IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 1, January 2020

Policy-Driven Academic Result Computation and Transcription: Ritman University Case

Oryina Kingsley Akputu[1], Kingsley Friday Attai[1], Abel Usoro[2], Adedolapo Omobukola Abiodun[3]

Dept. of Computer Science & Mathematics, Ritman University, Ikot Ekpene, Nigeria[1]

School of Engineering and Computing, University of the West of Scotland, Renfrewshire UK[2]

eClinicalWorks USA Westborough, Massachusetts[3]

Abstract: Important obligations of educational institutions include, creation, management, preservation and transcription of examination records of past and active students. Most universities adopt semi-automated and, in some cases, even manual means of processing academic results. The semi-automated approach of the Ritman University in particular, utilizes Microsoft Excel sheets to produce editable result templates. Examination officers across departments utilize the templates to compute student results. While such tedious approach might work well for other institutions with a few student population and fixed curriculum, it is less productive and laborious for the Ritman University which boasts of rapid growing student population and a dynamic curriculum. Often than not it is difficult to validate whether examination officers do incorporate certain policies of individual departments regarding result computation. Not to talk about the security challenges considering that results must be handled by several staff. In this paper, a new system has been developed to efficiently automate and aid result computation task of the university.

Keywords: Academic Result; Computation; Policy-Driven; CGPA; Rational Unified Process; Multi-Tier, Transcription, Systems

I. INTRODUCTION

Ritman University, like most tertiary institutions around the world, has continued to witness a sharp increase in population of previous and active students; relevant student information processing needs have doubled and redoubled. As a result, the work of academic and administrative staff in charge of computing students' results has often significantly expanded and become very cumbersome. Results computation is a (often continuous) process of combining qualitative and quantitative academic data (records) (e.g. scores, grade points, credit units etc.) into a definite and meaningful information such as statement of result also known as transcripts. Essentially either the student's result or transcript, as it is interchangeably called, should represent an evaluation or measure of their academic performance in terms of course work for a period of study. Often than not institutions or departments use transcripts to check the level of understanding of students in courses (subjects) taught. Prospective employers who wish to offer jobs to graduates may demand their transcripts as well. Graduates intending to further studies to other schools often universities. Thus, without Thus, without an efficient results computation and transcript management system, the aim for which students are evaluated by the institution may not be achieved; any mistake or compromise in the computation process might lead to a very big problem. Customarily, academic departments of universities delegate early tasks of result computation solely to their examination officers; the tasks involve computing the QP (Quality Point), GPA (Grade point Average) and CGPA (Cumulative Grade Point Average) [1]. Essentially, the QP score of the student in any course is obtained by multiplying the value of the grade by the credit hours of the course. TQP (Total QP) of all courses offered is obtained. The GPA is computed by dividing the TQP by the sum of credit hours of all the courses offered for the period under consideration. The CGPA is obtained by dividing the cumulative sum of the TQP over the past years by the cumulative sum of the credit hours over the same period. Details of these computations could be found in Ukem and Onoyom-Ita (2011). The computed results are submitted to the heads of department who manually validate the results; at this point it is challenging to properly check or compare versions of results prepared in the past with the current one under consideration to ensure consistency. Besides, by properly checkmating inputs of all entities involved, not only would human errors be spotted but unique policies or regulations of departments guiding grading schemes would be incorporated as well. Unfortunately, the departments in higher institutions have continued to depend on the final outputs processed by the Examination Officers. Most of these institutions like Ritman University do not have secure automated system that could checkmate the work of the Examination Officers. It is important to note that the Examination Officers process originally raw examination scores into results using-Microsoft Excel - a legacy software. It is important to note that the legacy software is not efficient as it adds to human errors. For instance, certain mathematical functions available to facilitate computation process frequently evolve or even get deprecated over time leading to mistakes in calculation. We opine that current approaches of computing examination results using the legacy

DOI 10.17148/IJARCCE.2020.9132

methods need improvement; introduction of an efficient and accurate automated approach could offer good prospects. This paper presents a case study implementation of the policy-driven systems that handle results computation of students' academic results in Ritman University. The suggested policy-driven approach can be adopted in the academic departments across institutions of learning. By policy-driven we mean, the system can incorporate unique policies and requirements for a given academic department. This work is significance in the fact that computation of students results and departmental reports and remarks on the student performances could be made available not just within the university but also remotely. It has potentials of reducing errors and offer smarter way of checkmating the activities of any fraudulent Examination Officer.

