Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 4 Issue 10, October - 2017

Automated Students' Results Management Information System (SRMIS)

UDEZE, CHINEDU L.[1]; UMOREN, PAUL U.[2]; OHERI, HENRY E.[3]; and ATTAH HONESTY H. [4]
Department of Computer Science, University of Calabar, Calabar, Cross River State, Nigeria.
1: udezechinedu@unical.edu.ng ; 2: umorenpaul@gmail.com ; 3: henroky@gmail.com ;
4: honesty4holiness@gmail.com.

***Abstract*—This project, Automated Students' Results Management Information System (SRMIS) was carried out to automate the manual processes of compiling Students Examination Results. It was necessitated because of some setbacks in manual result processing. The system was designed to automatically take raw scores from excel files and store them in a database. It used past processed results to help the next course registration prior to results upload. Its result processing features includes the computation of grade point average (GPA), generation of result reporting sheets and transcripts. Every session, it keeps track of student's status information as recorded in the student files, specifying if a student is legitimate. The database also holds the lists of admitted students each year and records their school fees payment status. The software engineering was done with the Incremental model using an object-oriented programming approach. Raw data input to the SRMIS is one of the most cumbersome tasks. A computerized input using file upload saves lecturers a lot of effort and time of data entry. This system uses the student's course registration data to match the uploaded results. The essence is to design an efficient computerized system that will replace manual result processing which is prone to lot of paper work and errors. This reduces the tedious tasks involved, and enhances students' performance through timely publication of results.**

Keywords—Cumulative Grade Point Average (CGPA), Transcript, Modeling, Database, Result Reporting Sheet, Process, Online Data Entry, Unified Modeling Language (UML), Entity-Relationship Diagram (ERD).

1. INTRODUCTION

Okonigene, Ighalo, and Ogbeifun (2008) observed that, the use of computers for information processing facilitates instant access to students' personal and course information, instant student information updating and monitoring of failed courses. For system support, storing course information such as course codes, course descriptions, credit units, and grade points for the purpose of automatic computation of the GPA are required. The system keeps up-to-date records of the entire student body in the Department and generates the graduating students list. To support procedures like registration, data upload, queries to the system friendly graphical user interfaces were developed for the ease of use. All these have been noted in the database design and implementation of this system.

As the first step of software engineering process of system design we started with interviewing lecturers and students, and compiled our observations into a case study of the state of manual student result preparation and data handling which served as the major issues addressed in the design. The following challenges were identified in the students' data processing in the Department. They include: delays in preparation and release of Senate-approved results to students to view their performance on time (on semester/session basis); resolve duplicate records/results; excessive paperwork in Results Processing; poor record-keeping of students status information; poor data and security management of students records and files; the unavailability of transcripts on demand. There is also the problem of poor availability of information to guide students during course registration which could result to presumptions in offering and dropping courses. The manual system leads to errors and omissions in CGPA computation, late preparation of students' final result for convocation and lack of an updated database of students' information for reference purposes.

"An Information System (IS) can be any organized combination of people, hardware, software, communications networks, data resources, policies and procedures that stores, retrieves, transforms and disseminates information in an organization" (O'Brien and Marakas, 2008). Hence, an MIS provides information in the form of predefined reports that support the decision-making processes of the managerial users. In an academic community, the automated SRMIS is intended to generate reports about students' results to guide the day-to-day decisions of the school management in the maintenance of student status and awarding of degrees.

The major aim of this research is to develop an automated SRMIS that can save the time involved in result processing, that is, CGPA computation in University of Calabar. This will permit easy and fast access to student information especially results and student status from a normalized database. When running in a network server, this can guide students during their program in the institution via computerized course registration with wireless devices. It can enable the enforcement of security measures through the allocation of access privileges; and check mischievous acts of mutilating scores on the result sheet.

2. LITERATURE REVIEW

Generating and organizing data in a useful way is called data processing (Royce, 1970). In his paper, Ukem (2012) stated that the errors associated with the existing manual method of processing of students results in most universities in Nigeria, make it not only desirable but imperative that computerized approach be used in measuring students progress. According to him, the manual methods being employed suffer a number of setbacks; they make the process to be time consuming and prone to error. They lead to examination results being published late, sometimes with wrong grades being entered and students GPAs being wrongly computed. This could lead to wrong conclusions in the awarding of class of degree. The solution to the problem, therefore, is to find a method of processing examination results that would be sufficiently accurate and reasonably timely. Welling (2007) stated that the principal means of inputting data to a computer system was on punch cards – the so called IBM cards that a generation of college students were admonished never to fold, spindle or mutilate.

Oxford dictionary defines data as "facts and statistics collected together for reference or analysis." Vossen (1991) in her book stated that, "in computing, data is raw facts that has to be translated into a form that is more convenient to move or process. Relative to today's computers and transmission media, data is information converted into binary digital form ". Mitra (2011) stated that the terms data, information and knowledge are frequently used for overlapping concepts. Beizer (1990) asserted that data is information, often in the form of facts or figure obtained from experiments and surveys, used as a basis for making calculations or drawing conclusion.

According to Juan (2004), data entry is the act of transcribing some form of information into another medium, usually through input into a computer program. Forms of data that people might transcribe include handwritten documents, information on spreadsheets, sequences of numbers, as well as computer code. Data entry requires hardware, and the proper design of input devices has received considerable attention, including concern for standardization of keyboard layouts. With regard to minimizing input actions, one guideline might be that a user should not have to enter the same data twice; even if it is sometimes forgotten. Quinn (2004) stated that the term Data Pipe-lining refers to the logistical problem of ensuring that the required data are at the required location at the right time.

Eludire (2011) observed that a number of problems associated with student academic record management include improper course registration, late release of student results, inaccuracy due to manual and tedious calculation and retrieval difficulties/inefficiency. According to him, the development of database concept is the answer to these problems, where the amount of redundant data is reduced and the possibility that data contained on a file might be inaccurate because they were never updated. Amar (2009)

indicated that publication of students' results in the manual system takes a very long time resulting in the students remaining idle for long. Sometimes the delay in declaration of result causes heavy losses to the students as generally they cannot join further studies, or appear in competitive exams and jobs because of the non-availability of certificates on time.

According to E. O. Ukem et al (2012), result systems can be improved upon by using back-up policies and audit trails. Although his solution was built with java and MySql database, it did not establish any evidence for processing results batched in files to increase data integrity and mitigate fraud. The research could not also address what happens to a student after s/he is on suspension, deferment of studies, medical treatment, supplementary exams or related cases. We have analyzed these scenarios and developed an audit trail for references.

