

A Deterministic Finite State Automata (DFA) Approach for Students' Examination Result Grading System (A Case Study of Achievers University Owo)

Oyekanmi E.O^{1*}, Azeez S.I²

^{1,2}*Department of Mathematical Sciences, Achievers University, Owo, Nigeria*

Abstract - Analysis of students' information and their academic records are as important as examination result computation. Out of different computer programs in different tertiary institutions for computing examination results and those published, none of them has proffered the analysis of the multi-level aggregated data of student's result with some iota of research model to the best of our knowledge. In this paper, students' result processing software was developed with analysis on the course unit load balancing at the end of an examination period. A deterministic Finite Automata (DFA) mode is used in the assignment of students' grade for each score posted. Personal Home Page Pre-Processor (PHP) with My Structured Query Language (MySQL) were used as programming tools in the development of Students' Result Processing Software (SRPS). The reports of the students' result records presented in this paper satisfies the needs of the students, the course advisers, the department, the faculty and the institution. Ultimately, the model employed in this paper provides a more proficient information and analysis for decisions on student(s) records in tertiary institutions.

Keywords-DFA, Examination Result, Load balancing, One-to-One learning initiatives, software development

I. INTRODUCTION

Computerizing students' result in this dispensation should no longer be a thing of one-to-one but one-to-many in terms of the implementation. The former computing initiative is to help accomplish the academic goals of transition from manual operation to computerized one.

Although, the definition of one-to-one result processing may vary somewhat from location to location but in this paper, the definition is basically on the use of Microsoft Excel for departmental result processing. Strictly speaking, this one-to-one learning initiatives has gained popularity and was relatively the instructional intervening approach for processing students' result, yet, many educational leaders and policy makers have weighed the costs versus the benefits of implementing this learning initiatives using three basic yardsticks which include:

A. The Students

Winnie Hu asserted that some schools have gone so far as to cancel their programs and as such the transfer of the result of the student to another department is difficult because it is a one-to-one initiative. He therefore claims that one-to-one

laptop initiatives do not have a positive impact on student achievement at school [4]. More importantly, result score can be mistyped with one-to-one result computing and an attempt to make such an error could either mar or make the student. Hence, there is need for more robust system that can dynamically do the processing accurately.

B. The Lecturer

Lecturers need not to be moving from one department to another to submit result when it could be done right there in his/her office. The Examination officer of any department needs not to be asking for result from another department, when it can be viewed straight away in his/her office automatically if the result has been posted by the lecturer who handled the course.

The reverse is the case with one-to-one result computing. The input stress of one-to-one result computation is much compare to one-to-many.

C. The Department

Department is free in computing their students' result at their pace and this is one of the benefits of the proposed software.

II. LITERATURE REVIEW

Ukem and Onoyom-Ita developed a web-enabled students' result processing application with the tool similar to what was used in this paper [1]. However, the students' capacity used in the testing was limited (and not properly documented). Emmanuel and Choji also used similar tools but the human user interaction concept was not thoroughly put into consideration while building [2]. This reflected from their recommendation. The security of it was also doubtful. Ukem, and Ofoegbu used Java programming language to build a stand-alone computer for students' result processing [3]. Relational Database Management System was used in the development for backend storage of students' result data. However, it is not web-enabled. Moses used Microsoft Excel Spreadsheet to build an Intelligent Knowledge-based System (IKBS) for students' result computation. It was reported to be working fine [5]. However, it was not reported that it could work centrally for others to access let alone working on either intranet or internet.

All the above have contributed to the knowledge in their own capacity, however none of them has actually focus on the area which was presented in this paper. The result computation application presented in this paper is a real-time implementation, it added more functionality to the existed one, reduced complexity and enhance maintainability while retaining good speed and accuracy. It also include a research model in the computation of the failed course units balancing at every time of attempt and a design for grade computation using a DFA approach.

