

# Object Oriented Database Management System: A UML Design Approach.

Uduak A. Umoh, M.Sc.<sup>1</sup>, Enoch O. Nwachukwu, Ph.D.<sup>2</sup>, Imo J. Eyoh, M.Sc.<sup>1</sup>,  
and Augustine A. Umoh, Ph.D.<sup>3</sup>

<sup>1</sup>Department of Mathematics, Statistics, and Computer Science, University of Uyo, Nigeria.

<sup>2</sup>Department of Computer Science, University of Port Harcourt, Nigeria.

<sup>3</sup>Department of Physics, University of Uyo, Nigeria.

E-mail: [uduakaustynumoh@yahoo.com](mailto:uduakaustynumoh@yahoo.com)

## ABSTRACT

In this paper, we adopt a state-of-the-art technology to design an Object-Oriented Database Management System (OODBMS) for the management of information in Nigerian Universities. We apply the object-oriented design tools to create a database model that is reliable, dependable and secure. We focus on creating a unified modeling language (UML) structure by specifying the use case, classes, and activities in the client-server application. The design strategies of the web-based OODBMS are also discussed and a server-based application with state-of-the-art facilities expected. We develop a working prototype of the system on a three-tier client server architecture based on the advantages discussed in the work MySQL DBMS, Apache web server, and PHP server pages are adopted as system development tools. We apply existing client-server and database technologies to provide an object-oriented web-based, data-retrieval system for the Universities that will enable access by the various users at all times.

(Keywords: object-oriented database management system, OODBMS, unified modeling language, UML, university data, data retrieval)

## INTRODUCTION

Object-Oriented Databases join the capabilities of object-oriented language with the traditional relational database management functions. Object-Oriented Database Systems (OODS) were limited in the past because of a lack of standardization. As a result of its present standardization, the relational database management system has offered high performance to its numerous users. This is why the Object Data Management group has put

forward a set of standards for object oriented Database [1].

Currently, most academic institutions in Nigeria use information systems based on a database to handle the problem of administration overload. The bulk of the problem has to do with the management and security of staff and student data. However, because of the system architecture, only administrative office personnel have access to the information. This could cause information insecurity and wastage of computer resources, manpower, and time. There is no reliable web-based database management information system for Nigerian Universities currently in use.

Our objective is to explore the facilities that are currently available in object-oriented technology and design a secure and reliable object-oriented web-based database management system (WBOODBMS) for Nigerian Universities otherwise called secure student information system (SSIS). We describe the object data model and the system design showing the interaction between the different system components with a three-tier SSIS object model. The system if implemented will provide information accessibility to a wide range of audience, adaptability of design, reliable and dependable data management facilities, and most of all security.

Object-Oriented Analysis (OOA) and Design (OOD) is an approach to software development that involves creating classes and object which model real-world entities, their data and the operation that take place in that data [2]. OOD has emerged as a software blueprint language for analyst, designers and programmers alike. It constructs a model which is an abstraction of underlying problem. The domain is the real world

from which the problem comes. Classes are the specification for object. A class wraps attributes (data) and behaviors (methods) into a single distinct entity [4].

This paper describes the problem of administration overload which, has to do with the management and security of staff and student data in Nigeria Universities. This paper proposes an object oriented web-based database management tool, otherwise called SSIS, to support editing, and management of educational information for Nigeria Universities. SSIS aims to facilitate the accessibility and retrieval of educational data by the users anytime and anywhere.

## THE OBJECT DATA MODEL

This section describes the object models. The object model specifies the constructs that are supported by an OODBMS. The basic modeling primitive are the objects and the literal. Each object has unique identifiers while a literal has none. The state of an object is defined by the values it carries for a set of properties. These properties can be attributes of object itself or relationships between this object and one more other object. The behavior of an object is defined by the set of operations that can be executed on or by the object. A database stores object and is based on a schema that is defined in ODL and contains instances of the types (Classes), defined by its schema [5, 13, 14]. This SSIS object model specifies what is meant by object literals, types, operations, properties attributes, relationships, etc. The constructs of the model and their specifications are described as follows:

### (i) Types (Classes)

There are two aspects of class definition; Implementation and interface. In the implementation aspect, the internal objects of the type are defined. This consists of a set of methods which is the body of operation as defined by the interface. The external characteristic of the objects of the type are defined by the interface which constitute the attributes and are explored to describe the state of each object of the type and the operations that can be invoked on each object of the type. Encapsulation is *reflected* by the separation between implementation and interface [2, 3].

SSIS object model supports inheritance relationships between classes: for example, a class, called the sub-class inherits the same interface of another class, called the super-class. A type may have an extent which is a set of all its instances (objects). It may also contain one or more keys which uniquely identify a specific object.

### (ii) Objects

An object is an instance of a class. An object-oriented model is built around objects. An identifier is used to differentiate it from other objects.

