

Improving Students Academic Transcript Generation in Nigerian Universities

Victoria Oluwatoyin Oyekunle,
Department of Computer Science,
Federal Polytechnic of Oil and Gas, Bonny, Rivers State, Nigeria

Abstract. At the end of each semester in schools of higher learning, examinations are written by students on the courses registered for and the grades are computed. In some universities, the scores and grades are entered into results sheet, the cumulative grade point average [CGPA] is calculated and recorded manually and kept in paper files that are carefully arranged into folders and kept in shelves. The aim of implementing this project is to create a software that can be used for Course registration, result computations and Student Transcript generation. With this system, we will be solving the problems encountered in the process of computing student's examination results, and retrieving the results when they are needed. From the results, the computation of students' academic results was made simpler, less tedious and easily accessible with a robust database system.

Keywords: Courses, Course unit, Grade point, Total load unit, Total unit passed, Total credit points, Grade point average, Cumulative grade point average, Students, Academic transcript

I. Introduction

The advent of information technology has brought about many significant changes in major aspects of our economy. Tedious jobs that could have taking hours and sometimes days to complete now take minutes and at times seconds to complete with the introduction of a computerized system. Most modern businesses and government-owned institutions and establishments depend on computers to help in operations management. They have been used to design payrolls, manage customer records and transactions, keep track of stock etc. On the global scale, the impact of computers have been felt through its use in reporting current events, world wide access to data and the use of robots in industry. The student transcript generating system involves the computation and storage of students' results. At the end of each semester in schools of higher learning, examinations are written by students on the courses registered for and the grades are computed. The scores and grades are entered into results sheet, the cumulative grade point average [CGPA] is calculated and recorded manually and kept in paper files that are carefully arranged into folders and kept in shelves. A transcript is given after graduation. It is a document that provides information about student's academic records such as courses taken by the student, grades received in different courses by the student, and degree(s) conferred to the student. A result administrator from the academic institution provides a transcript to the student. In order to generate, verify, and distribute the transcript, the institute requires sufficient manpower and time (Swarmy et al., 2020). Also, whenever students move from one higher institution to another, they are expected to submit an official transcript from the previous institution (Patel et al., 2020). The main disadvantage of the manual approach is that it is relatively easy for an enemy to tamper with the transcripts, and detecting such unauthorized changes is tough. The computation of students' academic results can be made simpler, straightforward, and less time consuming with the use of modern technology. When the computation of the results is computerized, it allows for faster access to them when needed and makes the computation easier. There is a need for a dependable system with an effective and simple application and procurement process for academic transcripts, as well as adequate methods for authentication of given transcripts and verification of transcript integrity (Khedkar et al., 2019). When using the student transcript generating software system

in computing results, it limits the time wasted in using manual calculations to compute the student results and it also limits the errors in calculation.

The rest of the paper is organized as follows. Section II covers the concept of examinations and grading. Section III explains the system design, the input and the design of the output. Section IV presents the implementation of the system and the program testing stage. Section V concludes the research work and contains recommendations of the study.

II. Related Work

Most Nigerian universities' present transcript generation and validation infrastructure is mostly focused on physical operations and paperwork. This frequently necessitates the presence of the student or a representative to deliver relevant documents and/or collect issued transcript copies. There are a few online options for transcript application and verification. Many of these, however, are only partially online, and the transcripts retrieved are physical records. Hence in this section, we will examine the current physical processes involved in the examination results recording and grading, course registrations and result processing in Nigerian Universities.

A. Examinations

The traditional approach to measuring a person's level of knowledge in a course or on a topic has been the "examination". These days, there are often more emphasis on "internal" assessments, which may consist of assignment and projects given out by the teacher and then marked or assessed by the same teacher. However, examinations remain a popular component of many assessments.

The main activities involved in preparing examination results can be described under two sub-heading which are recording, and updating.

1. RECORDING: This is done immediately after the performance of a student is measured. It entails putting the results of the performance measurements into record files. The results include results generated from assignments, tests, attendance and examination marks.

2. UPDATING: this involves the modification of the previous result to get the current one. These modifications are carried out by adding previous results to the present one and finding the average.

