

Implementation of an Effective Course Registration Credit Unit Monitoring System

*¹Godwin, A. Otu, ¹ Stephen, E. Iheagwara, ³Fredrick, I. Okonkwo, ²Suleiman, A. Usman, ¹Modupe, S. Oyebanji

Page | 4762

¹Department of Computer Science, Air Force Institute of Technology Kaduna, Nigeria.

² Department of Cyber Security, Air Force Institute of Technology Kaduna, Nigeria

³Department of Information and Communication Technology, Air Force Institute of Technology Kaduna, Nigeria.

Date Received: 09-07-2023

Date Accepted: 17-11-2023

DOI: <https://doi.org/10.48198/NJPAS/23.B08>

ABSTRACT

Automation of course registration is a major breakthrough with the advancement in information technology. Registration processes that take longer time most times eating deep into the semester and with laborious inputs from registration officers have evolved into a seamless and easy process due to the development of robust information systems. An information system requires accurate data for it to give a reliable output, so it is of the essence to be able to input accurate credit units of courses to the course registration system. Computing the credit units of courses by students and course registration officers is mostly manual, in this research, automation of this vital pre-process is implemented. The system is built using java programming language which makes it robust, simple and easy to use. It is a credit unit calculator that does not require being online before it can either be deployed or used. The system prompts the user to enter level, the semester, maximum credit load for the semester and their corresponding credit units. At each execution of the system the courses, corresponding credit units, total credit load for the semester is displayed provided the total credit units entered is not greater than the credit load for the semester, otherwise a prompt displays the excess credit units. Subsequent research can enhance the system to be able to send the output of the pre-registration document from the student directly to the registration officer, and also there should be a module embedded to the system that will enable the registration officer to perform the pre-registration document verification. A module to check double registration and determination if a given course is either first or second semester can also be added.

Keywords: Credit unit, Credit load, Information system, Course registration, Information technology.

Introduction

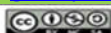
The portal used to carry-out student registration encompasses many modules which perform varied processes. This student registration system streamlines the application, registration, and monitoring of students in an institution. According

to Chaka (2013), online registration systems are platforms that enables users to sign up for memberships, events and training by filling a form. Online registration systems are now setting aside manual processes, such as registering by telephone, mail or at events using hardcopy forms.

Corresponding Author: Prof. Godwin A. Otu

Department of Computer Science, Air Force Institute of Technology Kaduna, Nigeria.

goddy2fine@gmail.com



The course registration exercise is an important academic activity that if done properly will be a fundamental step that leads the student to academic success. At the inception where the student is offered admission it might be a less complicated exercise, but as the student progresses to higher level it becomes complicated because of carryovers and elective courses. These courses must be registered either such that the credit units of the registered courses do not exceed the maximum credit load or carryover courses are registered first.

Many researches have focused on the need to develop online registration systems to automate registration processes and reduce paper work as well as the work of course advisers and registration officers. Chaka (2013) designed and implemented a system with an efficient database for storage that automates registration processes and makes the process accessible to the students regardless of their location. Most of these systems developed are built with the aim of automation with little emphasis on either the improvement or ease of use-ability of the system. This research on solving the credit unit problem during course registration process because it saves students from registering below or under the expected credit unit for a semester. The system, which will be easy to use and can work online and offline, will identify if the credit load is exceeded during registration and compute the excess credit unit(s), as the case may be. The credit unit monitoring system unlike other student registration systems is not a web application, but an independent module that will produce an advisory pre-registration document which the student can use to perform the actual course registration. It can be dub a credit unit calculator.

The need to study portals used for registration in tertiary institutions specifically their respective modules and making informed decisions about how the portal is to be improved it is an effective way of improving the system. This approach enhances the functionality of the entire system,

proffers low-cost reduction during system maintenance and also not altering system architecture which is inevitable if an entire overhaul is done. Many researches, designed, documentation and implementation have been made daily to enhance registration. Developers keep turning in their techniques for registration systems due to the fact that these systems are web applications and are also built using web tools which turned out to be changing daily because of the dynamic nature of the web. It is of essence to look at some of the strides done by most researchers.

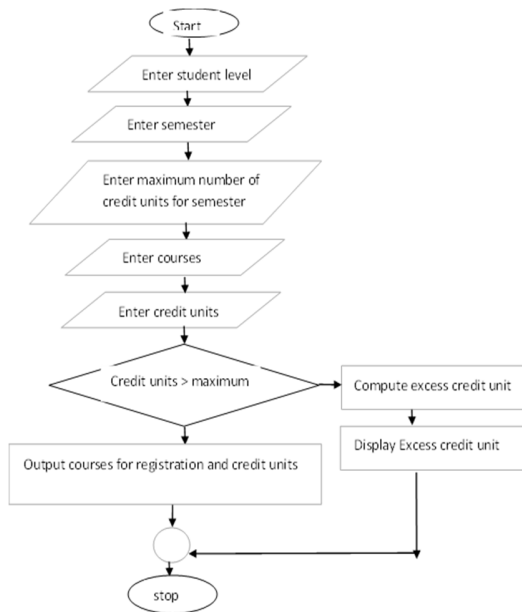
The system being develop in this research is an independent sub-system that will only assist student to specifically identify the courses they can register in a semester. The output can also assist course level advisors to guide the students on the right decisions to take in their academic pursuit. And also since it is not a web base application it can be executed without necessarily going online.