### (iii) State and Behavior

The state of an object is determined by the value of its properties. This property can either be attribute or a relationship. Attributes are defined by their names and types. A relationship is defined between two types that an object of one type refers to one or more objects of the other type. The behavior of an object is specified as a set of operation signature. Each defines the operation name, the name and the type of each argument and the return value type. An object is defined by a set of attributes and a set of relationships and a set of methods [6, 7].

**Illustration:** A Class diagram of the unified modeling language (UML) notation in Figure 2 is used to show an object-oriented database schema. It contains five classes; "student", "Personal Data", "Department", "Course", and "Result". "Student" is the central class, "Department" is a subclass of "Student" and an association relationship is established from "Department" to "Student". There may exist a 'bidirectional relationship between student' and "Department". This shows that a "Department" instance has many "Student" instances and the other indicates that a "Student" belongs to one "Department".

## SSIS SYSTEM DESIGN

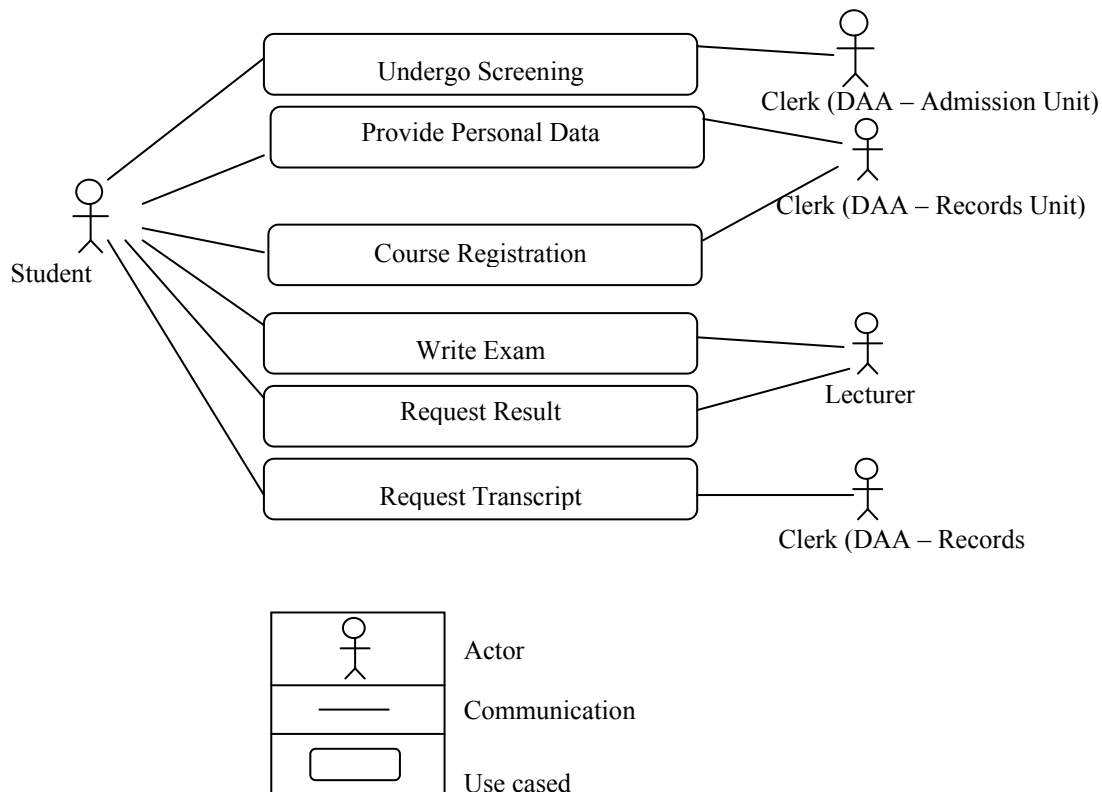
UML design tools and notations are employed to graphically depict object-oriented analysis and design models. UML is a language for specifying, visualizing and constructing the artifacts of s/w

systems, as well as for business modeling. It shows the interactions and relationships between its different classes and components. In this section we describe the SSIS system design by showing the interactions and relationships between the system's classes and components. As in any DBMS, there are multiple levels of users of the system. The following object-oriented design tools are explored to model the students' information system.