Hence, it will replace the manual method of recording results in paper files with recording results in database files and designing software capable of computing the grade points and grade point average.

B. The University Grading System

The University grading system provides a means of finding the average performance of a particular student in several courses in a given semester. The university does this by conducting a written examination of 2-3 hours duration in each course registered by students at the end of the semester in which the course is concluded. registration of courses take place at a specified period at the beginning of each semester during which a student must register for all the compulsory and required courses prescribed by the department. In addition, each student must register for the general studies (GST) courses which he/she is required to pass in order to qualify for the award of a degree. The examinations carry a minimum mark of seventy percent (70%) of the total marks allocated to the course. Continuous assessments carried out during the course of the semester carry the remaining thirty percent (30%).

C. Gradings

For a pass in a course a student must have a score of 40% and above. In case of a failure, the student loses all units assigned to the course. In each semester, the minimum workload for

a full-time student is 12 units while the maximum is 24 units. The marks obtainable by a student in examinations are graded using the alphabet A-F. Also, each of the grades is associated with a rating called the grade point and it is usually within the interval 0-5. The grading system used for examination classification is given in Table 1.

Table 1: Table showing scores, grades and assigned grade points

| Marks obtainable | Grade | Grade points |
|------------------|-------|--------------|
| 70% and above | A | 5 |
| 60% - 69% | B | 4 |
| 50% - 59% | C | 3 |
| 45% - 49% | D | 2 |
| 40% - 44% | E | 1 |
| Below 40% | F | 0 |

From the above grading system, it makes it easy to evaluate the overall performance of a student over successive semesters and cumulative grade point average (CGPA) can be easily calculated for each semester. The class of the degree a student can obtain is awarded on the basis of the overall cumulative grade point averages (GPA) as indicated in Table 2.

Table 2: Table showing grade classes assigned to each CGPA range

| CGPA | Class of degree |
|-------------|-------------------------------|
| 4.50 - 5.0 | First class |
| 3.50 - 4.49 | Second class (upper division) |
| 2.40 - 3.49 | Second class (lower division) |
| 1.50 - 2.39 | Third class |
| 1.00 - 1.49 | Pass |

III. Materials and Methods

This section gives a complete functional, logical and physical design of the system. The logical design is presented through a context diagram, and a dataflow diagram. The physical design is presented through the description of the database designed to keep the results using screen shots that explain the concepts better, the interfaces designed with Microsoft Csharp to accept input data into the database, and the output design which is in form of reports.

A. Context Diagram of the System

This type of diagram according to Kossiakoff (2003) usually “pictures the system at the centre, with no details of its interior structure, surrounded by all its interacting systems, environment and activities.” The objective of a system context diagram is to focus attention on external factors and events that should be considered in developing a complete set of system requirements and constraints.

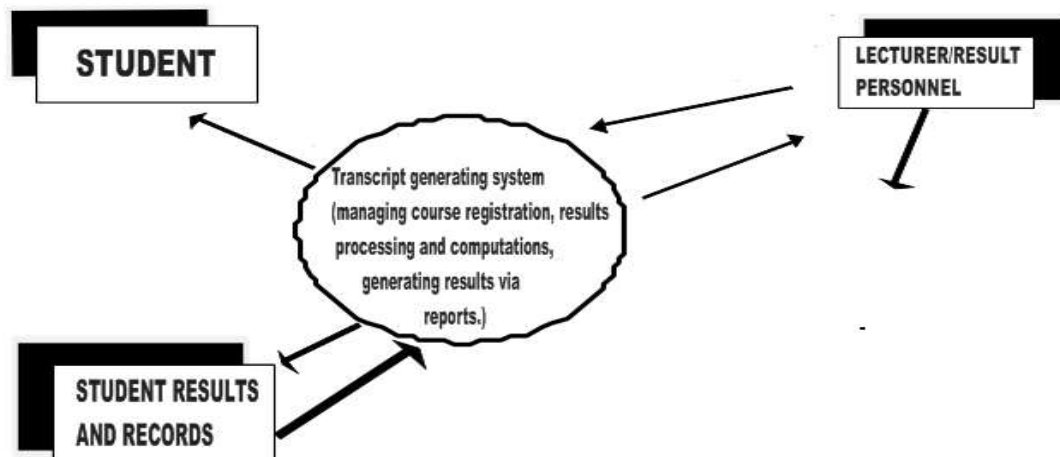


Figure 1: Context diagram for the transcript generating system

B. Data Flow Diagram of the System

A Data flow diagram (DFD) is a graphical representation of the “flow” of data through an information system, modeling its process aspects. They are used to illustrate the manual and computer processes of a computer system.