Just like the solution by Akinmosin James (2014) of Nassarawa State University Keffi, the software has a login form for authentication of users and Student Registration forms for registering students every semester after payment of dues, before results are uploaded using internet browsers. His solution was built with the Oracle Procedural Language/ Structured Query Language (PL/SQL) and adopts forms and reports for its implementation. However, the interface widely uses "Grades form for inputting student grade and Grades Edit form for editing erroneously entered grades." This is a security threat and a loophole in the design that could aid intruders in accessing the database. It's a weakness that may not properly check the mutilation of grades by users who find a way to access those forms.

Similar to the solution by Idogho, Akpado and Agajo (2011) for Federal Polytechnic Auchi, the automated software works on several tiers with a browser at the front-end, a PHP engine, and a MySQL server at the back-end. Their system promised to reduce admission list processing to 24 hours using the PhpMyAdmin database management system. Nevertheless, the publication did not indicate the details of how the Student Exam Scores are entered into the system; whether by forms or file upload. Secondly, even though the use of Macromedia flash 8.0 and Dreamweaver 8.0 were pronounced, this particular paper on the software was not explicit on the normalization done to reduce redundancy in the database. This would have helped to predict the memory management capabilities of the program.

According to Bijoy, Sanjay, Bhibak, Nishal and Zarmit (2016), PHP is object-oriented, platform independent and portable on Microsoft Windows, Apple Macintosh, Linux etc. Based on such computerized systems, duplicity and loss of data is minimized as is found in manual procedures. This proposed system for the Centre of Computers and Communication Technology Chisopani, Sikkim, India can help faculties and students to track the top 10 students in the institute.

Efficient systems focus on some key goals. For instance, after due analysis, Duan and Zhang (2010) outlined some systems performance goals including usability, sophistication, integrity and security. Hence mass introduction of data via file uploads facilitates this measure; scheduling of data processing and data enquiries are functional means of achieving this.

Bharamagoudar, Geeta and Totad (2013) developed a web-based Student Information Management System in India which could send emails to students to validate their mailbox on registration. They were able to achieve this using technology such as HTML, CSS, Javascript, PHP and SQL. According to their description, it is a paperless work that assists in automating existing manual methods and can be remotely monitored and controlled on a server-based network.

Hemn and Wu Fei (2014) proposed a system in China that can provide students' general and educational information. According to them, the Students Information Management System (SIMS) can be used to create, read and update the details of a student and also generate reports about his/her skills and experience. Such systems save time of retrieval and prevent data loss.

Mariusz C. (2010) in his solution University Study-Oriented System (USOS) in Poland stated that the main functional parts are the admin, web, admission/registration of students, database of results, course and diploma catalog, statistics etc. According to him, this solution is used by 27 higher education Polish institutions. In such a system, before transferring any module for production use it has to pass through sample database and university test. Documentation

comprising system specification and implementation were updated regularly. Such solutions enhance communication between students and lecturers.

According to Ajay and Abhishek (2012), a good database does not permit anomalies and uses an orderly fashion to save relevant information for data integrity. Hence, for accuracy and ease of retrieval, tables have to be normalized. They also developed their solution using PHP embedded in HTML and linked to MySQL database with a PhpMyAdmin that can run on both WAMP and LAMP. PHP was preferred because of the ease of use on diverse platforms with minimal change to the script; and for its compilation speed and efficiency. In their system, they added an 'export' feature to reduce the bulk of time taken for individual student registration. It can enroll students in bulk from one level to another.

It is obtainable from an article on Wikipedia titled, Student Information System (SIS), that an SIS provides capabilities for course registration, grade documentation, transcript generation, students test results and assessment scores recording, students' schedules including disciplinary records, attendance monitoring and the overall management of student-related data in a school. It is not to be confused with a learning management system where materials for courses, assignments and tests can be published.

Bhatt et al (2016) proposed a Credit-based Grading Scheme (CBGS) in India. It is simply a PHP-MYSQL solution that can compile results of students. The reports of the system are generated in either excel or PDF format depending on the usage. Although the grading scheme differed from that of the Nigerian system, the formula for computing the Grade Point Average (GPA) is similar: the summation of the product of the credit hours and grade points divided by the summation of the credit hours. One major advantage of such automated systems over the manual procedure is the ease of searching and list generation.

Nmaju et al (2013) proposed a system Academic Records Information System (ARIS) at the University of Port Harcourt, Nigeria using the incremental software model and prototyping technique. It was also acceptable by their proposal that reporting sheets can only be generated when there are no pending scores for courses registered by students; which implies that the results of all such courses must have been approved by University Senate and uploaded on time.

In a publication by Charletta F. G. (2004) at North Illinois University USA, he noted a lawsuit filed against Microsoft by a lady in Los Angeles over security holes in the company's software. The plaintiff was a film maker, Marcy Hamilton who charged that because of shoddy workmanship by Microsoft, she had become a victim of identity theft. According to her, her Social Security Number (SSN) and bank information was stolen online. Hence, in this SRMIS, we have greatly considered issues of security breaches and have recently incorporated a log file and audit trail to help monitor activities that go on in the database. This will make the system more credible and enable the management account for changes made at any time by a particular user.

In terms of privacy, JISC InfoNet (2007), prescribed that digital records should be protected with passwords and electronic security measures. They maintained that institutions should avoid unauthorized disclosure of information in students' records. Hence, only staff members who need the information in their job description should be given access to it; and again with privileges only to the data that concerns them and not to the whole files. More so, since these records include personal data, according to the Data Protection Act 1998, the student as the data subject should be given access to the information whether in soft or hard copy.

Gunathilake et al (2009) proposed an open-source web-based MIS for the University of Ruhuna, Sri Lanka. This they were able to implement with the LAMP/WAMP technologies. They were able to categorize their users based on administrator, super admin, top admin, general, lecturer and student. The pilot version was targeted at their Faculty of Science and they achieved a password encryption with the primary DES algorithm.

According to a research article by Walia and Gill (2014), the objectives of creating a web-based framework for results processing are to abate the time needed to access students' record and provide a more secure platform. Over time, this

has proven to be a more effective way of university administration.
Based on the Guidelines for the Appraisal and Disposition of Student Records (2013), it requires that since this SRMIS contains some automatic deletion functionality, an archivist, administrator or competent specialist should be consulted before such actions are taken. Since these student records contains sensitive data such as academic performance, identification information, health and ethnic data, gender, religious and political data, some retention periods may be required – at least five years after graduation. It is expected that access to the data of living students and alumni should be with the permission of the affected persons and under certain conditions; and may not warrant to the disclosure of identity especially for research purposes. It is advisable that such records should be retained long enough to satisfy the needs such as statistical analysis for decision making, and legal interests of the institution. This publication also stated that when appraising students' records, instead of total destruction of data, copies can be backed-up in lower cost formats. To save cost of maintenance, after appraisal, data with no archival value can be disposed after legal retention periods in accordance with the legislature guiding data protection. Documentation should also keep details of disposed data.

