrator: Bursar

The bursar is required to select the administrator role, supply admin name and password before gaining access to the School Management Software. When authenticated, the bursar is given the privilege to perform the operations described in Figures 4.4-4.5.

The school fees payment subsystem (Fig 4.4) allows the bursar to insert record of school fees.

Fig 4.4. Student Fees Payment

The account subsystem (Fig 4.5) encompasses the school fee, other income and expenditure. It also presents a structured balance sheet of the school’s account.

Fig 4.5. Management of the School Account
4.3.3 Administrator: Principal

The principal is required to select the administrator role, supply admin name and password before gaining access to the School Management Software. When authenticated, the principal is given the privilege to perform the following operations:

Generate school fees payment history: This form allows the principal to generate school fees payment history for a particular student within a stipulated time period (Fig. 4.6)

![Fig 4.6. Generate School Fees Payment History](image)

Update login info: This form (Fig. 4.7) allows the principal/vice-principal to update or make changes to his/her login credentials.

![Fig 4.7. Update Login Information](image)
CHAPTER FIVE
SUMMARY, CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION AND SUMMARY

The design and implementation of an information system for Secondary School Management is completed. During the analysis phase, it was found out that the existing system employed a file system of storing data and manual retrieval of information. However, the existing mode of operation was inefficient and less productive.

The developed information system for school management automates all operations performed by the administrators (Bursar, Registrar, Vice-principal and Principal) of Comprehensive High School, Ayetoro (CHSA), thereby improving productivity and workflow. The developed information system provides the following functionalities:

1. An internal communication mechanism that performs most activities involved in administration and managerial activities of the school.
2. A well normalized database system for the conceptual, logical and physical model of the repository data store.
3. Easy insertion and retrieval of student and staff record.
4. Reduction of manpower and time wasting.
5. An automated billing system for school fees and other account activities.
6. A result automation and computation for accurate and easy result computation.
7. Reduction of errors encountered in the existing system due to total human dependency.
8. A well designed security provision to prevent unauthorized access and validating ethical users.

Compared to other existing systems, the system developed is better in the following ways:

1. It automates all operations performed by the administrators (Bursar, Registrar, Vice-principal and Principal) of Comprehensive High School, Ayetoro (CHSA) which effectively reduces the resources (time and money) spent on the existing system.
2. It provides a Relational Database Management System (RDBMS) which acts as a repository data store for the information system and a friendly Graphical User Interface (GUI) for easy usage.
3 It serves as an internal communication mechanism for administrators and for easy storage and retrieval of information.
4 It enables security provision by authenticating each administrator before granting he/she access to information.

5.2 RECOMMENDATIONS

The global trend in the world today is that the world is said to be a global village as a result of the internet. For the purpose of future development and/or software forward engineering, the following issues should be taken into considerations:

1 Integration of XML (Extended Mark-up Language) enabled web service that allows the connection of multiple school branches via the internet. The functions and operations of the information system will be hard-coded in the web services such that data are encrypted during storage and decrypted when retrieved.
2 A Short Message System (SMS) alert can be embedded to the system in order to update parent/guardian about payment notification and dissemination of information.
REFERENCES


APPENDIX

SAMPLE PROGRAM LISTING

Registration Subsystem

```csharp
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace SchoolManagementSoftware
{
    public partial class frmStudentBiodata : Form
    {
        public frmStudentBiodata()
        {
            InitializeComponent();
        }

        private void frmStudentBiodata_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 'secondarySchool_DBDataSet3._getStudents' table. You can move, or remove it, as needed.
            //this._getStudentsTableAdapter.Fill(this.secondarySchool_DBDataSet3._getStudents);
            txtAge.Text = db.ExecuteScalar("getAge").ToString();
            txtUpContAdd.Text = db.ExecuteScalar("getContactAddress").ToString();
            txtGuardAddrs.Text = db.ExecuteScalar("getAddress").ToString();
            txtUpPhoneNo.Text = db.ExecuteScalar("getPhoneNo").ToString();
            txtUpEmail.Text = db.ExecuteScalar("getEmail").ToString();
            txtUpCurrentCls.Text = db.ExecuteScalar("getCurrentClass").ToString();
            else
```
private void rbtnYes_CheckedChanged(object sender, EventArgs e)
{
    if (rbtnYes.Checked)
    {
        lblDrugs.Show();
        txtDrugs.Show();
    }
    else
    {
        lblDrugs.Hide();
        txtDrugs.Hide();
    }

    if (rbtnNo.Checked)
    {
        lblDrugs.Hide();
        txtDrugs.Hide();
    }
    else
    {
    
    }
}

private void rbtnNo_CheckedChanged(object sender, EventArgs e)
{
    if (rbtnYes.Checked)
    {
        lblDrugs.Show();
        txtDrugs.Show();
    }
    else
    {
        lblDrugs.Hide();
        txtDrugs.Hide();
    }

    if (rbtnNo.Checked)
    {
        lblDrugs.Hide();
        txtDrugs.Hide();
    }
    else
    {
    