The remaining part of the paper are arranged as follows: section II presents a brief and relevant literature on selected policy frameworks for results computation with a review of previously proposed student information systems. Section III describes the methodology used for the system design; it also covers some requirement specification as well as system design and implementation. Section IV gives further discussion on training of Examination Officers for efficient deployment; in addition, some recommendations are given. Section V concludes the paper.

II. EXISTING WORKS

A. Result Computation in Educational Institutions

Most institutions of learning use academic results to initiate policies for effective educational administration and management. In England and Wales for instance, every institution that runs A-level General Certificate of Education must publish their average academic results in a national 'league table' [2]. The government's justification for such a policy hinges in the fact that it helps parents make choices of schools for their children or wards, based on institutional performance on the league table. In Nigerian institutions the assessment of student performance in examination is not only used to measure institutional performance but also that of teachers or instructors [3]. In most cases, the evaluation takes a shift from institutionally perspective to student centred perspective. In the evaluation, the provision by the NUC (National Universities Commission), stipulates that final grades of student in a semester should consist of two parts, namely, final examination grade and other assessments prior to final examination [3]. Specifically, the policy requires that formative and summative evaluation of students in any course in the universities take 30% and 70% for assessment and test, respectively. For college this is put at 40% and 60% for assessment and test respectively. Much as justification for this policy hinges on efforts to adapt to the focus on student centeredness in teaching and learning, the execution is inconsistent across institutions [3].The inconsistency could be blamed on lack of unified automated framework to interpret policies underlying student assessment and results computation process.

B. Existing Result Computing Systems

Several systems and frameworks have been proposed on student assessment or result computation. Specifically, the work of Beka and Beka, presented an automated result processing system designed to increase throughput and reduce response time, using Java and MySQL. It should be noted that, their work has underlying limitations. One limitation is found in usability constraint; the application is confined to a physical location. In the work, emphasis was placed on speed and accuracy in computing GPA and generating transcripts. A similar implementation to what is used in Ritman University was presented in efforts of Ekpenyong and later work of Ukaoh and Amadin [4], [5]. These works used programming capability of the Microsoft Excel to build result computation frameworks. Specific sets of instructions for computing student results were hard-coded in cells and cell references of the application. While this might be efficient for result computation in schools with a very small population, it may become limited as the student population increases. Other efforts on result computation systems are summarized in Table I.

III. SYSTEM DESIGN METHODOLOGY

The system design methodology uses the Rational Unified Process (RUP) [6]; it is one of the best known software methodologies in wide use today. The Relational Database System (RDBS) and MySQL were utilized for database design model and Database Management System (DBMS) respectively. One great merit of the choice database system is its capability to provide convenient access to data at various levels of aggregation. Other merits include capability for buffer management and query processing or to offer SQL-Extensions [7]. Nevertheless, like many other software development methodologies, we further discuss three developmental activities so important to engineering the suggested system. First is requirement specification. Second is the system design and implementation, which is essentially carried out based on institutional policy specifications on how the system must be produced and managed. The third developmental activity is the database design. The fourth activity pertains to system validation; to ensure it does what the institution or departments want.

A. System Requirements

We group the system requirement in this paper under two categories. One group pertains to functional and non-functional requirements. Example under this group is that the system has to accept raw scores uploaded by the examination officers at input point. The inputted raw scores should be mapped to individual student records in the

DOI 10.17148/IJARCCE.2020.9132

database. Furthermore, GPA or CGPA, as the case may be, should be computed for individual records. At the end, the student result, at the request of the administrator, must be generated in the form of a transcript. Nevertheless, another category of requirement is grouped under institutional policies regulating result computation and transcription in the University.