A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel. Detailed design information such as database, screen accesses, error messages, and end-of-file conditions are not specified in the data flow diagram.

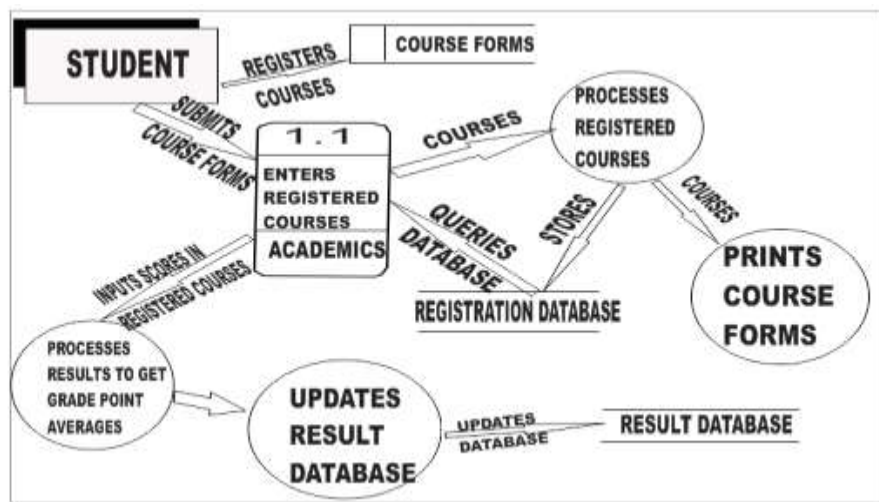


Figure 2: Data flow diagram for the system

From Figure 2, details relating to students like their Course details (record of all registered students in the department, the courses registered for, the semester and session, and the current level of the student), Course registration details (record of courses registered by students in a given department and their corresponding units), Department details (record of all codes and names of departments in each faculty in the university), Department courses details (record of all courses offered in each department per semester per level), Faculty details (records of all codes and names of faculties in the university), Result details (records of all students per session and semester) all serve as inputs to the transcript generating system. These inputs are processed and output given in form of reports.

IV. Implementation and Results

The system was implemented with Microsoft's Visual Csharp.Net (C#) on the Dot Net framework 4.0 using Visual Studio Integrated Development Environment 2017 (VS-IDE) for Interface Designs of Active server Pages (ASP) and Code Editing, Microsoft SQL Server 2008 for Database Management of the records. Internet Information Service (IIS) for the local web hosting and Crystal Reports 2017 for designing the feedback of Course Registrations and Results.

After a successful deployment of the computerized system, the administrator can launch the program using any browser (internet explorer, flock, Mozilla Firefox, Google chrome, opera). After the application has been launched, the application home page or main window comes up immediately as seen in Figure 3.