3. SYSTEM DESIGN AND METHODOLOGY

The tools used for physical and database design are: Unified Modeling Language (UML) diagrams for description of the class diagram, and Entity-Relationship Diagrams (ERD) used for the design of the database table structures. The description covers the logical and database structural designs as well as its database normalization (McConnell, 2004). The entire web-based program is designed and developed with object-oriented codes, mainly with server-side PHP scripts.

3.1. System Requirement Analysis and Specification

The primary focus of system requirement analysis and specification is to understand in details what the user(s) really want the system to do. In order for us to actually know what the user needs, requirement analysis was conducted with the sole aim of gathering desirable features for the system.

3.1.1. Method of Data Collection

A thorough investigation of the current system was carried out in order to obtain detailed information about the application area to be designed. In the course of our findings, several effective methods of information gathering or data collection were employed which include: interviewing the examination officers; discussion with pertinent stakeholders of the system such as the HODs and lecturers; evaluation and inspection of relevant documents such as result reporting sheet, raw score samples, and transcript format.

1. Departmental Registration

a. New Student (UME/DIRECT ENTRY)

Firstly, a student whose name appears clearly in the approved list of admitted students dispatched from JAMB office to the department through the university authority should be allowed to proceed to the department for screening.
Secondly, upon the successful screening exercise conducted by the screening officer on each student, the student is permitted to register in the system by uploading his/her academic credentials and other important documents to the system, after obtaining matriculation number.
Thirdly, A student who fails to register during a session should not be allowed to write any examinations for that session and automatically implies voluntary withdrawal.

b. Returning student

Firstly, a student who has completely paid his/her department/faculty and student association charges should be allowed to access his/her accounts and to update his/her profile.
Secondly, s/he should also be permitted to access other functions of the system.

2. Course Registration

a. New student (UME)

Firstly, a student should be allowed to register his/her prescribed courses for the current semester including the GSS courses.
Secondly, a student should be allowed to view the course coordinator and the co-lecturer for every course registered per semester from the course allocation list uploaded to the system.

b. New student (Direct Entry)

Firstly, the system should of priority auto-register first, the GSS courses for each student.
Secondly, allow student to then register courses prescribed for the present level and semester in that session including the number of elective courses specified in the course registration module.

c. Returning student

Firstly, the system should automatically select all the REPEAT courses of the past session (if any) for a student.
Secondly, the system should guide the student to select all the TAKE courses (carry-over courses) of the previous session for registration before any new course.
Thirdly, a student should be allowed to select elective courses specified in the course registration module.
Fourthly, a successful registration of courses per semester will be done when the total credit hours for that semester does not exceed the maximum credit hours stated in the database (usually 24 credits per semester).

d. Extra-year student

Firstly, a student is permitted to register all his/her failed courses of the previous session.
Secondly, the system should automatically select all the GSS courses failed in past session if any.
Thirdly, the system should deny any course registration for a student who has spent the maximum number of years in the university (a maximum of 6 years for a UME student in a 4-year program).

e. Probation student

Firstly, a student should not be allowed to select any of the TAKE courses of previous session.
Secondly, the system should auto-select all the REPEAT courses from the past session and not allow registration of new courses from the current semester/session.

f. Problematic student

i. Exam Malpractice and Related Offenses

Firstly, a student who is currently facing the penalty of suspension should not be allowed to do course registration or access any feature of the system until he/she is restored.
Secondly, when a student is under suspension, the results should not be truncated, rather and he/she is allowed to stay the maximum six (6) years in the system, excluding the years on suspension.
Thirdly, a student whose previous session registration has been nullified should be allowed to register courses of his/her previous level not the next level.

ii. Health Challenges

Supplementary examination can be granted for the lost semester(s); which could be registered separately and uploaded.

3.1.2. Evaluation and Inspection of Documents

The following documents were used for evaluation, analysis and data entry: Student Personal Data Form; Result Sheet Format; Examination Reporting Sheet and Transcript.

3.1.3. System Requirements Specification

The system is expected to accept raw scores uploaded by the lecturers during the input process; map the results to individual records in the database; Compute GPA/CGPA of each student; and generate Students Result Reporting Sheet/transcripts for faculty use.
The functional requirements according to user types include: Students, the administrator, Head

of Department, Examination Officer, Academic advisers and Other Staff members.

3.1.4. Functional Requirements of Registration

Admission List Processing: At the beginning of a new session, prior to the addition of new students to the database, a new admission list with JAMB registration numbers is uploaded as a file. In most cases, not all admitted students are cleared for studies; hence, a list of cleared students with matriculation numbers is uploaded and matched by the program to ensure that all the names are in the previous list. All such students whose names appear in both lists are thus assigned user-name and password, and an account is created for each of them in the database. It is at this point that they can make due payments and start course registrations before result uploads could be matched to their names in the system.

The system is able to do the following:

a. Keep track of the numbers of years counted for each student.
b. Allow probation students to register only their repeated courses.
c. In registering courses, the system considers special cases like suspension, deferment of admission, expulsion, supplementary exams and leave on health grounds.
d. In every course registration, the system is able to automatically select all the REPEAT courses of the previous session, for the present course registration process of each student.
e. It enforces the student to select all his/her TAKE courses prior to selecting any of the current semester courses.
f. The system successfully submits the registered courses to the system if and only if the total credit unit does not exceed the maximum credit hours allotted to that student or his/her class.
g. Raises an alarm to prevent the student from registering more than required elective courses, if any.
h. Restrict any change or modification of some sensitive information such as student registration/matriculation number, name, level, year of admission and mode of entry.
i. Authenticate all documents uploaded for the student using document clearance number system before submitting to the database.
j. The system is able to upload course allocation, results for registered courses, staff list and course list from admin or specific staff accounts.
k. The system allows the admin to manage the list of staff and keep the course list up to date.
l. The Admin should be able to grant access to any particular record such as, student record, staff record and courses.

3.1.5. Treatment of Special Students' Cases

A. Offenses/Malpractice: The system takes into account some offenses which may result in rustication, expulsion, suspension or nullification of some academic work.
B. Deferment of Studies: The system provides facilities for a student who wishes to defer his studies.
C. Change of Programme: The system also provides facility for modification of the student's course. If s/he is to lose some years, the last valid session is also specified from where all academic work commences.
D. Health Cases: A student with approval of a medical report has such a case created on his account indicating suspension of studies until further notice.
E. Supplementary Examination: Approval of supplementary examination is done by the HOD into the system after the Registrar must have granted it.
F. Reinstatement of Students: On all these cases, the student is expected to report back to the department after which he is reactivated back into the system.