Table I. Other Efforts on Result Computation System

Author(s)	Work Title	Implementation Tools	Remark
Oyekanmi and Azeez [8]	A Deterministic Finite State Automata (DFA) Approach for Students' Examination Result Grading System (A Case Study of Achievers University Owo)	Adobe Dreamweaver CS6, PHP and MySQL Server	The proposed result processing system used a DFA to provide a proficient analysis for decisions on student results and records.
Udeze, *et al.*[9]	Automated Students' Results Management Information System (SRMIS)	CSS, JavaScript, HTML, PHP, MySQL Server.	A resourceful multifaceted system that manages students' records including results processing and fee payment status
Nwosu and Mamah [10]	Result Processing System for Hand Held Mobile Devices	HTML, PHP and MySQL	A web-based system that sends computed results to respective student's mobile devices.
Ekanem, Ozuomba and Jimoh [11]	Development of Students Result Management System: A case study of University of Uyo	PHP, MySQL, HTML, CSS, JAVASCRIPT	An efficient web-based system similar to Ritman University's suggested system.
Okeke, Enemuoh and Ezenwegbu [12]	Implementation of ICT As a Change Agent in Computing Students Result in Chukwuemeka Odumegwu Ojukwu University	Microsoft visual basic.net, PHP and MySQL Server	A web-based system similar to Ekanem, Ozuomba and Jimoh (2017) with secure and restricted privileges to respective users.
Osagie, and Mallam [13].	Data Analysis and Result Computation (DARC) Algorithm for Tertiary Institutions	Fortran algorithm for Data Analysis and Result Computation (DARC).	A flexible and suitable algorithm to manage, analyse and compute students' results
Ukem and Ofoegbu[14]	A Software Application for University Students Results Processing	Java and MySQL Server	A stand-alone system similar to Lawal (2018) and Beka and Beka (2015)
Okeke, Enemuoh and Ezenwegbu[12]	Implementation of ICT As a Change Agent in Computing Students Result in Chukwuemeka Odumegwu Ojukwu University	Microsoft visual basic.net, PHP and MySQL Server	A web-based system similar to Ekanem, Ozuomba and Jimoh (2017) with secure and restricted privileges to respective users.

- Functional and Non-Functional Requirements

In order to elicit actual user needs, requirement analysis was carried out with the objective of gathering desirable features of the suggested system. The functional requirements in this paper describe the behaviours of the system. Essentially the functional requirements make provision according to various user types namely, students, the administrators (Heads of department (HOD), Examination officer, examination and record officer). Specifically, one policy requires facility for modification of the student's courses. Another policy requires modification of the student's course. More system behaviour is summarized under functional requirement in Table II; other requirements which are more technically demanding appear in selected use cases under system design section. The non-functional requirements help in achieving the objectives and improvement of the functionalities of the system. Table III summarizes some of what constitute the non-functional requirement of the system.

- Institutional policies requirement

Besides functional and non-functional requirements, there are certain policies enforced by various institutions regulating the result computation and transcription. We report some selected policies in Ritman university. One policy stipulates access privileges for returning students. A student who has paid their departmental fees and as well as student association charges should be allowed to access and update the student account. To do so, it requires the student having privileges to explore certain functionalities of the system, like view results and check registered courses.

DOI 10.17148/IJARCCE.2020.9132

B. System Design and Implementation

Since the suggested system is a multitier (web-based) application, the Three-Tier framework was adopted as the design architecture. Choice of this framework is informed by its reported offer for openness, scalability, reliability, flexibility and integration[15]; it also offers merits in its re-usability and security features. Making change in one layer often has no effect on other layers; so, less work is required for maintenance and error correction. Subsections below explain the implementation of various layers in the system architecture.

- The System Design

The system design comprises of three sub-tiers viz. client tier (presentation layer), Web tier or Middle layer, and Data layer. In line with system specifications, the presentation layer lies on the top of the application development hierarchy for reception of client requests through web browsers (using URL -Universal Resource Locator) or data entered into a form. Essentially this layer fetches query results according to the nature of the request and coordinates communication among tier layers and the user. In practice the Middle layer consists of a web server which transmits requests between the client layer and data layer. However, this paper reports design on a local host machine via XAMPP server which will be discussed under implementation subsection below. The data layer which essentially comprises of database server represents the third layer of the system design. Using the UML (Unified Modelling Language), we report selected system use cases that reflect how users interact with the system's functional requirement Fig. 1 shows a model use case for three main actors in the systems. Altogether, there are 3 administrators, the Examination Officer, the HOD and the Registrar who is signatory to student transcripts. The Examination and record unit is only permitted to store and secure records, hence can only print out already validated results. Fig. 2 shows UML activity diagram which reflects user flow of interaction between various system objects and how these are interlinked. In general, the administrative user can add and manage students, study level, session, programs and many more activities as shown in the Fig. 2. The fact that different users are given access rights based on level of authority also implies the action each is privileged to perform varies. While both the HODs and the University Registrar have administrative privileges to validate computed results, the latter can sign the transcripts also.