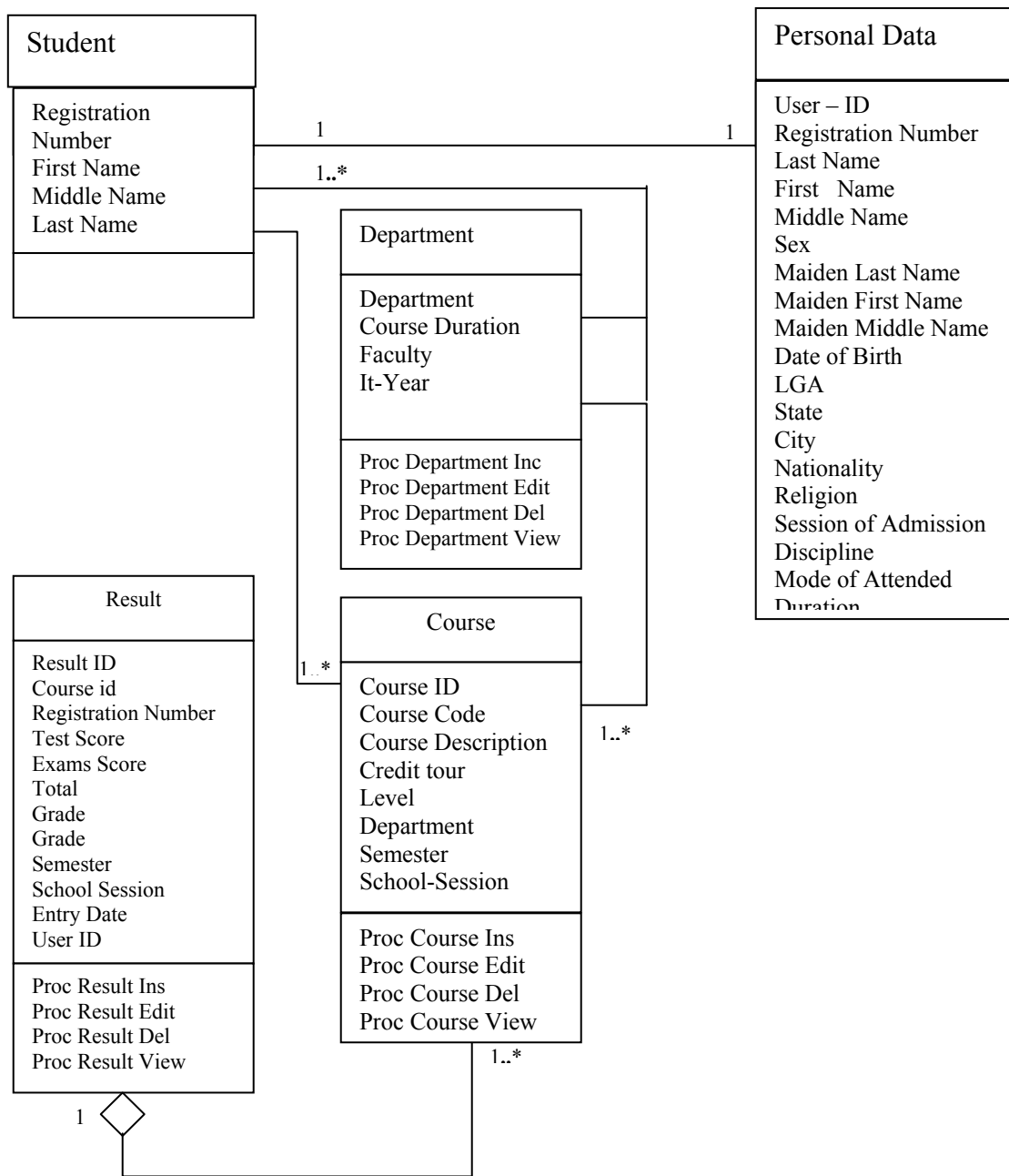
**Use Case:** The use case technique is adapted in this work to handle all objects that are needed to be present in the application as shown in Figure 1. It helps to determine system requirements, communication with clients and to generate test cases. From the use case diagram, the following objects to be modeled are identified: student, personal data, course, results, and transcripts.

**Class Diagram for SSIS:** SSIS class diagram in Figure 2 gives the overview of the system. It shows the different classes and the relationships amongst the system components. This is represented using a single class model and a three-tier object model. This is because a single class model is static that is, it displays what interacts but not what happens when the different system components interact. Our Class diagram models the student information system. The central class is the student, associated with it are personal data of the student, department the student belong, courses offered by the students and the results class respectively. The UML class notation is applied with a rectangular box which is divided into three parts; name, attributes, and operation, [12, 13].

Table 1 shows the multiplicity of an association end in SSIS class diagram, it shows the number of possible instances of the class associated with a single instance of the other end.



**Figure 1:** SSIS Use Case Diagram.



**Figure 2: SSIS Class Diagram.**

**Table 1: Multiplicities of Association for the Class Diagram.**

Multiplicities	Meaning
0..1	Zero or one instance.
0..* or *	No limit on the number of instances (including none).
1	Exactly one instance
1..*	At least one instance
◇	Aggregation

**Activity Diagrams:** Figures 3, 4, 5, and 6 show SSIS activity diagrams. These diagrams focus on the flow of activities involved in the four different processes. These activities depend on one another in each of the processes. For example, in the screening process, the three classes of activity involved are; 1) User, 2) Application/Web server, and 3) Database server. The process begins at the black start circle at the top and ends at the concentric white/black stop circle at the bottom. The ovals show the different activities that are carried out in SSIS while the hollow diamonds indicate a branch and its subsequent merge which mark the end of the branch.