3.2. Software Process Model

A software development methodology is a framework that is used to structure, plan, and control the process of developing an information system/software. The incremental model hereby adopted, is a popular version of the systems development life cycle for software engineering

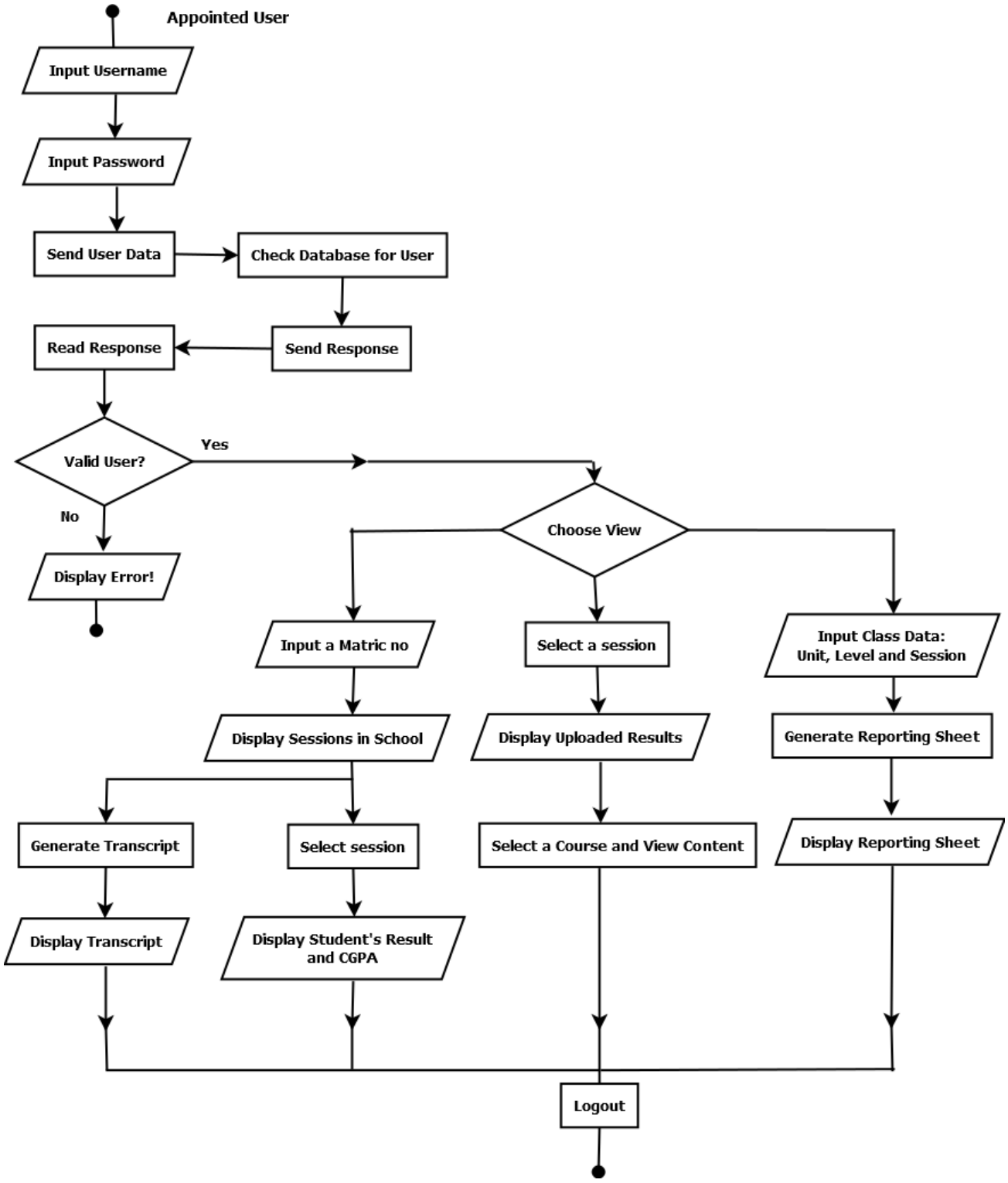
used to build several versions of the system as separate increments. In the name, the incremental model adopts an evolutionary approach using iterative development cycles to meet requirements at specific deadlines. Each step of iteration on the increment goes through the analysis, design, coding and testing phases before delivery.

4. SYSTEM DESIGN

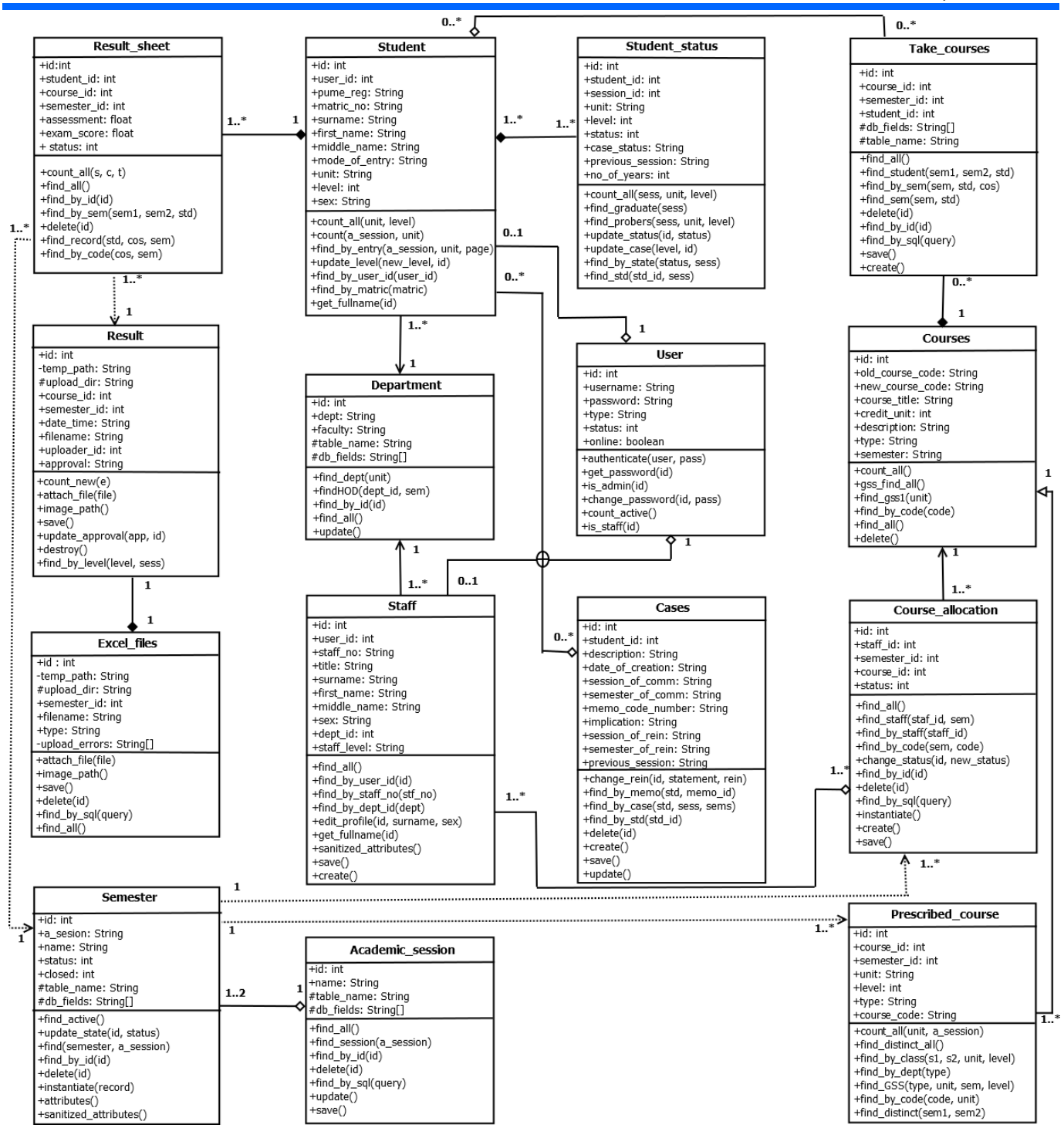
Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements of the user. During this phase, the team focused on planning efforts for both the testing and data entry activities of the computerized SRMIS. This is often conducted via modeling such as UML, ER-Diagram and Use Case Diagram (Pressman, 2001).