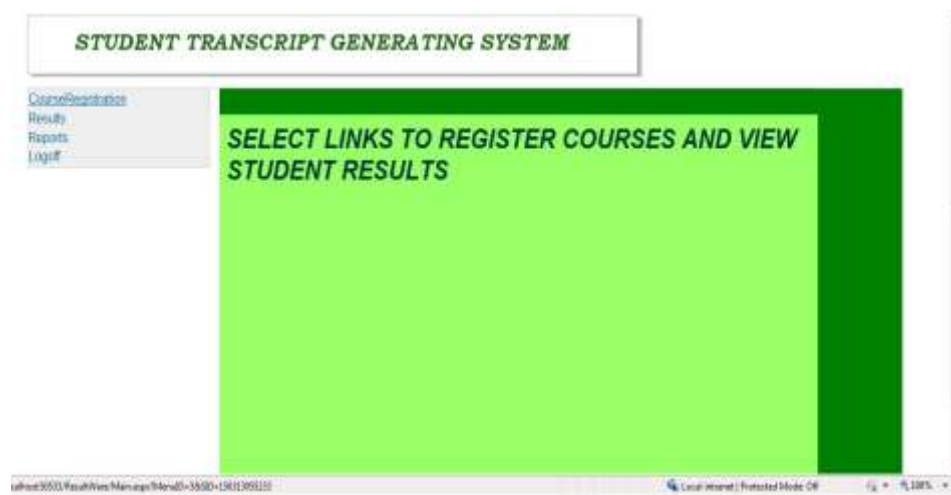


Figure 3: The Application's main menu window

A. Student Course Registration

In order to supply the Course Registration details for a semester, click on 'Course Registration' on the home page. Figure 4 below is displayed. The fields are then populated with the relevant details.



Figure 4: The Course Registration module window

To enter new details, click the 'NEW' button, enter details into the fields and click 'OK' button to commit record. When the student's faculty field is filled, the department field is automatically populated with the names of all departments under the faculty name. Figure 5 illustrates that.

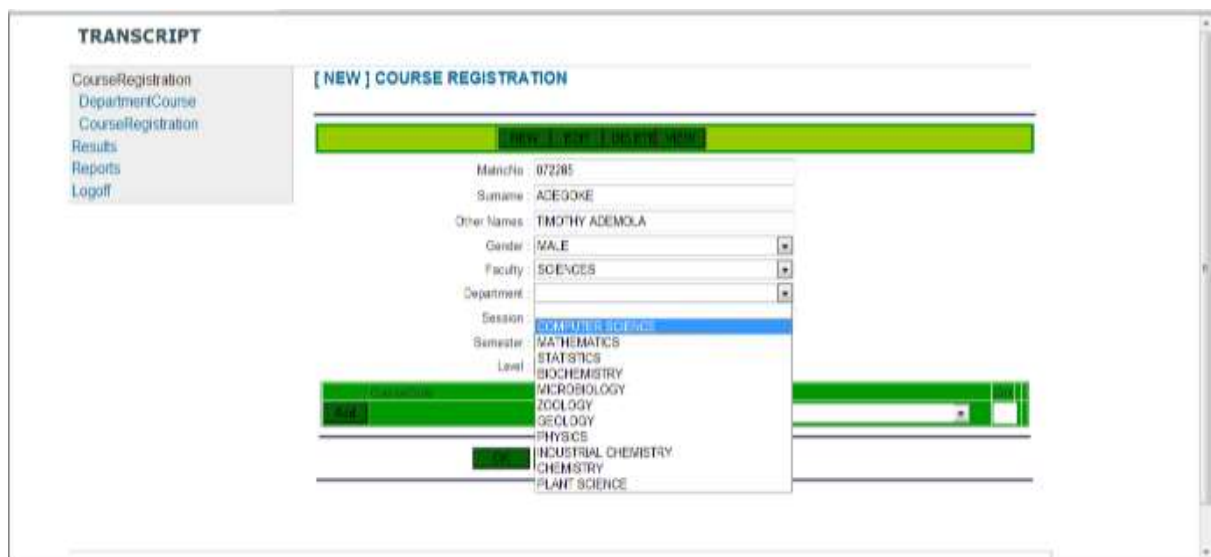


Figure 5: When the ‘Faculty’ field is filled, the ‘Department’ field is automatically populated

When the ‘Course’ drop down list is clicked, all courses offered by the department are automatically generated from the database. When the course is selected and the unit specified and the ‘ADD’ button is clicked, the ‘Course code’ is generated from the database as seen in Figure 6.