III. DATABASE DESIGN

SRPS has 39 tables with a table having a minimum of 3 fields and maximum of 11 fields. Normalization is put into consideration for optimization, flexibility and easy debugging. The tables took care of immediate past records of each students currently in the institution or department as at the time when the software would be installed for easy computation of their previous cumulative. Few out of the tables are discussed in Table I-IV.

Table I shows the registration table for every students whose records will be stored in the database. The field with * shows the student's matric number and is used as the primary key.

TABLE I STUDENT'S ADMISSION LIST TABLE

| | |
|-----------|-----------|
| utmeid | vchar(16) |
| fullname | vchar(20) |
| *matricid | vchar(12) |
| email | vchar(35) |
| phoneno | vchar(15) |
| session | vchar(9) |
| savedate | timestamp |

Table II stored the registration of three (3) different categories of staff that can use the software. The staff is either a lecturer which is the default for all, while some are advisors or Chief-exam officers. Once a staff login, a mutex is set for the staff to enforce a mutual exclusion concurrency control policy.

TABLE II LECTURER'S REGISTRATION TABLE

| Field | Type | Comments |
|--------------|-----------|--|
| *stafflogid | vchar(15) | |
| stafflogpass | vchar(12) | |
| logclass | int(1) | 1-Lecturer; 2-ChiefExam Officer; 3-Advisor |
| staffmutex | int(11) | 0->(free); 1->(Not Free) |
| savedate | timestamp | |

Table III stored the registration of each lectured-courses by lecturers. The ** is a foreign key which could be more than one in this table. In other words, a lecturer can be allocated more than one course in a semester.

TABLE III LECTURER'S COURSES TABLE

| Field | Type | Comments |
|--------------|-----------|-------------|
| id | int(11) | Primary Key |
| **stafflogid | vchar(15) | |
| leccode | vchar(7) | |
| courselev | int(3) | |
| lecseme | int(1) | |
| lecssess | vchar(9) | |

Table IV is used in storing the academic records of each students per semester and session. The score is automatically zero (0) once a course is registered by a student. The zero is changed after the lecturer posted the overall score percentage value.

TABLE IV STUDENTS' SCORE TABLE

| Field | Type | Comment |
|------------|-----------|-------------|
| id | int(11) | |
| **matricid | vchar(15) | Foreign key |
| ccode | vchar(7) | |
| cunit | int(1) | |
| cscore | int(2) | |
| cssess | vchar(9) | |
| cseme | vchar(6) | |
| savedate | timestamp | |

IV. ARCHITECTURE DESIGN

Fig. 1 shows the architecture of the intra-connectivity of hardware involved. Although, the software was a web-based application which can be used in the form of stand-alone or online. The installation of SRPS is done on the intranet server and wireless switch is used to distribute it to other remote / intra computers for easy access. All transactions are automatically done in the intra-server from where the reports needed are generated.

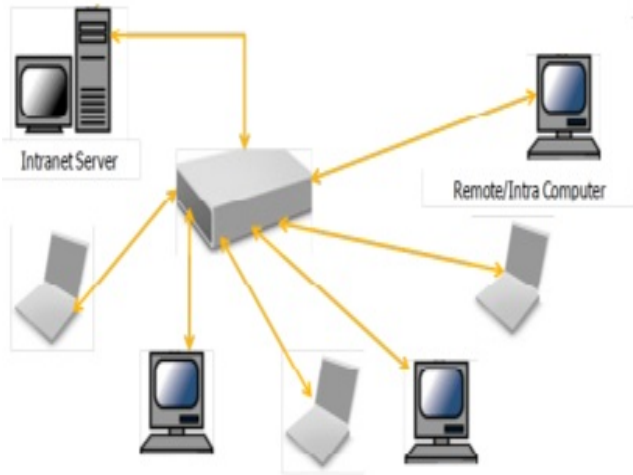


Fig. 1 Intra-communication Architecture of SRPS

- 1) *Tools Used* : Adobe Dreamweaver CS6, an Integrated Development Environment (IDE), is used to create the Graphic User Interface and to write the codes. PHP with Structured Query Language (SQL), pronounced as 'Sequel-Language', were used to communicate with MySQL Server - a Relational Database Management System at the back-end. Other supporting languages used were Jquery, Cascade Style Sheet for the robustness of the software.
- 2) *Wampserver Configuration*: Apache Version : 2.2.11 , PHP Version : 5.3.0

V. SOFTWARE DESIGN

The software has four sections and each sections has their respective modules. Some sections have some modules in common while others do not. However, each has a LOGIN page in common.