4.1. Physical Design

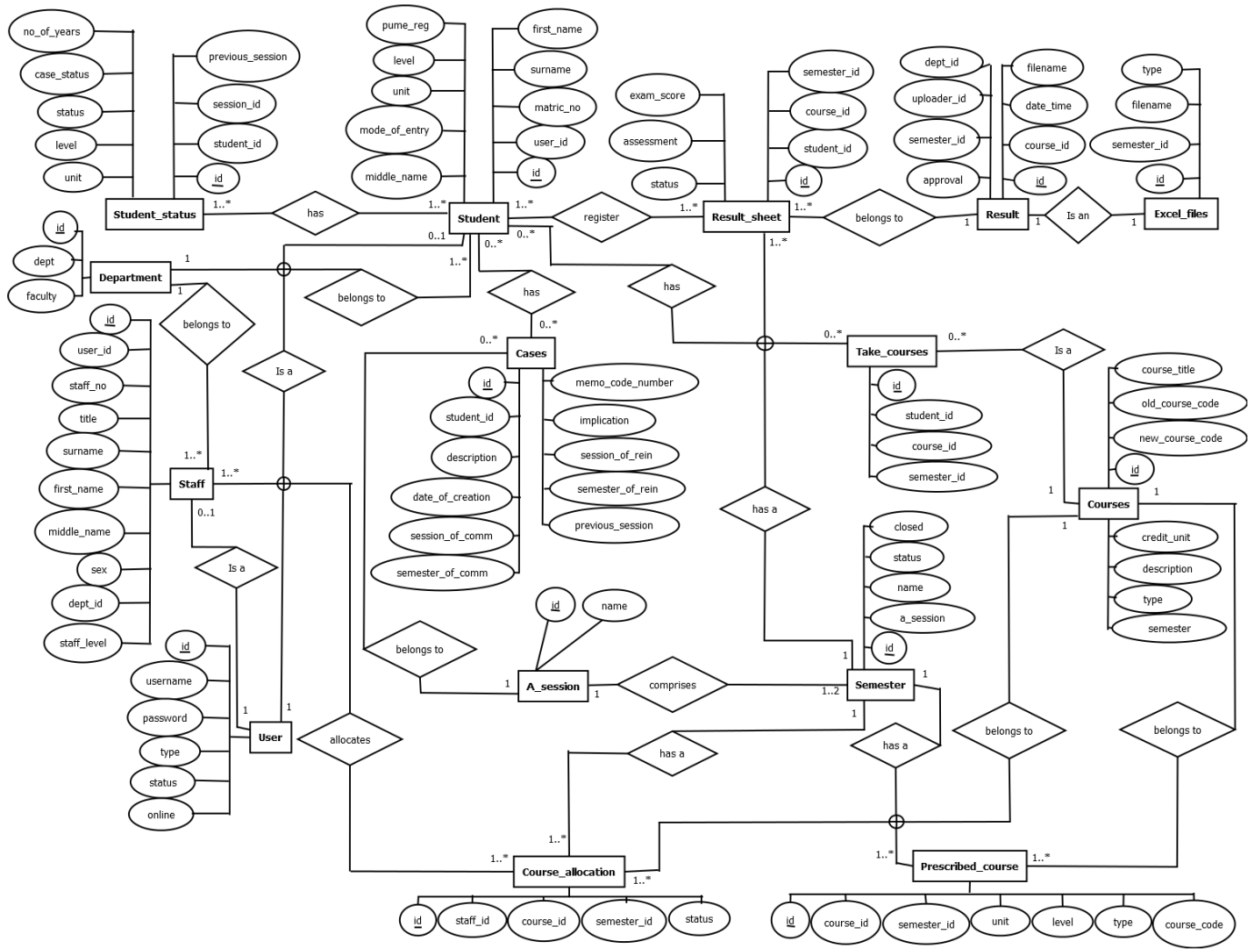
Physical design is like a set of blueprint for the actual construction of a building. It describes the actual processes of entering, verifying and sorting procedures, the format of report, etc.
Before the physical database design can be performed, the following were considered: the maximum size of the database, which was determined by how many entities are in the database and how much data is held in each entity. The following tables were used because their records increase consistently per session: Student, Prescribed_courses, Course_allocation, Result_sheet, Results, Excel_files and Student_status entities.



Activity Diagram.



Class Diagram.



Entity-Relationship Diagram (E.R.D.)

4.2. Database Design

Database design is an organized collection of records stored for easy updating and retrieval. They layout/design of the database used for this system involved the following tables: Take_courses, Student, Course_allocation, A_session, Course, Excel_files, Student_classes, Officers, Cases, Result_sheet, Result, Prescribed_course, Semester, Student_status, User, Staff etc. Each of them has a primary key, a number of foreign keys and several other fields normalized to minimize redundancy. Some of the tables are displayed below.