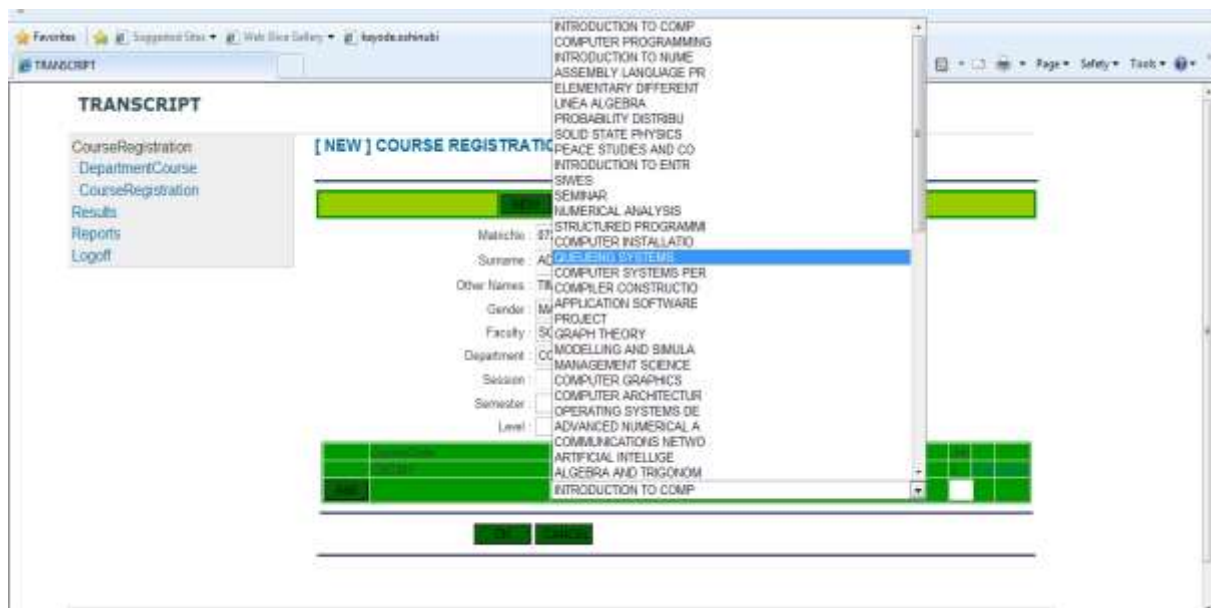


Figure 6: The ‘Course’ drop down list showing all courses offered in the department

Figure 7 shows an already populated course registration module. To edit a submitted record, click ‘EDIT’, edit the required fields then click ‘OK’. To delete a record already submitted, click the ‘DELETE’ button and then ‘OK’.



Figure 7: An already populated Course Registration module

B. Result Entry

This module allows the scores of a student in the courses registered for in a semester to be input. A screenshot is shown in Figure 8.

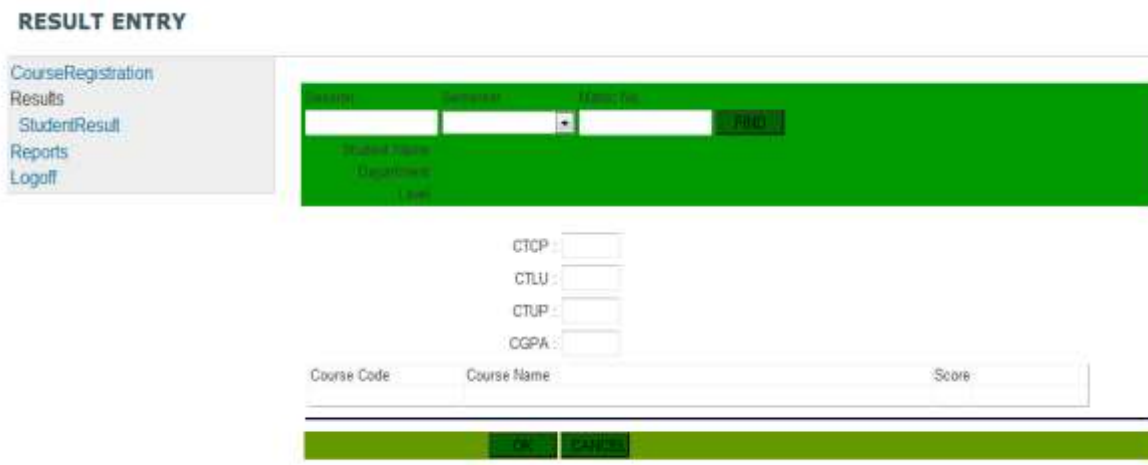


Figure 8: Result entry module

When the ‘Session’, ‘Semester’, and ‘Matric No’ fields are filled, and the ‘FIND’ button is clicked, the application generates the student’s name, department, level, and the courses registered for the semester. The student’s scores in those courses can then be keyed in. The student’s previous Cumulative total credit points (CTCP), Cumulative total load unit (CTLU), Cumulative total unit passed (CTUP), and Cumulative grade point average (CGPA) for previous semesters can be input too. The ‘OK’ button is then clicked to commit the records. Figure 9 displays this.