Fig. 2 is the index page for result processing. This page is used by the advisor, exam-officer or lecturer to login to their respective platform for any result transactions. The login page automatically display the respective page after the credentials are authenticated.

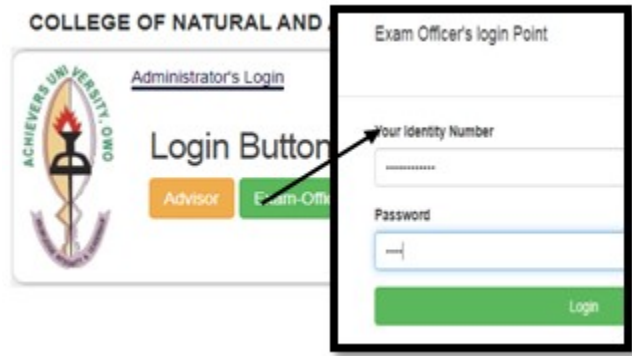


Fig. 2 Index page for result processing

Fig. 3 shows the flowchart of the credentials authentications for the three (3) categories of staff. Each staff's page has permitted and, of course, major transaction details.

VI. DETERMINISTIC FINITE STATE AUTOMATA (DFA) FOR GRADING

DFA is quintuple $A = (Q, \Sigma, \delta, q_0, F)$ where

Q is set of states

Σ is the alphabet (of input symbols)

$\delta: Q \times \Sigma \rightarrow Q$ is the transition function

q_0 The start state

$F \subseteq Q$ Final states

DFA is usually represented using both graph and table. The graph is a directed graph whose nodes are states and whose arcs are labelled by one or more symbols from alphabet Σ , this graph is called a state transition diagram. Also, all the information presenting a DFA can be given by a single table-transition. In transition table, the initial and final states are denoted by \rightarrow and $*$ respectively. It is possible that DFA accepts or reject a language depending on the transaction that it involves.

Mathematically, DFA can be defined as:

$A = (Q, \Sigma, \delta, q_0, F)$ accepts a string w , iff $\delta(q_0, w) \in F$