- The System Implementation:

A single all-in-one XAMPP stack which is an open source solution was used to implement all core technologies in each layer of the multitier application, namely, Apache, MySQL, and PHP. Even though Apache, MySQL, and PHP could be installed separately, it makes more sense using the all-in-one stack option for its offers of centralized control over individual components. Besides, in XAMPP, the main components are preconfigured to facilitate communication among various layers. Three front-end technologies were used for the presentation layer namely, HTML, CSS and JavaScript. For the middle layer, the Apache component of the XAMPP is an inbuilt web server. The backend functionalities of the Data layer were powered by PHP and MySQL which manipulate the database on the web server. The Visual Studio Code was employed as a code editor for scripting and debugging presentation layer languages (i.e. HTML, CSS and JavaScript). It has a rich ecosystem of extensions to PHP. Fig. 3 shows a screen shot for the administrator's interface of the system. The student can also interact with the system only after been added by the administrator. Moreover the student interacts under limited privileges: viewing courses and results only.

Table II. Some Functional Requirement

Functional Requirement	Importance	Description
Security	Essential	User authentication is required.
Web Interface	Essential	Provides interactions between user and database.
Database	Essential	Stores necessary information and computed results
File System	Essential	Stores results and transcripts.
Search	Required	Search for students, results and transcripts.

Table III. Some Non-Functional Requirement

Non-Functional Requirement	Importance	Description
Security	Essential	User authentication is required.
Web Interface	Essential	Provides interactions between user and database.
Database	Essential	Stores necessary information and computed results
File System	Essential	Stores results and transcripts.
Search	Required	Search for students, results and transcripts.

DOI 10.17148/IJARCCE.2020.9132

IV. FURTHER DISCUSSIONS

The computation of results in the suggested system involves the following principal entities: Examination officers, HODs, Examination and record unit, and the university Registrar. Nevertheless, the student could also be granted limited access for viewing results and registered courses. It should be noted however, that the suggested system for the university cannot replace the existing approach unless it is approved by the university council which has not yet done so. Even when such approval is granted, we suggest that a parallel conversion strategy be adopted by the institution. It is important to compare merits and demerits of the new and old system before the older is completely faced out if need be. Besides system conversion we further discuss other two key issues below that would enhance optimal usage of the system.

A. Training Examination Officers

To achieve optimal use of the suggested system, the computation staff of various departments of the institution need practical training on how to use the system. Training sessions should be designed to cover the purpose and essence for which the system was designed. Specifically, the trainers could point out the nature of the system, its workability as well as necessary maintenance measures that must be taken. They should in addition point out the type of data and format of input as well as every other aspect that concerns the suggested system.

B. Recommendations

After full testing and evaluation, the suggested system could be recommended for higher institution of learning with growing student populations. Such institution may likely experience difficulties in computation and management of student result. Other corporate bodies should encourage to either employ or adopt the suggested approach and modify according to their special requirements for managing student data.