Figure 9: Result entry window showing populated fields. To edit a score, click 'Edit'

C. Output of the System

There are various outputs that can be generated from this application. They can be printed out by clicking the 'Print' icon on the report. They are;

- a) Student course form.
- b) Student per-session transcript.

Some sample reports for the output of the application are shown in Figure 10 and Figure 11.



Figure 10: Student Course form

| Session | Semester | GPA |
|-------------------|----------------------|--------------------------|
| 2008/2009 | First | 4.81 |
| <u>CourseCode</u> | <u>CourseName</u> | <u>Unit</u> <u>Grade</u> |
| PHY203 | NETWORK ANALYSIS | 3 A |
| MAT201 | MATHEMATICAL METHODS | 3 A |
| CSC215 | COMPUTER LOGIC | 3 A |
| CSC213 | ASSEMBLY LANGUAGE | 3 A |
| MAT203 | LINEAR ALGEBRA | 2 B |
| MAT207 | REAL ANALYSIS | 3 A |
| GST211 | USE OF ENGLISH2 | 2 B |
| GST213 | CONFLICT STUDIES | 2 A |
| Session | Semester | GPA |
| 2008/2009 | Second | 4.60 |
| <u>CourseCode</u> | <u>CourseName</u> | <u>Unit</u> <u>Grade</u> |
| CSC212 | C++ | 4 B |

Figure 11: Student transcript per-session

V. Conclusions

In this paper, the objective was to enhance the generation of Student Transcript in Nigerian Universities using a secured database. This application supports result administrators in managing students’ registration and their results in an efficient manner. It provides an easier approach to results computation and transcript generation which can be produced by the application after necessary details are input by the administrator. The system can function both as a web application and a windows desktop-based application and it has the capability of supporting examination management in any university if properly implemented. The modular nature of the design makes it easy for further integration and upgrade whenever it becomes necessary. This application will perform effectively for the intended purpose of reducing the time and stress encountered during manual computations. It is therefore recommended for use in all Nigerian Universities and Affiliate institutions.

References

- Churcher, C. (2012). *Beginning Database Design*. Apress Publishing Inc, United States of America.
- Date, C. J. (2012). *An Introduction to Database Systems* (4th ed.). Addison- Wesley Publishing Company, Canada.
- Dewson, R. (2008). *Beginning SQL Server 2008 for Developers: From Novice to Professional* (1st ed.). Apress; edition.
- Ekiti State University Student Handbook (2014). Ekiti State University press. Ado-Ekiti, Ekiti State, Nigeria.
- Khedkar, S., Powar, A., Powar, N., Kille, C., & Kansara, H. (2020). *Transcripts DApp—A Blockchain-Based Solution for Transcript Application*. IC-BCT 2019. Blockchain Technologies. Springer, Singapore.
- Patel, K., & Das, M.L. (2020). Transcript Management Using Blockchain Enabled Smart Contracts. In Hung, D. & D’Souza, M. (Eds), *Distributed Computing and Internet*

- Technology*. ICDCIT 2020. Lecture Notes in Computer Science, vol. 11969. Springer, Cham.
- Petkovi', D. (2016). *Microsoft® SQL Server™ 2016 a beginner's guide* (6th ed.). The McGraw-Hill Companies, United States of America.
- Swamy, J.M., & Parmar, K. (2020). Secure and Decentralized Academic Transcript System Based on Blockchain Technology. In: Vasudevan, H., Michalas, A., Shekokar, N., & Narvekar, M. (Eds.), *Advanced Computing Technologies and Applications*. Algorithms for Intelligent Systems. Springer, Singapore.
- Syerson, B., & Murach, J. (2020). *Murach's SQL Server 2019 for Developers* (Illustrated ed.). R Mike Murach and Associates, Inc, United States of America.