The language of the automata A is

$L(A) = \{w | A \text{ accepts } w\}$

More formally,

$L(A) = \{w | \delta(\text{start}(A), w) \in \text{Final}(A)\}$

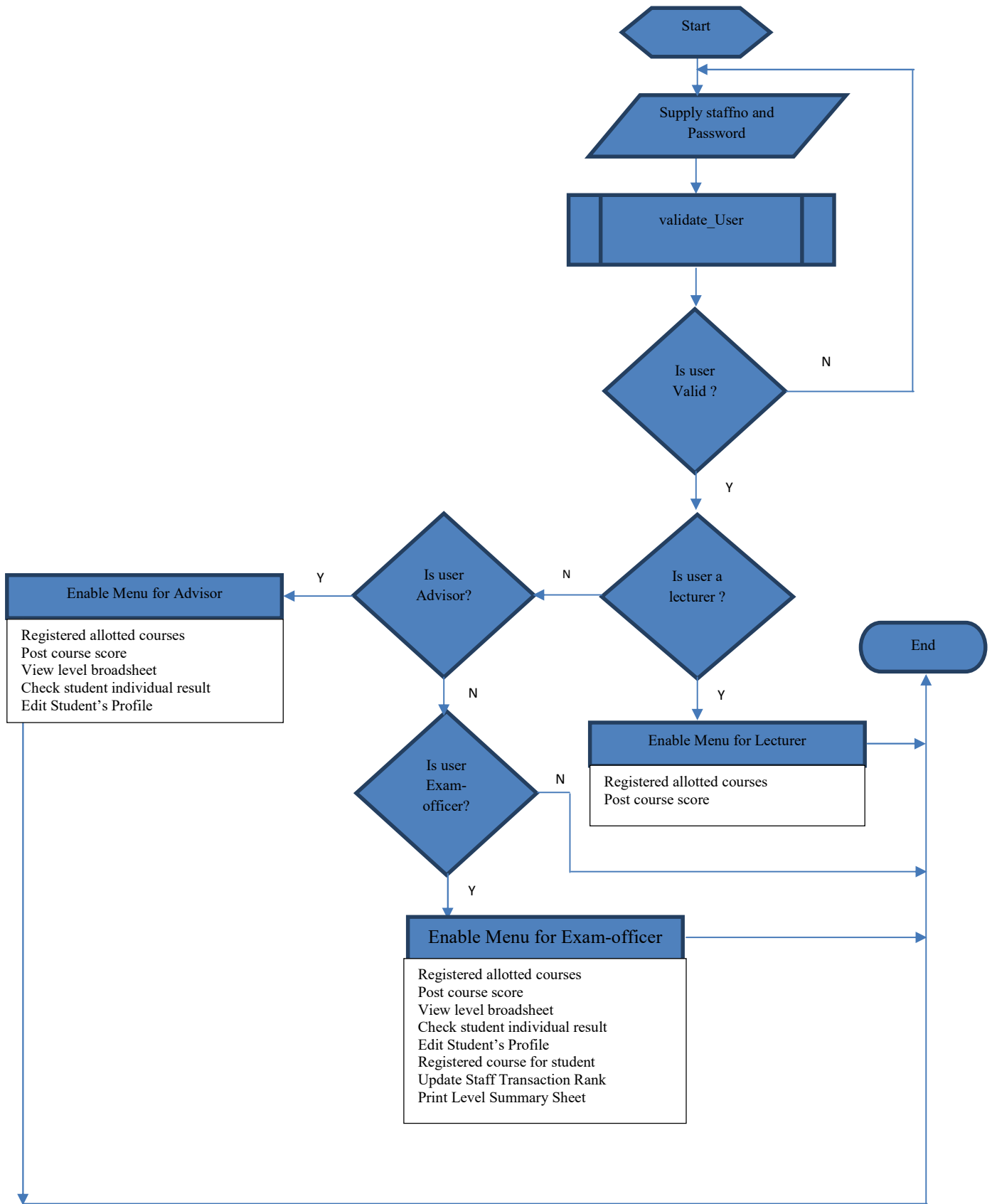


Fig. 3 Flowchart of SRPS authentication process

The Implementation of the Model for Unit Load balancing

A student’s result can be computed if and only if he/she registered some units per semester. If a student registered a course, such course has two state values: 1 and 0 which are associated with ‘Pass’ and ‘Fail’. If the course is passed, it will be accumulated to his/her cumulative unit passed, otherwise it will remain failed until it is passed.

In this paper, the pass mark for a registered course is assumed to be greater or equal to 45 (scores >= 45). Any mark below 45 resulted to failure. Total unit register (TUR), Total unit passed (TUP) and Total unit failed (TUF) are used in the course system for smooth computation of students’ examination result.