### THREE-TIER SSIS OBJECT MODEL

Client/server model “entails two autonomous processes working together over a network. The client processes request specific services which the server processes respond to and process” [10]. The majority of end-user applications consist of three components: presentation, processing, and data. Client/server model can be static, two-tier or three-tier. A single class model is static that is, it displays what interacts but not what happens when the different system components interact.

In a two-tier model, the PC client assumes the bulk of the responsibility for the application (functionality) logic with respect to the processing component while the database engine handles the data intensive task. Although a two-tier environment has application development speed advantage, it faces a number of potential version-control and application-redistribution problems since the bulk of the application logic exists on the PC client.

The three-tier architecture attempts to overcome some of the limitations of the two-tier scheme by separating the business application into three logical components: presentation, application logic, and data management. These logical components are “clean layered” in such a manner that each runs on a different machine or platform and communicates with the other components via the network [7]. Three-tier model allows for openness, parallel development of each tier, scalability, middle-tire code reuse and transparency [9, 10].

In our work, we employ three-tier architecture to develop SSIS object model as shown in Figure 7.

The UML diagram is split into logical tiers; User services, Business services and Data services. The SSIS Object Model is developed according to the use case defined in Figure 1.

The basic objects in the system are defined and the interaction between the objects determined. It is a simplified object model of the system with three distinct parts (Application/Business, User, and Data Services), joined together to achieve good design and extensibility, [7, 9, 12]. The model if implemented on a network is capable of permitting access by different categories of users’ (e.g. administrators, students, academic staff, etc.) diagram in this work as shown in Figure 8 to describe SSIS code module and the physical configurations of the software and hardware. A component diagram gives the physical analogy of SSIS class diagram while a deployment diagram shows the relationships among software and hardware components involved in the system transaction.

### SSIS DATABASE

In our present work, we select the main entities for the database based on the use-cases earlier defined and develop the relations between entities as shown in Figure 9.

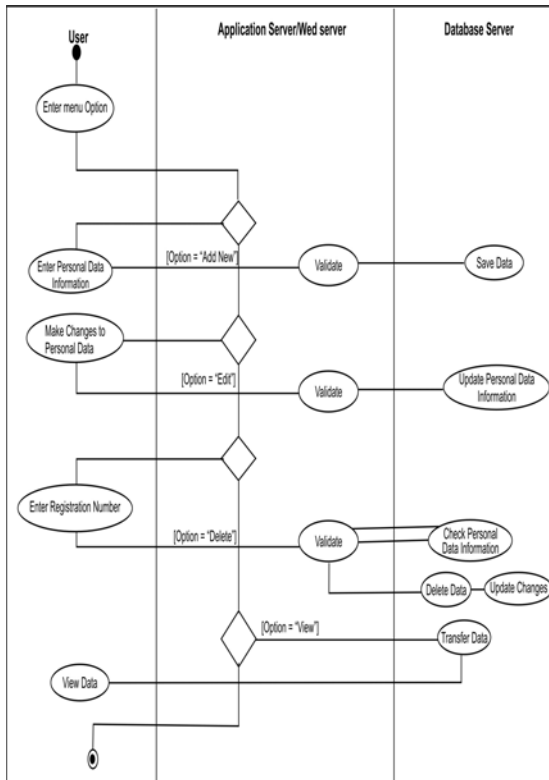
### SYSTEM IMPLEMENTATION

In this section, we develop and implement the SSIS Web database and application prototype around the three-tier model earlier designed.

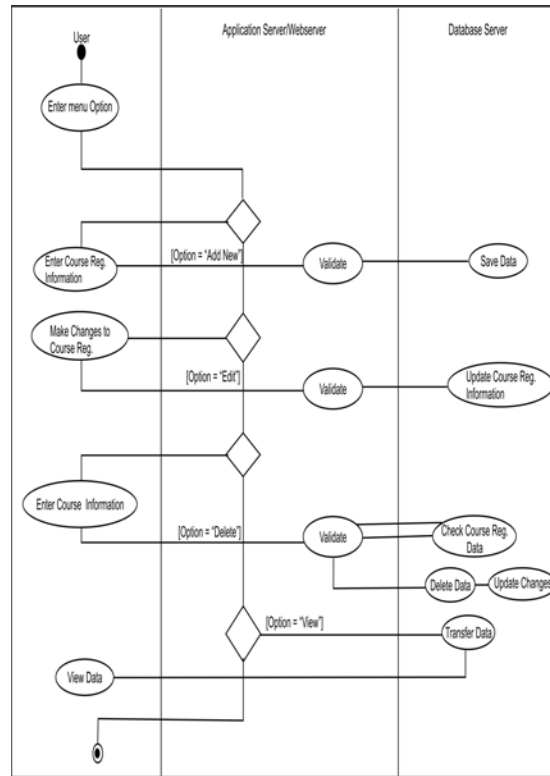
**SSIS Database Implementation:** Data Services Tier is implemented within a relational database management system (MySQL Server). A MySQL Server database (named SSIS) handles all data-based operations (creation, storage, retrieval, maintenance and integrity checks) on this tier. This database consists of tables and relations. After creating the tables and relations, the database is populated with data for the purpose of implementation.