    }
}

private void chkUpdate_CheckedChanged(object sender, EventArgs e)
{
    if (chkUpdate.Checked)
    {
        groupBox6.Show();
    }
    else
    {
    
    }
}
private void btnSaveBiodata_Click(object sender, EventArgs e)
{
    if (rbtnYes.Checked)
    {
        DBAccess db = new DBAccess();
        db.AddParameter("@RegistrationNo", txtRegNo.Text);
        db.AddParameter("@Surname", txtSurname.Text.ToUpper());
        db.AddParameter("@FirstName", txtFirstName.Text);
        db.AddParameter("@LastName", txtLastName.Text);
        db.AddParameter("@Drugs", txtDrugs.Text);
        db.AddParameter("@Age", cmbAge.SelectedItem.ToString());
        db.AddParameter("@DateOfBirth", DateTime.Parse(dtpDOB.Text));
        db.AddParameter("@StateOfOrigin", cmbStates.SelectedItem.ToString());
        db.AddParameter("@Religion", cmbReligion.SelectedItem.ToString());
        db.AddParameter("@ContactAddress", txtContactAddrs.Text);
        db.AddParameter("@SchoolAttended", txtSchAttended.Text);
        db.AddParameter("@Guardian", txtNames.Text);
        db.AddParameter("@Address", txtAddress.Text);
        db.AddParameter("@PhoneNo", txtPhoneNo.Text);
        db.AddParameter("@Email", txtEmail.Text);
        db.AddParameter("@Genotype", cmbGenoType.SelectedItem.ToString());
        db.AddParameter("@Gender", cmbGender.SelectedItem.ToString());
        db.AddParameter("@DateAdded", DateTime.Now);
        db.AddParameter("@ClassOfAdmission", txtClassOfAdmin.Text);
        db.AddParameter("@CurrentClass", txtCurrentClass.Text);
        db.AddParameter("@Term", cmbTerm.SelectedItem.ToString());
        int i = Convert.ToInt32(db.ExecuteNonQuery("SMS_Student_Insert"));
        if (i > 0)
        {
            MessageBox.Show("Student's Biodata was successfully saved.", "Student Biodata Alert!", MessageBoxButtons.OK, MessageBoxIcon.Information);
            txtRegNo.Clear();
            txtSurname.Clear();
            txtFirstName.Clear();
            txtLastName.Clear();
            cmbAge.Text = null;
            cmbStates.Text = null;
            cmbReligion.Text = null;
            txtContactAddrs.Clear();
            txtSchAttended.Clear();
            txtNames.Clear();
            txtAddress.Clear();
            txtPhoneNo.Clear();
            txtEmail.Clear();
            cmbBloodGroup.Text = null;
        }
    }
}
cmbGenoType.Text = null;
txtClassOfAdmin.Clear();
txtCurrentClass.Clear();
cmbterm.Text = null;
//txtTerm.Clear();
cmbGender.Text = null;
txtDrugs.Clear();
}
else
{
    MessageBox.Show("Student's Biodata was successful.", "Student Biodata Alert!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtRegNo.Clear();
txtSurname.Clear();
txtFirstName.Clear();
txtLastName.Clear();
cmbAge.Text = null;
cmbStates.Text = null;
cmbReligion.Text = null;
txtContactAddrs.Clear();
txtSchAttended.Clear();
txtNames.Clear();
txtAddress.Clear();
txtPhoneNo.Clear();
txtEmail.Clear();
cmbBloodGroup.Text = null;
cmbGenoType.Text = null;
txtClassOfAdmin.Clear();
txtCurrentClass.Clear();
// txtTerm.Clear();
cmbterm.Text = null;
txtDrugs.Clear();
cmbGender.Text = null;
}
else
{
    DBAccess dba = new DBAccess();
dba.AddParameter("@RegistrationNo", txtRegNo.Text);
dba.AddParameter("@Surname", txtSurname.Text.ToUpper());
dba.AddParameter("@FirstName", txtFirstName.Text);
dba.AddParameter("@LastName", txtLastName.Text);
dba.AddParameter("@Drugs", rbtnNo.Text);
dba.AddParameter("@Age", cmbAge.SelectedItem.ToString());
dba.AddParameter("@DateOfBirth", DateTime.Parse(dtpDOB.Text));
dba.AddParameter("@StateOfOrigin", cmbStates.SelectedItem.ToString());
dba.AddParameter("@Religion", cmbReligion.SelectedItem.ToString());
dba.AddParameter("@ContactAddress", txtContactAddrs.Text);
dba.AddParameter("@SchoolAttended", txtSchAttended.Text);
dba.AddParameter("@Guardian", txtNames.Text);
dba.AddParameter("@Address", txtAddress.Text);
dba.AddParameter("@PhoneNumber", txtPhoneNo.Text);
dba.AddParameter("@Email", txtEmail.Text);
dba.AddParameter("@Genotype", cmbGenoType.SelectedItem.ToString());
dba.AddParameter("@Gender", cmbGender.SelectedItem.ToString());
dba.AddParameter("@DateAdded", DateTime.Now);
dba.AddParameter("@ClassOfAdmission", txtClassOfAdmin.Text);
dba.AddParameter("@CurrentClass", txtCurrentClass.Text);
dba.AddParameter("@Term", cmbterm.SelectedItem.ToString());
int i = Convert.ToInt32(dba.ExecuteNonQuery("SMS_Student_Insert"));
if (i > 0)
{
    MessageBox.Show("Student's Biodata was successfully saved.", "Student Biodata Alert!", MessageBoxButtons.OK, MessageBoxIcon.Information);
    txtRegNo.Clear();
    txtSurname.Clear();
    txtFirstName.Clear();
    txtLastName.Clear();
    cmbAge.Text = null;
    cmbStates.Text = null;
    cmbReligion.Text = null;
    txtContactAddrs.Clear();
    txtSchAttended.Clear();
    txtNames.Clear();
    txtAddress.Clear();
    txtPhoneNo.Clear();
    txtEmail.Clear();
    cmbBloodGroup.Text = null;
    cmbGenoType.Text = null;
    txtClassOfAdmin.Clear();
    txtCurrentClass.Clear();
    cmbterm.Text = null;
    cmbGender.Text = null;
}
else
{
    MessageBox.Show("Student's Biodata was unsuccessful.", "Student Biodata Error!", MessageBoxButtons.RetryCancel, MessageBoxIcon.Error);
    txtRegNo.Clear();
    txtSurname.Clear();
    txtFirstName.Clear();
    txtLastName.Clear();
    cmbAge.Text = null;
    cmbStates.Text = null;
    cmbReligion.Text = null;
    txtContactAddrs.Clear();
    txtSchAttended.Clear();
    txtNames.Clear();
    txtAddress.Clear();
    txtPhoneNo.Clear();
    txtEmail.Clear();
    cmbBloodGroup.Text = null;
    cmbGenoType.Text = null;
private void groupBox1_Enter(object sender, EventArgs e)
{
}