The balancing comes in when

$$TUR = TUP + TUF \tag{1}$$

In order words,

$$TUR - TUP = TUF \tag{2}$$

Two strings is used as the valid input before a final state is attained. The lowest strings is ‘00’ while the highest is ‘99’. The assumption is that the highest score a student can have is 99. This approach is used to reduce the number of transition states before the final state. This input string is therefore represented as

$$\Sigma = \{0, 1, 2, \dots, 9\} \tag{3}$$

Since the alphabet is from 0 to 9, a combination of two out of these alphabets is taking as score value entered by a lecturer.

$$Q = S, U, V, W, X, Y, A, B, C, D, F$$

At

$$S, \Sigma = \{0-9\}$$

$$U, \Sigma = \{0-3\}$$

$$V, \Sigma = \{4\}$$

$$W, \Sigma = \{5\}$$

$$X, \Sigma = \{6\}$$

$$Y, \Sigma = \{7-9\}$$

Table V shows the transition table of the input strings into the finite state machine. After the initial state S, there are 5 other transition states before getting to any of the other 5 final states. In other words, $\{S, U, V, W, X, Y\} \in Q$ and $\{A, B, C, D, F\} \in F$. Fig. 4 shows the transition diagram that was used in getting this solution

TABLE V COURSE SCORE TO GRADE TRANSITION TABLE

| $\Sigma \backslash Q$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|-----------------------|---|---|---|---|---|---|---|---|---|---|
| →S | U | U | U | U | V | W | X | Y | Y | Y |
| U | F | F | F | F | F | F | F | F | F | F |
| V | F | F | F | F | F | D | D | D | D | D |
| W | C | C | C | C | C | C | C | C | C | C |
| X | B | B | B | B | B | B | B | B | B | B |
| Y | A | A | A | A | A | A | A | A | A | A |
| *A | S | S | S | S | S | S | S | S | S | S |
| *B | S | S | S | S | S | S | S | S | S | S |
| *C | S | S | S | S | S | S | S | S | S | S |
| *D | S | S | S | S | S | S | S | S | S | S |
| *F | S | S | S | S | S | S | S | S | S | S |

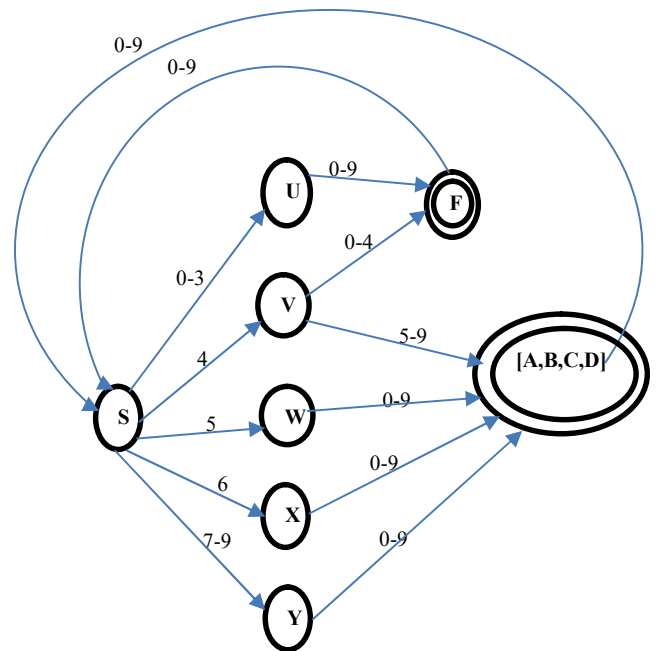


Fig. 4 Course score transition diagram

Appendix I shows Table VI-IX and Appendix II shows Table X-XI.

Table VI shows a typical registered courses by a student. In this table, at year 2014/2015, the student registered 15 units which include 6 different courses at First semester and 15units at second semester containing 5 different courses. The summary shows that 30 units were registered with 22units passed and 8units failed. The failed courses were emphasized with * at the end of the course code.

In Table VII, the same student registered 30 units in the following session and two (2) carried over courses were redone out of three (3) that the student failed in the previous year. These were included within the registered units. Again, PHY101 (3) is re-failed by the student while the other courses registered were passed. The summary here, shows that 60 units has been done so far with 49 units passed and 11 units failed.

Table VIII shows the third year spent by the same student. CIS101, PHY101 were both redone in this year together with other courses. In summary, 91 units have been done so far, with 79 units passed and 12 units failed.