**SSIS Application Implementation:** We implement our Secure Student Information System (SSIS) Web application based on the following main modules:

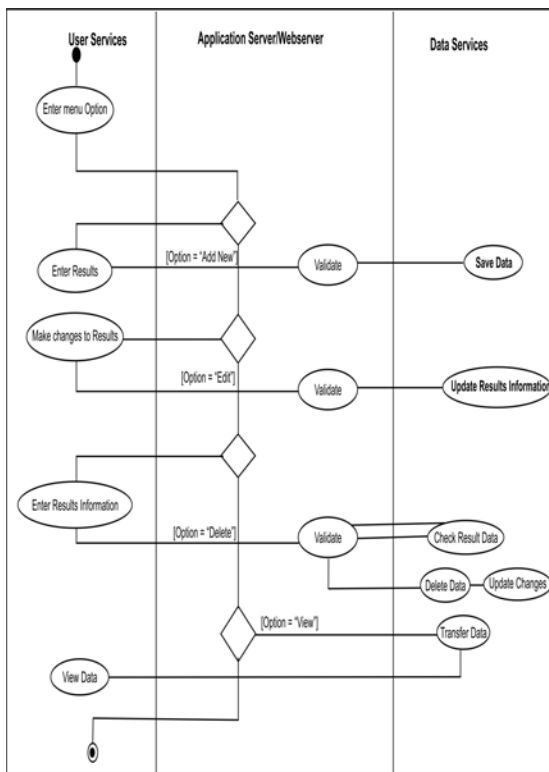
- (i) The Administration Module
- (ii) The Registration Module
- (iii) The Reports Module



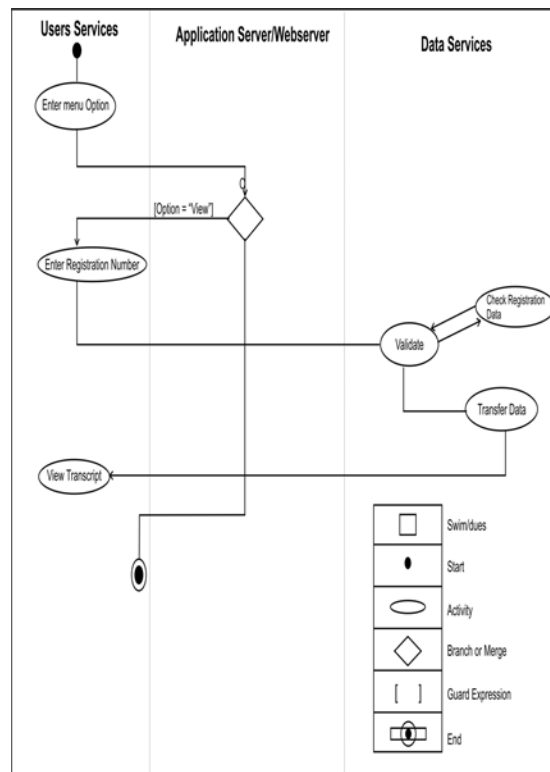
**Figure 3:** Activity Chart for Personal Data based Operation.