private void btnSaveChanges_Click(object sender, EventArgs e)
{
    DBAccess db = new DBAccess();
    db.AddParameter("@RegistrationNo", cmbStudentNames.SelectedValue.ToString());
    db.AddParameter("@Age", txtAge.Text);
    db.AddParameter("@ContactAddress", txtUpContAdd.Text);
    db.AddParameter("@Address", txtGuardAddrs.Text);
    db.AddParameter("@PhoneNo", txtUpPhoneNo.Text);
    db.AddParameter("@Email", txtUpEmail.Text);
    db.AddParameter("@CurrentClass", txtUpCurrentCls.Text);
    db.AddParameter("@Term", txtupterm.Text);
    int i = Convert.ToInt32(db.ExecuteNonQuery("SMS_Student_Update"));
    if (i > 0)
    {
        MessageBox.Show("Student's Biodata update was successful.", "Update Alert!", MessageBoxButtons.OK, MessageBoxIcon.Information);
        cmbStudentNames.Text = null;
        txtAge.Clear();
        txtUpContAdd.Clear();
        txtGuardAddrs.Clear();
        txtUpPhoneNo.Clear();
        txtUpEmail.Clear();
        txtUpCurrentCls.Clear();
        txtupterm.Clear();
    }
    else
    {
        MessageBox.Show("Student's Biodata update was successful.", "Update Alert!", MessageBoxButtons.RetryCancel, MessageBoxIcon.Information);
        cmbStudentNames.Text = null;
        txtAge.Clear();
        txtUpContAdd.Clear();
        txtGuardAddrs.Clear();
        txtUpPhoneNo.Clear();
        txtUpEmail.Clear();
        txtUpCurrentCls.Clear();
    }
}
private void cmbStudentNames_SelectedIndexChanged(object sender, EventArgs e)
{
    if (cmbStudentNames.SelectedValue != null)
    {
        DBAccess db = new DBAccess();
        db.AddParameter("@RegistrationNo", cmbStudentNames.SelectedValue.ToString());
        txtAge.Text = db.ExecuteScalar("getAge").ToString();
        txtUpContAdd.Text = db.ExecuteScalar("getContactAddress").ToString();
        txtGuardAddrs.Text = db.ExecuteScalar("getAddress").ToString();
        txtUpPhoneNo.Text = db.ExecuteScalar("getPhoneNo").ToString();
        txtUpEmail.Text = db.ExecuteScalar("getEmail").ToString();
        txtUpCurrentCls.Text = db.ExecuteScalar("getCurrentClass").ToString();
        txtupterm.Text = db.ExecuteScalar("getTerm").ToString();
    }
    else
    {
        //do nothing!
    }
}