Table IX-XI show the report of how the carried over courses are captured at each semester in the year 2014/2015 to 2016/2017. The total cumulative unit passed also reflected. The failed units are computed by adding each unit of the failed courses together. The sum of the failed units with the cumulative total units passed (CTUP) are added together to form cumulative total units registered (CTUR). In this report, the failed courses were displayed with the unit weight and number of time that is repeated.

VII. CONCLUSION

A robust students' result computation software have been developed which can compute student's Gross Point Average (GPA), Cumulative Gross Point Average (CGPA), Cumulative Total Unit Pass (CTUP) and Cumulative Total

Unit Fail (CTUF). A DFA machine was also designed to get two (integer value, one at a time as) input alphabets which were then processed to form the equivalent grade for such an input. The designed finite machine was optimized by using two strings instead of three in the case of 100 and implementation of combined 4 grades as one final state. Any other value after the two strings is send back to start state. Lastly, load balancing of the units' computation was done based on TUR, TUP and TUF.

REFERENCES

- [1] Ukem O. and Onoyom-Ita E. "A Software Application for the Processing of Students Results", Global Journal of Pure and Applied Sciences Volume 17 No. 4 (2011)
- [2] Emmanuel B. and Choji D. "A Software Application for Colleges of Education Students Results Processing", Journal of Information Engineering and Applications Vol. 2, No. 11, (2012)
- [3] Ukem E. and Francis A. "A Software Application for University Students Results Processing", Journal of Theoretical and Applied Information Technology Vol. 35 No. 1 (2012)
- [4] Hu W. "Seeing no progress, some schools drop laptops". The New York Times. Retrieved from <http://www.nytimes.com/2007/05/04/education/04laptop.htm>, May 7 (2007)
- [5] Moses E. "A Real-Time IKBS for Students Results Computation", International Journal of Physical Sciences (Ultra Scientist of Physical Sciences) Volume 20, Number 3(M), September – December(2008).

APPENDIX I

Table VI Registered Courses of a typical student with matric A in year 2014/2015

| First Semester | | | | | | |
|-----------------|--------|-----------|-----------|--------|---------|--------|
| Course Code | BIO101 | CIS101* | CHM101 | CIS107 | PHY101* | GNS101 |
| Course Unit | 3 | 2 | 3 | 3 | 3 | 1 |
| Score | 45 | 37 | 45 | 50 | 30 | 46 |
| Second Semester | | | | | | |
| Course Code | BIO102 | CIS102 | CHM102 | BMT112 | PHY102* | |
| Course Unit | 3 | 3 | 3 | 3 | 3 | |
| Score | 70 | 45 | 60 | 56 | 35 | |
| Summary | | | | | | |
| CTUR = 30 | | CTUP = 22 | CTUF = 08 | | | |

Table VII Registered Courses of a typical student with matric A in year 2015/2016

| First Semester | | | | | |
|-----------------|-----------|-----------|--------|--------|--------|
| Course Code | CIS201 | CIS203 | CIS205 | PHY101 | PHY201 |
| Course Unit | 3 | 3 | 3 | 3 | 3 |
| Score | 45 | 45 | 56 | 35 | 53 |
| Second Semester | | | | | |
| Course Code | CIS202 | CIS204 | CIS206 | PHY102 | ESP222 |
| Course Unit | 3 | 3 | 3 | 3 | 3 |
| Score | 50 | 50 | 78 | 50 | 47 |
| Summary | | | | | |
| CTUR = 60 | CTUP = 49 | CTUF = 11 | | | |