**Figure 4:** Activity Chart for Course Registration Data based Operation.



**Figure 5:** Activity Chart for Result Data based Operation.



**Figure 6:** Activity Chart for Transcript based Operation.

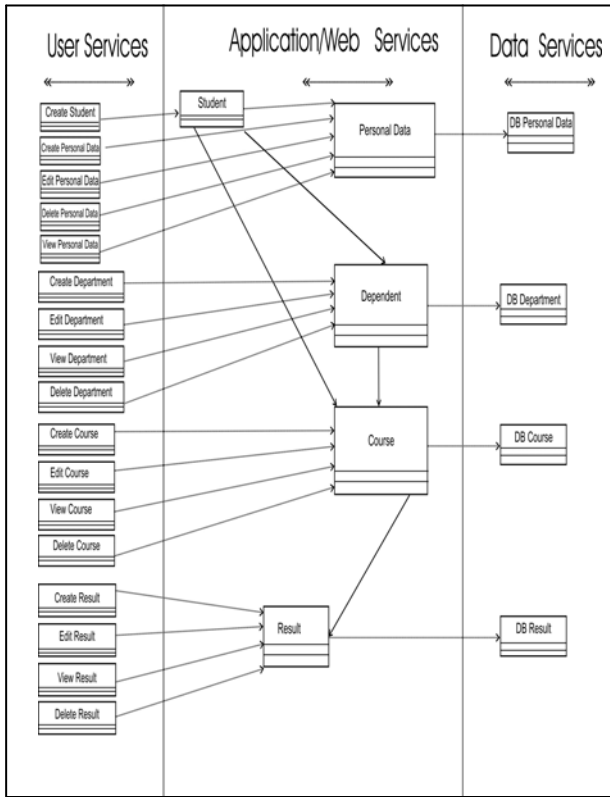


Figure 7: Three-Tier SSIS Model

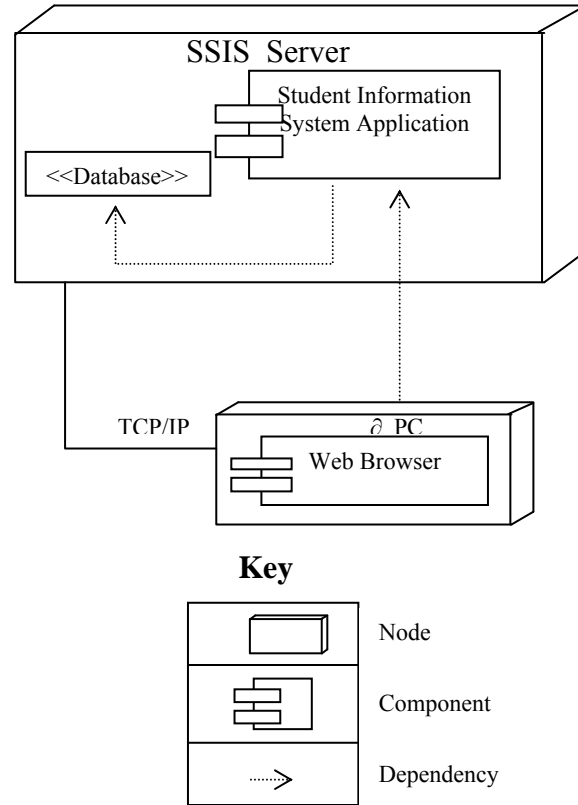


Figure 8: SSIS Component and Deployment.

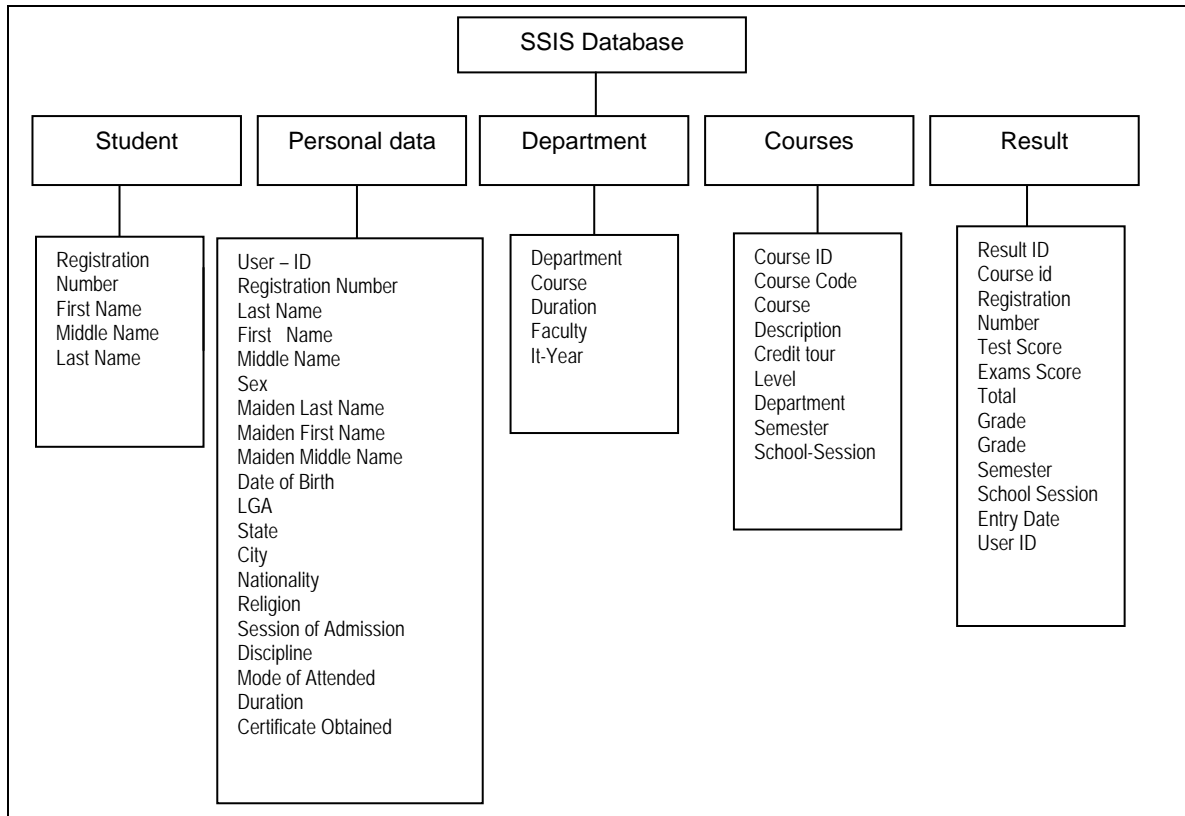


Figure 9: SISS Database.