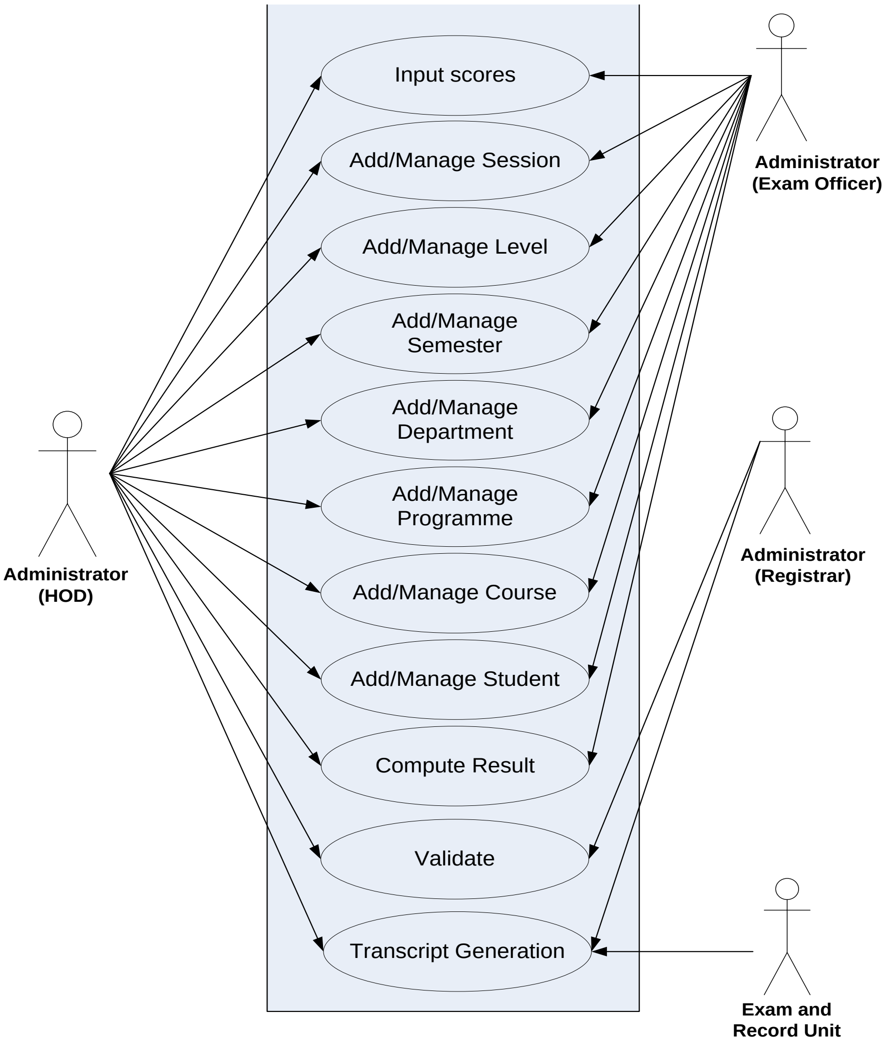


Fig. 1. Use case for Administrators actors

DOI 10.17148/IJARCCE.2020.9132

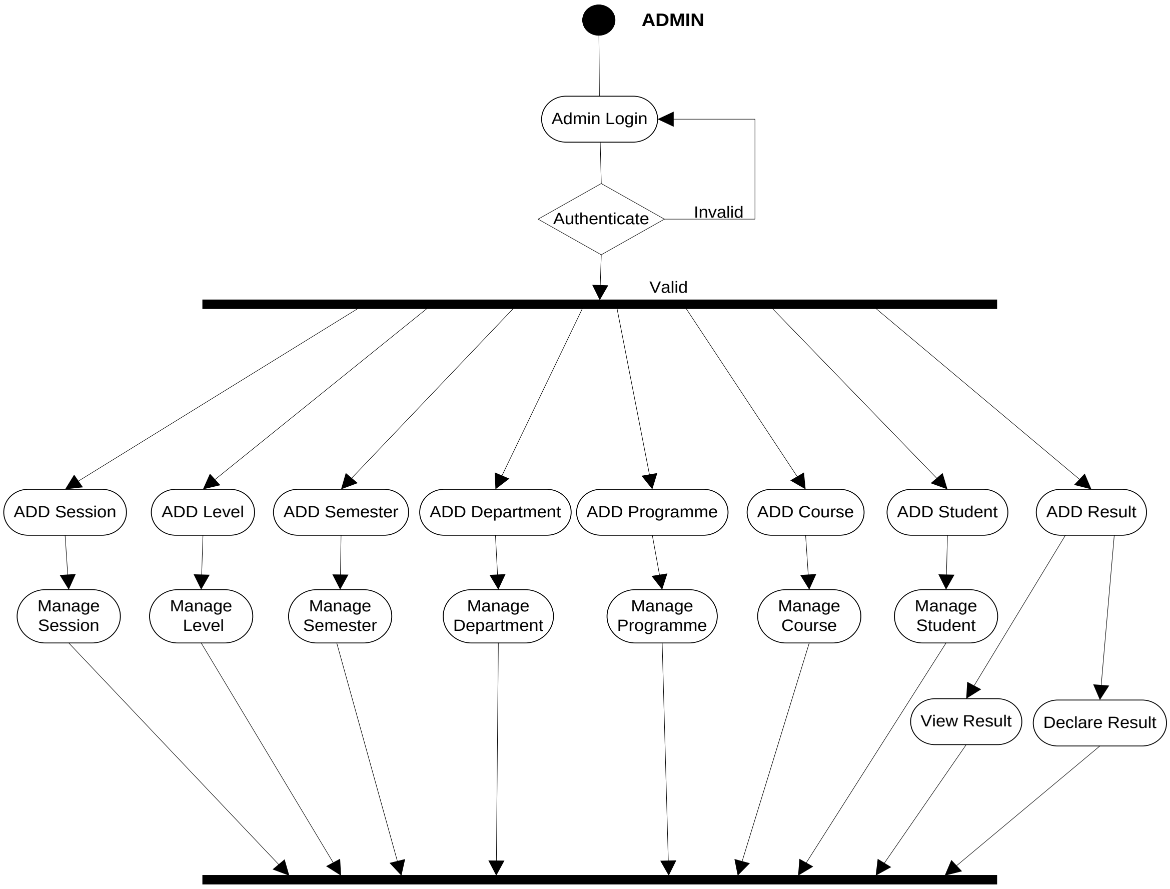
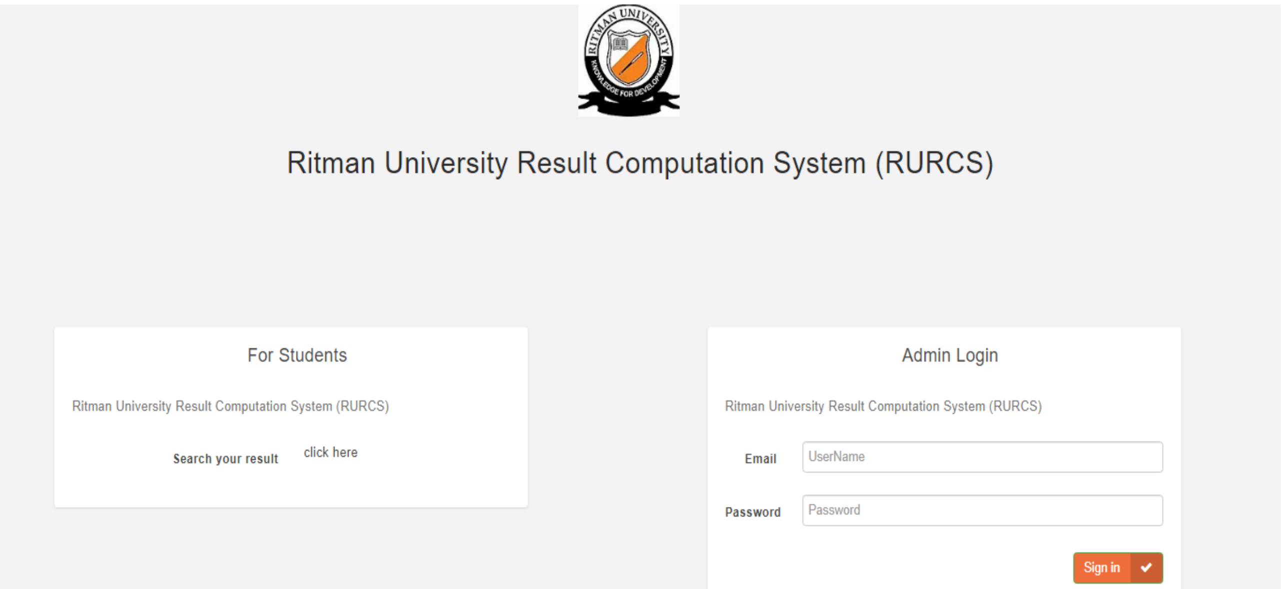


Fig. 2. UML Activity Diagram for Result Computation



Ritman University Result Computation System (RURCS)

For Students

Ritman University Result Computation System (RURCS)