5. SYSTEM IMPLEMENTATION

5.1. Features and Choice of Implementation Language

The programming technologies/languages used in the implementation of this project are as follows: Cascading Style Sheet (CSS), Javascript, Hypertext Markup Language (HTML), Hypertext Pre-processor (PHP), MySQL (Structured Query Language) Database Management Systems and Windows or Linux Apache-MySQL-PHP (WAMP/LAMP) server. The system has a three-tier architecture which consists of the front-end, the middle-end and the back-end which are discussed below:

Front-end: In web based application, Front End is the content rendered by the browser. The content may be static or generated dynamically with a scripting language. In this case, the content of the developed system is generated

dynamically. The technologies used for the front-end are HTML, JavaScript and CSS.

Middle-end: A middle-end is a dynamic content processing and generation level application server. This layer creates easy accessibility between the client and the server. It is the logical layer of the system. The scripting language used to develop the middle-end of this system is PHP.

Back-end: This tier is also known as database or data store, comprising of data sets and the database management system (DBMS) software that manages and provides access to the data. Here information is stored and retrieved. This tier keeps data neutral and independent from application servers or business logic. MySQl is used in this system to implement the back-end layer.

5.2. Software Testing

During the development of this system, unit testing was carried out by the developers on each of the result processing modules.

Results from the integration test of this project were satisfactory; although, exhaustive debugging may not be realistic in a very large and complex program. The system was tested on different browsers and operating systems such as Opera, Google Chrome, Safari, Mozilla Firefox, Microsoft Internet explorer, Windows and Linux. The result of the test was satisfactory.

5.3. Software Maintenance Issues

Preventive, adaptive and corrective maintenance were carried out on this system for future transfer-ability of administration.

6. CONCLUSION

This team finally presented a software application meant to automate the processing of students results following standard statutes and guidelines. The application was successfully developed, tested, and found to be working as expected. The Result Processing System was developed using PHP and MySQL. It has some qualities such as reduction in the cost of processing, reduction in time spent in computing GPAs/CGPAs, generation of transcripts, and elimination of duplicate records, which makes it superior to the manual system of students' results processing. Based on the findings of this research, the following conclusions have been made: Input design for Result Processing - a subsystem of the SRMIS - is of utmost importance in the area of pipe-lining, as it allows files containing already prepared data to be uploaded instead of entering pieces of data one after the other into the system via keyboard and forms. This project is essential to any academic organization that works with large amount of student data; as it allows file uploads as input into the system and addresses cases affecting student status while abiding by due registration processes.

6.1. Recommendations

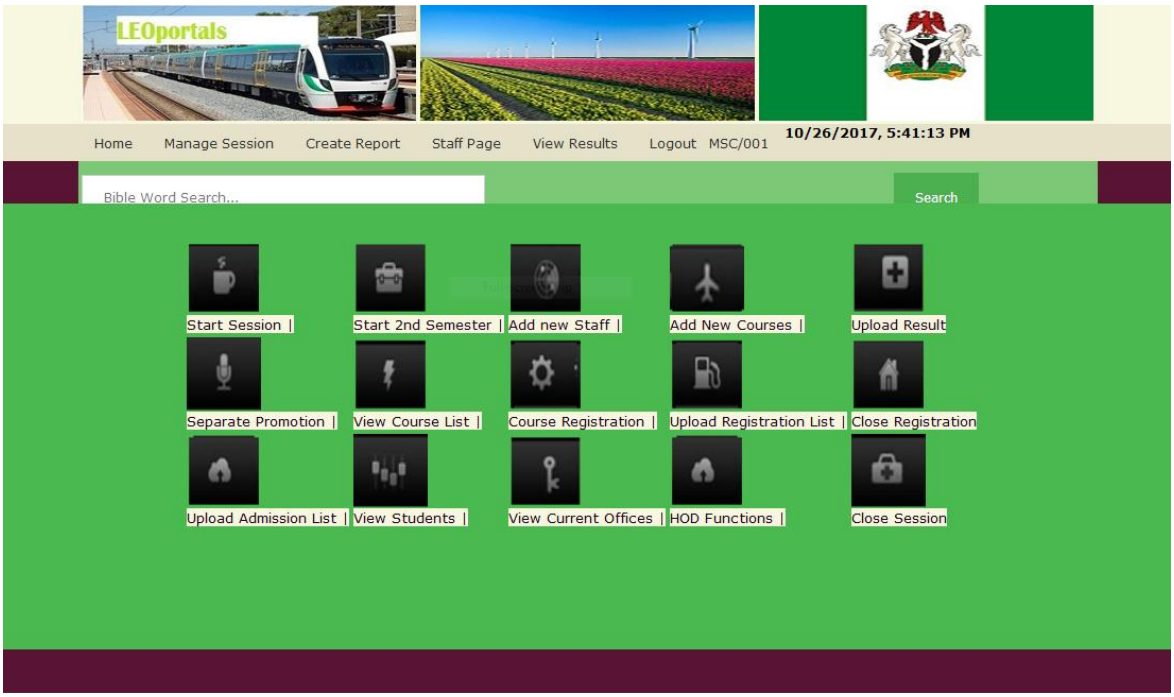
The mobile result processing system should be developed to enhance flexibility

in operation, although this solution can be viewed on most mobile browsers via a network. The system should be recommended for other departments in the faculty and the entire school which will hasten the compilation of results for graduation and convocation ceremonies. There should be regular orientation of new staff and students on the statutes implemented in the system following a user manual. Users should ensure at all times that the SRMIS is not left unlocked on their computer. There should be provision of network connection ports on sitting desks, if possible wireless connection, to prevent delay in registration and result compilation. There should be a vetting team to ensure that the results uploaded were in the recommended format with accurate values.

ACKNOWLEDGEMENTS

We thank the Supervisor, Prof. Zsolt Lipcsey and the former Head of Department Dr. I. O. Isaac; Dr. O. A. Ofem and the entire members of the Department of Computer Science, University of Calabar, Calabar, Nigeria. We also appreciate other contributors like Unah Henry, Ofor Williams, Okoligbe O. Trust, Furo I. Dawari, Nzeh Margaret-Mary, Enebeli O. David, Udeze O. Daniel and Takon Takon Atu. We also acknowledge Channel of Blessings Ministry, Calabar, Cross River State.

SCREEN SHOTS FROM THE ONLINE SYSTEM (Taken from www.leoportals.com using an administrative account)



Main Admin Menu for Results Processing.