The discussion of each module is sub – divided into Task performed. The task represents what the module does with respect to the program. The Administrative module coordinates all data based operations (add, edit, delete and view) and web page functionalities for user profiles, departments, screening, courses, and results. The Registration Module handles all data-oriented operations (add, edit, delete and view) and web page functions for student personal data and course registration. While the Reports Module controls all report generation operations and web page functionalities.

To accomplish these tasks, the necessary menu options are given and the authorized users are provided access to the system based on their privileges and responsibilities.

## DESIGN CAPABILITIES

We employ the following design tools to develop our system;

- a. Macromedia Dreamweaver MX 2004 (Web Development Tool) is used under Windows XP with VMWare Workstation Version 4.0 (Virtual Machine software).
- b. Apache version 2.0 is employed as web server, PHP 4.2.2 is used as web scripting language and MySQL version 3.23.54 is employed as database Server which, run on the Red Hat Linux operating system

The system is dynamic, robust (Portability), user friendly, maintainable and expandable.

## SCREEN SHOTS OF SSIS

In this section, we presented some of the screen shots of the system. The similar functions and pages are not repeated for each type of user. Figure 10 shows the application's Login Page. This is the only entry point for the system by all users through the supply of identification password.



Figure 10: User Login Page.

Once the user authenticates the application successfully, the Home Page is loaded within the browser as shown in Figure 11.

Figure 12 shows the administration home page which allowed permissible administrative procedures to be carried out.

Figure 13 displays student registration page where students' permissible procedures are carried out.

Figure 14 shows the report page where its content is highly customized, meaning that the ADMIN and STAFF groups have a different view of the page from the STUDENT group.



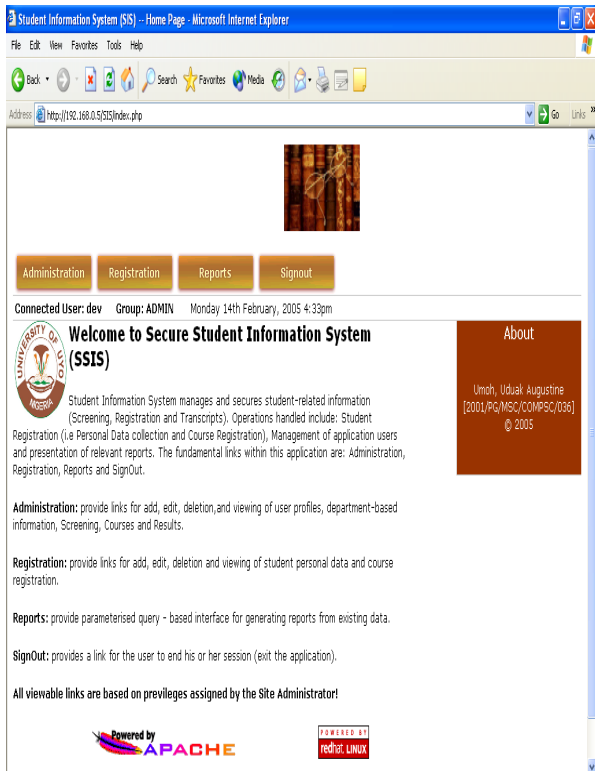


Figure 11: The SSIS Home Page.



Figure 13: The Registration Page.



Figure 12: Administration Home Page

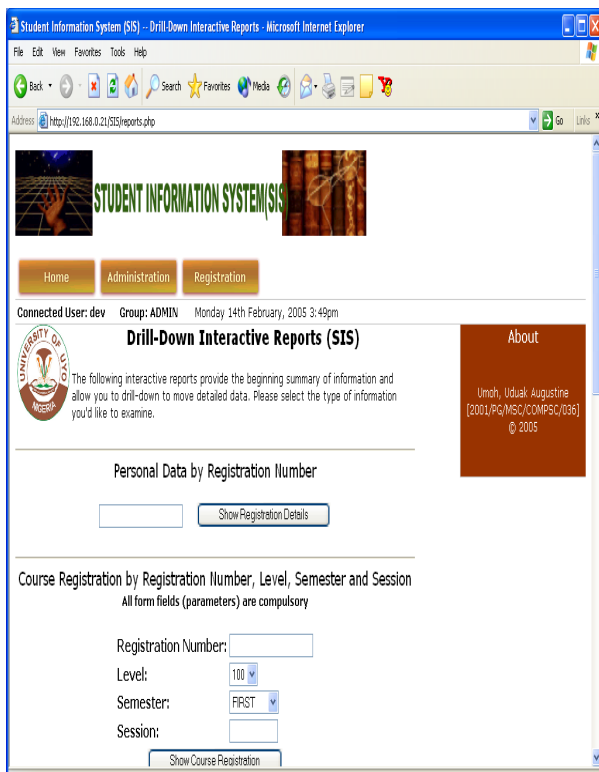


Figure 14: The Reports Page Viewed by a Member of the ADMIN Group.