Table VIII Registered Courses of a typical student with matric A in year 2016/2017

| First Semester | | | | | | |
|-----------------|-----------|-----------|--------|--------|--------|--------|
| Course Code | CIS101 | PHY101 | CIS301 | GNS111 | CIS303 | CIS305 |
| Course Unit | 2 | 3 | 3 | 2 | 3 | 3 |
| Score | 38 | 45 | 60 | 62 | 58 | 50 |
| Second Semester | | | | | | |
| Course Code | CIS302 | CIS304 | CIS306 | CIS308 | CIS310 | |
| Course Unit | 3 | 3 | 3 | 3 | 3 | |
| Score | 50 | 50 | 50 | 60 | 62 | |
| Summary | | | | | | |
| CTUR = 91 | CTUP = 78 | CTUF = 13 | | | | |

Table IX Summary report of student with matric A in year 2014/2015

| First Semester | | | | | | | | | | | | |
|------------------|-----|-----|-----|-------------------|------|------|-------|--------------|-------|-------|-------|---|
| Current Semester | | | | Previous Semester | | | | Current CGPA | | | | REMARK |
| TUR | TUP | TWP | GPA | PTUR | PTUP | PTWP | PCGPA | CTUR | CTUP | CTWP | CCGPA | |
| 15 | 10 | | | 0 | 0 | | | 15 | 10 | | | CO:CIS101(2R), PHY101(3R) Total Units: 5 |
| Second Semester | | | | | | | | | | | | |
| Current Semester | | | | Previous Semester | | | | Current CGPA | | | | REMARK |
| TUR | TUP | TWP | GPA | PTUR | PTUP | PTWP | PCGPA | CCTUR | CCTUP | CCTWP | CCGPA | |
| 15 | 12 | | | 15 | 10 | | | 30 | 22 | | | CO:CIS101(2R), PHY101(3R), PHY102(3R) Total Units: 8 |

APPENDIX II

TABLE X SUMMARY REPORT OF STUDENT WITH MATRIC A IN YEAR 2015/2016

| FIRST SEMESTER | | | | | | | | | | | | |
|------------------|-----|-----|-----|-------------------|------|------|-------|--------------|-------|-------|-------|--|
| Current Semester | | | | Previous Semester | | | | Current CGPA | | | | REMARK |
| TUR | TUP | TWP | GPA | PTUR | PTUP | PTWP | PCGPA | CCTUR | CCTUP | CCTWP | CCGPA | |
| 15 | 12 | | | 30 | 22 | | | 45 | 34 | | | CO:CIS101(2R), PHY101(3R2), PHY102(3R) Total Units: 11 |
| SECOND SEMESTER | | | | | | | | | | | | |
| Current Semester | | | | Previous Semester | | | | Current CGPA | | | | REMARK |
| TUR | TUP | TWP | GPA | PTUR | PTUP | PTWP | PCGPA | CCTUR | CCTUP | CCTWP | CCGPA | |
| 15 | 15 | | | 45 | 34 | | | 60 | 49 | | | CO:CIS101(2R), PHY101(3R2), PHY102(3R) Total Units: 11 |

TABLE XI SUMMARY REPORT OF STUDENT WITH MATRIC A IN YEAR 2016/2017

| FIRST SEMESTER | | | | | | | | | | | | |
|------------------|-----|-----|-----|-------------------|------|------|-------|--------------|-------|-------|-------|---|
| Current Semester | | | | Previous Semester | | | | Current CGPA | | | | REMARK |
| TUR | TUP | TWP | GPA | PTUR | PTUP | PTWP | PCGPA | CCTUR | CCTUP | CCTWP | CCGPA | |
| 16 | 14 | | | 60 | 49 | | | 76 | 63 | | | CO:CIS101(2R2), PHY101(3R2), PHY102(3R) Total Units: 13 |
| SECOND SEMESTER | | | | | | | | | | | | |
| Current Semester | | | | Previous Semester | | | | Current CGPA | | | | REMARK |
| TUR | TUP | TWP | GPA | PTUR | PTUP | PTWP | PCGPA | CCTUR | CCTUP | CCTWP | CCGPA | |
| 15 | 15 | | | 76 | 63 | | | 91 | 78 | | | CO:CIS101(2R2), PHY101(3R2), PHY102(3R) Total Units: 13 |