MATRIC NO 08/50089
NAME [redacted]
LEVEL 300
YEAR OF STUDY 3/4
PROGRAMME BSc. COMPUTER SCIENCE
SESSION 2010/2011

FIRST SEMESTER RESULT

S/NO	COURSE CODE	COURSE TITLE	CREDIT UNIT	CONTINUOUS ASSESSMENT	EXAM SCORE	TOTAL SCORE	GRADE
1	ECO111	PRINCIPLES OF ECONOMICS I	3	18	37	55	C
2	GSS3101	ENTERPRENEURSHIP DEVELOPMENT I	2	11	44	55	C
3	MTH313	DISCRETE MATHEMATICS	3	24	60	84	A
4	CSC301	COMPUTER LOGIC DESIGN	3	15	56	71	A
5	CSC311	OPERATING SYSTEMS I	3	26	58	84	A
6	CSC321	COMPUTER ARCHITECTURE I	3	10	66	76	A
7	CSC331	COMPILER CONSTRUCTION	3	27	62	89	A
8	CSC381	COMPUTER LABORATORY IIIA	1	30	36	66	B
9	CSC390	INDUSTRIAL ATTACHEMENT II	6	29	43	72	A

SECOND SEMESTER RESULT

S/NO	COURSE CODE	COURSE TITLE	CREDIT UNIT	CONTINUOUS ASSESSMENT	EXAM SCORE	TOTAL SCORE	GRADE
1	ECO122	PRINCIPLES OF ECONOMICS II	3	13	56	69	B
2	GST302	ENTERPRENEURSHIP DEVELOPMENT II	2	22	54	76	A
3	MTH352	ORDINARY DIFFERENTAIL EQUATIONS I	3	21	64	85	A
4	CSC312	OPERATING SYSTEMS II	3	17	53	70	A
5	CSC322	COMPUTER ARCHITECTURE II	3	13	50	63	B
6	CSC332	DATA-BASE DESIGN AND MANAGEMENT	3	26	51	77	A
7	CSC342	SYSTEMS ANALYSIS AND DESIGN	3	20	47	67	B
8	CSC382	COMPUTER LABORATORY IIIB	1	26	36	62	B

Sessional GPA: 4.56 Cumulative GPA: 4.53 FIRST CLASS

Students' Results verification page.

NATIONALITY: NIGERIAN **STATE:**
FACULTY: SCIENCES **DEPARTMENT:** COMPUTER SCIENCE

Course	No.	Title	Credit Hours	Grade	Grade Points
FIRST YEAR					
2008/2009					
First Semester					
CHM	101	INTRODUCTION CHEMISTRY I	3	C	9
CSC	110	INTRODUCTION TO COMPUTER SCIENCE I	2	E	2
CSC	118	COMPUTER LABORATORY IA	1	E	1
GSS	110	USE OF ENGLISH I	2	D	4
GSS	113	HISTORY AND PHILOSOPHY OF SCIENCE	2	C	6
PHY	111	ELEMENTARY MECHANICS	3	C	9
MTH	111	ELEMENTARY MATHEMATICS I	3	A	15
MTH	181	PROBABILITY AND STATISTICS	3	E	3
Second Semester					
CHM	102	INTRODUCTION TO CHEMISTRY II	3	A	15
CSC	110	INTRODUCTION TO COMPUTER SCIENCE II	2	B	8
CSC	118	COMPUTER LABORATORY IB	1	C	3
PHY	118	LABORATORY PHYSICS	1	D	2
GSS	110	USE OF ENGLISH II	2	B	8
GSS	112	PHILOSOPHY AND LOGIC	2	A	10
GSS	111	NIGERIA PEOPLE AND CULTURE	2	B	8
MTH	135	ELEMENTARY MATHEMATICS II	3	A	15
PHY	115	GENERAL PHYSICS II	3	E	3
MTH	182	BASIC SAMPLING THEORY	3	A	15
Total(1)			**41**		**136**
GPA					**3.32**
SECOND YEAR					
2009/2010					
First Semester					
CSC	220	COMPUTER PROGRAMMING I	3	B	12
CSC	222	INTRODUCTION TO COMPUTER SYSTEMS	2	A	10
CSC	228	COMPUTER LABORATORY IIA	1	B	4
MTH	281	ESTIMATION AND TEST OF HYPOTHESIS	3	A	15
MTH	261	VECTORS AND MECHANICS I	3	A	15
MTH	235	CALCULUS	3	A	15
MTH	211	LINEAR ALGEBRA	3	A	15
BIO	101	INTRODUCTURY BIOLOGY I	3	B	12
Second Semester					
CSC	220	COMPUTER PROGRAMMING II	3	C	9

Automatic Transcript Generation

Page 1

UNIVERSITY OF CALABAR
CALABAR

FACULTY: FACULTY OF PHYSICAL SCIENCES
DEPARTMENT: COMPUTER SCIENCE
PROGRAMME: B.Sc. COMPUTER SCIENCE

YEAR OF STUDY: 4/4
SESSION:
SEMESTER: FIRST/SECOND

EXAMINATION REPORTING SHEET
SESSIONAL RESULTS

S/NO	MATRIC NO	REPEAT COURSES	CARRY OVER COURSES	FIRST SEMESTER RESULTS	CSC4000	CSC4151	CSC4211	CSC4311	CSC4401	CSC4411	CSC4441	CSC4451	CSC4481	SECOND SEMESTER RESULTS	CSC4000	CSC4072	CSC4122	CSC4165	CSC4162	CSC4452	CSC4832	TCH	TGP	GPA	CTCH	CTGP	CGPA	CLASS OF DEGREE	REMARKS
				CH	6	2	3	3	3	3	2	2	1	CH	6	3	3	2	2	3	3								
1		CSC2201F/F CSC2222F/F CSC3312F	CSC3311, MTH3121,	3 CSC2201 E 3 CSC3311 E 3 MTH3121 C		A		E	D	D	E		E	2 CSC2222 B 3 CSC3312 B	B	C	C	C	C		E	47	120	2.55	178	431	2.42		TAKE: CSC4211.
2		ECS1301F PHY2262F/F	MTH3121, CSC3301, CSC3311, CSC3331, CSC3381, MTH3512, CSC3312,	3 ECS1301 D 3 MTH3121 B 3 CSC3301 C 3 CSC3311 B 3 CSC3331 C 1 CSC3381 C	A						C			3 PHY2262 D 3 MTH3512 D 3 CSC3312 B		B	C	D	B		C	46	153	3.33	185	395	2.14		TAKE: CSC4211, CSC4401, CSC4311, CSC4411, CSC4481. CARRY F IN PHY1152.
3		CSC2281F/F MTH2351F CSC3582F	MTH3121, CSC3311, CSC3381, CSC3312, CSC3332, ECS1202,	1 CSC2281 F 3 MTH2351 C 3 MTH3121 B 3 CSC3311 F 1 CSC3381 C		B	C		C	D		B		1 CSC3582 C 3 CSC3312 C 3 CSC3332 E 3 ECS1202 D		E	E	C	B	C		47	114	2.43	186	383	2.06		RPT: CSC3311. TAKE: CSC4000, CSC4311, CSC4481. CARRY F IN CSC2281, PHY1152.
4		CSC2282F/F ECS1202F CSC3312F CSC3382F	CSC3381, MTH3512,	1 CSC3381		E	B	C	B	F	D		F	1 CSC2282 F 3 ECS1202 F 3 CSC3312 C 1 CSC3382 E 3 MTH3512 C	B	C					E	40	94	2.35	177	420	2.37		RPT: CSC4411, ECS1202, CSC4481. TAKE: CSC4122, CSC4145, CSC4162,

Generation of Reporting Sheet (Major Output)

REFERENCES

[1] Amar, E. and Mohini (2009). Single portal for integrated examination system, emerging technologies in e-governance, *E-Governance*, pages 287-293.