## CONCLUSION

In this paper, we employ object-oriented design tools to design, develop, and implement an object-oriented, web-based database management system for Nigerian Universities. We illustrate the variant design techniques and components. We also present a three-tier Secure Student Information Object Model for Nigerian Universities which will help to alleviate the problems of information overload, insecurity, and wastage of computer resources, manpower, and time associated with the existing manual system.

In the future, more user types can be added to the system for the implementation purposes. To facilitate accurate information from institutions, users system interfaces can be defined separately in the requirement for the faculty and department. Special features can be incorporated into the system to enable the users to print the academic official records or reports. According to the format of those reports, and with additional requirements, these user types can be added to the system. Another type of user for future enhancements is the system administrator who can be given the capability to see the current online users, their activities, and to perform any online support maintenance.

## REFERENCES

1. RGG Catell. 1996. *The object Database Standards: ODMA-93(Release 1.2)*. Morgan Kaufmann Publishers, Inc.: San Francisco, CA.
2. Bertino, F. and Martino, L. 1994. *Object-Oriented Database System*. Addison Wesley Publishing Company: New York, NY.
3. Amini, A. and Ghada, M. 1997. "Badr.The Alex Object Manager". In: Proceedings of ISCC'97, Second IEEE Symposium on Computer and Communications. Alexandria, Egypt. 2000-2004.
4. Elheky, M.G. and G.M.Badr. 1997. "Alex Object-Oriented Database Management System". Computer Science and Automobile Control Department, Faculty of Engineering, Alexandri University. Technical Report 1997.
5. Kueng, P. 1994. "Comparison of Tier OODBMSs". *The Swiss Computer Magazine OUTPUT*. 60-63 June 1994.
6. Elmasri, R. and Navathe. S. 1994. *Fundamental of Database System, Second edition*. Addison, Wesley: New York, NY.
7. Edwards, J. 1999. *Three Tier Chart/Server at Work. Revised Edition*. John Wiley and Sons: New York, NY.
8. Ramakrishnan, R. and Gehrke, J. 2000. *Database Management System. Second Ed*. McGraw Hill, New York, NY.
9. Orfali, R.H. and Edwards, J. 1996. *The Essential Client/Server Survival Guide*. John Wiley & Sons: New York, NY.
10. Pressman, R.S. 2001. *Software Engineering a Practitioner's Approach, Fifth Edition*. McGraw Hill, New York, NY.
11. Neubauer, B.J. 2002. "Data Modeling in the Undergraduate Database Course: Adding UML and XML Modeling to the Traditional Course Content." *Journal of Computing in Small Colleges*. 17(5):147-153.
12. Siau, Keng and Qing Cao. 2001. "Unified Modeling Language (UML): A Complexity Analysis." *Journal of Database Management*. 12(1):26-34.
13. Urban, S.D. and S.W. Dietrich. 2003. "Using UML Class Diagrams for a Comparative Analysis of Relational, Object-Oriented, and Object-Relational Database Mappings." *ACM SIGCSE Bulletin*. 35(1):21-25.
14. Wagner, P. 2005. "Teaching Data Modeling: Process and Patterns." Paper presented at the Proceedings of the 10th annual SIGCSE Conference on Innovation and Technology in Computer Science Education. Capacrica, Portugal, June 2005.

## ABOUT THE AUTHORS

**Uduak A. Umoh** holds an M.Sc. degree in Computer Science from University of Port Harcourt, Nigeria. She is currently a Ph.D. student in Computer Science, University of Port Harcourt, Nigeria. Her area of research is Neural Networks and Fuzzy Logic.


**Enoch O. Nwachukwu** is a Professor in the Department of Computer Science, University of Port Harcourt, Nigeria. He holds an M.Sc. degree in Computer Science from University of Manchester, England and a Ph.D. in Computer Science from University of Manchester, England. His major research interest is Software Engineering, Networks, and Expert Systems.

**Imoh J. Eyo** is a Lecturer in Computer Science Department, University of Uyo, Nigeria. She holds an M.Sc. degree in Computer Science from University of Ibadan, Nigeria.

**Augustine A. Umoh** is a Lecturer in Geophysics/Physics Department, University of Uyo, Nigeria. He holds an M.Sc. in Physics from the University of Ilorin, Nigeria, and a Ph.D. in Communications Physics from University of Ilorin, Nigeria.

#### **SUGGESTED CITATION**

Umoh, U.A., E.O. Nwachukwu, I.J. Eyoh, and A.A. Umoh. 2009. "Object Oriented Database Management System: A UML Design Approach". *Pacific Journal of Science and Technology*. 10(2):355-365.

 [Pacific Journal of Science and Technology](http://www.akamaiuniversity.us/PJST.htm)