Search your result click here

Admin Login

Ritman University Result Computation System (RURCS)

Email UserName

Password Password

Sign in

Fig. 3. Screen shot of main admin area

V. CONCLUSION

We have presented a secure policy-based academic result computation and transcript generation system. The paper has reviewed existing systems in the literature and that currently in used at the Ritman university. Several difficulties in computing and handling the student results have been identified. Among these difficulties is the inability to protect confidentiality and security of the results in addition to tediousness and error prone processes. The suggested system is flexible, runs on the web and has prospects of alleviating most of the limitation in the exiting systems. In terms of integrity of results processed, the system is reasonably secure as it enforces data integrity from a RDBMS, MySQL. The system design promises minimizing data redundancy in addition to being user-friendly. Additionally, there are other attractive merits offered by the suggested system such as automation of the computation task, thereby reducing result processing time with even improved accuracy. The suggested system is recommended for academic institutions with growing student populations and with a demand for proper student information management policy.

REFERENCES

[1]. E. O. Ukem and E. O. Onoyom-Ita, "A SOFTWARE APPLICATION FOR THE PROCESSING OF STUDENTS RESULTS," 2011.

[2]. H. Goldstein and S. Thomas, "Using Examination Results as Indicators of School and College Performance In several areas of public service , waiting In education these the publication of Education and Science , elected Conservative government," *J. R. Stat. Soc.*, vol. 159, no. 1, pp. 149–163, 1996.

[3]. S. U. Anyanwu and F. N. Iwuamad, "Student-centered teaching and learning in higher education: Transition from theory to practice in Nigeria," *Int. J. Educ. Res.*, vol. 3, no. 8, pp. 349–358, 2015.

[4]. M E Ekpenyong, "A Real-Time IKBS for students results computation," *Int. J. Phys. Sci.*, 2008.

[5]. K. C. Ukaoha and I. F. Amadin, "Computerized result processing system: A case study of the department of computer science, University of Benin," in *IEEE International Conference on Adaptive Science and Technology, ICAST*, 2015, vol. 2015-January.

[6]. P. Kruchten, *The Rational Unified Process: An Introduction - Philippe Kruchten - Google Books*, 2nd ed. Addison-Wesley Professional, 2000.

[7]. C. Bell, *Expert MySQL*, vol. 9781430246602. Apress Media LLC, 2013.

[8]. E. O. Oyekanmi and S. I. Azeez, "A Deterministic Finite State Automata (DFA) Approach for Students ' Examination Result Grading System (A Case Study of Achievers University Owo)," *Int. J. Res. Innov. Appl. Sci.*, vol. IV, no. XI, 2019.

[9]. C. L. Udeze, P. U. Umoren, O. H. E, and H. H. Attah, "Automated Students' Results Management Information System (SRMIS)," *J. Multidiscip. Eng. Sci. Technol.*, vol. 4, no. October 2017, pp. 2458–9403, 2017.

[10]. N. J. Nwachukwu and M. M. Chinedu, "Nwosu and Mamah Result Processing System for Hand Held Mobile Devices," 2019.

[11]. A. J. Ekanem, S. Ozuomba, and A. J. Jimoh, "Development of Students Result Management System: A case study of University of Uyo," Varɛpsilon Ltd, 2017.

[12]. O. Ogochukwu Clementina, E. Chioma Lorretta, and E. Nnamdi Chimaobi, "Implementation of ICT As a Change Agent in Computing Students Result in ChukwuemekaOdumegwuOjukwu University (COOU), in Anambra State, Nigeria," 2016.

[13]. A. U. Osagie, "Data Analysis and Result Computation (DARC) Algorithm for Tertiary Institutions," *IOSR J. Comput. Eng.*, vol. 14, no. 3, pp. 63–69, 2013.

[14]. E.O.Ukem & F.A.Ofoegbu, "A software application for university students results processing," *J. Theor. Appl. Inf. Technol*, vol.35, no.1, 2012.

[15]. J. M. Gallaugher and S. C. Ramanathan, "Choosing a client/server architecture: A comparison of Two-and Three-Tier systems," *Inf. Syst. Manag.*, vol. 13, no. 2, pp. 7–13, 1996.

DOI 10.17148/IJARCCE.2020.9132