[2] Beizer, B.S. (1990). *Software Testing Techniques*, 2nd ed. Van Nostrand-Reinhold, USA.
[3] Bel, D. (2005). *Software Engineering for Students: A Programming Approach*. Pearson Education Limited, UK, USA.
[4] Converse, T., Park, J. and Morgan, C. (1990). *PHP5 and MySQL Bible*. Wiley Publishing, Inc., 10475 Crosspoint Boulevard Indianapolis, IN 46256 : http://www.wiley.com.
[5] Juan. (2004). *Information Technology*. Pearson Education Limited, UK, USA, 2004.
[6] Kruchten, P.(1999). *From Waterfall to Iterative Development - A Challenging Transition for Project Managers*. Addison Wesley Longman.
[7] McConnell, S. (2004). *Code Complete, 2nd edition*. Microsoft Press, 2004. ISBN1-55615-484-4.
[8] McConnell, S. (2006). *Software Estimation: Demystifying the Black Art*. Microsoft Press. ISBN0-7356-0535-1.
[9] Mitra, A. (2011). *Classifying data for successful modeling*. Information and Science Technology, vol 16,4:234 346.
[10] Okonigene, R.E., Ighalo, G.L and Ogbeifun, E. (2009). Developed personal record software . *The Pacific Journal of Science and Technology*, 9(2):407-412.
[11] Pratap, K.J. M. (2010). *Software Engineering (A Lifecycle Approach)*. NEW AGE INTERNATIONAL (P) LIMITED, PUBLISHERS, 4835/24, Ansari Road, Darya-ganj, New Delhi - 110002 Visit us at www.newagepublishers.com. ISBN (13) ; 978-81-224-2846-9.
[12] Pressman, R. S. (2001). *Software Engineering Approach: A Practitioner's Approach, fifth edition*. McGraw Hill Higher Education, New York, USA.
[13] Quinn, M. J. (2004). *Parallel Programming in C with MPI and openMP*. Dubuque, Iowa: McGraw-Hill Professional.
[14] Royce, W. (1970). *Managing the Development of Large Software Systems*. IEEE WESCON.
[15] Sidney, L. (2008). *Computer Programming.* Addison-Wesley,U.K.
[16] Ukem, E. et al. (2012). A software application for university students results processing. *Journal of Theoretical and Applied Information Technology*, Vol. 35 No.l:10.
[17] Vossen, G. F. (1991) . *Data Models, Database Languages and Database Management Systems*. U.S.A: Addison Wesley Publishing Company.
[18] Welling, L. (2007). *PHP and mySQL Bible*. Wiley publishing inc. Indiana, Indianapolis.
[19] Williams, L. (2008). *A (Partial) Introduction to Software Engineering Practices and Methods, fifth edition*. Addison-Wesley.

[20] Akinmosin James (2014). Automated Students Result Management System using Oracle's Database, Forms and Reports. *Journal of Information Engineering and Applications*, (Vol 4, No. 11). Retrieved from www.iiste.org .
[21] Idogho O. P., Akpado K., and Agajo J. (2011). Interactive Intranet Portal for Effective Management in Tertiary Institution. *International Journal of Advanced Computer Science and Applications*, (Vol. 2, No. 6). Retrieved from www.ijacsa.thesai.org .
[22] Bijoy C., Sanjay K. P., Bhibak S., Nishal M. and Zarmit L. (2016). Accessing a portion of MIS: Result Management System. *International Journal of Engineering Trends and Technology*, (Vol. 34, No. 5). Retrieved from www.ijettjournal.org .
[23] Duan R. and Zhang M. (2010). Design of Web-based Management Information System for Academic Degree and Graduate Education.
[24] Bharaagoudar S. R., Geeta R. B. and Totad S. G. (2013). Web-based Student Information Management System. *International Journal of Advanced Research in Computer and Communication Engineering*, (Vol.2, Issue 6). Retrieved from www.ijarcce.com .

[25] Hemn B.B. and Wu Fei (2014). Student Management Information System. *International journal of ComputerEngineering and Technology(IJCET)*, Vol. 5, Issue 2.
[26] Mariusz Czerniak (2010). Student Management Information System for Polish Universities at its tenth anniversary. Warsaw: MUCI.
[27] Ajay S. B. and Abhishek K. (2012). Student Database Management System.
[28] https://www.wikipedia.com/ Student Information System
[29] Bhatt J., Jain R., Kadge S. and Parate P. (2016). Result Generation System for CBGS Scheme in Educational Organization. *International Journal of Research in Engineering and Technology (IJRET)*, Vol. 5, Issue 2 from http://www.ijret.org
[30] Nmaju O., Nwachukwu E. O., and Ugwu C. (2013). A Novel Web-based Student Academic Records Information System. *West African Journal of Industrial and Academic Research*, Vol. 7, No. 1.
[31] Charletta F. G. (2004). Ethics in Information System: Student Performance in Evaluating Ethical Dilemmas. *Communications of the International Information Management Association*, Vol. 4, Issue 1.
[32] JISC InfoNet (2007). HEI Records Management, Guidance on Managing Students Records.
[33] Gunathilake R. M., Indrathilake G. R. and Wedagedera J. R. (2009). Open Source, Web-based Management Information System for Faculty Administration. MIS Faculty of Science. 6th Ac. Session.

[34] Suvoica (2013). Guidelines on Appraisal and Disposition of Student Records. *International Council on Archives, Section on University and Research Institutions Archives*.
[35] O'Brien J. A. and Marakas G. M. (2008). *Management Information Systems, Eight Edition*. McGraw-Hill/Irwin.

[36] Walia S. and Gill K. S. (2014). A Framework for Web-based Student Record Management System using PHP. *International Journal of Computer Science and Mobile Computing (IJCSMC)*, Vol. 3, Issue 8. Retrieved from www.ijcsmc.com