Materials and Methods

Algorithm

1. Start
2. Enter student level
3. Enter semester
4. Enter maximum credit units for the semester
5. Enter courses
6. Enter credit units
7. If Credit units > maximum credit units
8. Prompt “ you cannot register the courses due to excess unit”
9. Compute and display excess units to be dropped
10. Else
11. Display courses to be registered and their respective credit units
12. Stop

FLOWCHART



Implementation

```

package myLatest;

import java.util.Scanner;

public class Demonstration {

    public static void
main(String[] args) {

        System.out.println(" enter
1 for one hundred level");
        System.out.println(" enter
2 for two hundred level");
        System.out.println(" enter
3 for three hundred level");
        System.out.println(" enter
4 for four hundred level");
        System.out.println(" enter
0 to exit the application");
        Scanner keyboard = new
Scanner(System.in);
        int level =
keyboard.nextInt();
        switch (level) {

            case 1:

monitorOneLevelCreditLoad ();
                break;

            case 2:

```

```

monitorOtherLevelCreditLoad ();
                break;
            case 3:

monitorOtherLevelCreditLoad ();
                break;
            case 4:

monitorOtherLevelCreditLoad ();
                break;
            default:

System.out.println("Invalid
Level");
        }
    }

    public static void
monitorOneLevelCreditLoad () {

        int [] courseCodes;
        String [] courseNames;
        int maximumCreditLoad;
        int numberOfCoursesOffered;
        int creditUnit [];
        int sumOfCreditUnit = 0;
        int regularCoursesOffered;
        int sumOfUnits;
        int level;
        String semester;

        switch (level) {

            Scanner keyboard = new
Scanner(System.in);
            System.out.println(" enter level ");
            level = keyboard.nextInt();
            System.out.println(" enter semester");
            semester = keyboard.next();
            System.out.print(" enter
maximum credit load for semester ");
            maximumCreditLoad =
keyboard.nextInt();
            System.out.println(" ");
            System.out.print(" enter
number of courses offered ");
            numberOfCoursesOffered =
keyboard.nextInt();
            System.out.println();
            courseNames = new String
[numberOfCoursesOffered + 1 ];
            courseCodes = new int
[numberOfCoursesOffered];

```

```
        creditUnit = new int
[numberOfCoursesOffered];
        System.out.println();
        for (int x = 0; x <
courseNames.length; x++) {
            System.out.print("
enter course name ");
            courseNames [x] =
keyboard.nextLine();

            System.out.println();
        }
        for (int x = 0; x <
numberOfCoursesOffered ; x++) {
            System.out.println("
enter course code ");
            courseCodes [x] =
keyboard.nextInt();
        }
        for (int x = 0; x <
numberOfCoursesOffered ; x++) {
            System.out.print("
enter course credit unit ");
            creditUnit [x] =
keyboard.nextInt();

            System.out.println();
            if (creditUnit[x] >
3) {

                System.out.print("re-enter course
credit unit ");
                creditUnit[x]
= keyboard.nextInt();

                System.out.println();
            }
        }
        for (int x = 0; x <
numberOfCoursesOffered ; x++) {

            sumOfCreditUnit =
sumOfCreditUnit + creditUnit[x] ;
        }

        System.out.println("THE COURSES
FOR " + level + " " + semester + "
SEMESTER AND THEIR CORRESPONDING
UNITS");
        for (int x = 0; x <
numberOfCoursesOffered ; x++) {

            System.out.println(courseNames
```

```
[x] + " " + courseCodes [x] + " " +
creditUnit[x] );
        }
        if (sumOfCreditUnit >
maximumCreditLoad ) {
            System.out.println("
exceeded the credit load by "+
(sumOfCreditUnit- maximumCreditLoad) );
        }
    }
}
public static void
monitorOtherLevelCreditLoad () {
    int CarryOvercourseCodes [];
    int CarryOvercreditUnit [];
    String [] CarryOvercourseNames;
    int [] RegularcourseCodes;
    String [] RegularcourseNames;
    int maximumCreditLoad;
    int numberOfCoursesOffered;
    int RegularcreditUnit [];
    int sumOfCreditUnit = 0;
    int sumOfCarryOverUnits = 0;
    int carryOverUnits [];
    int
numberOfCarryOverCoursesOffered;
    int regularCoursesOffered;
    int sumOfUnits;
    int level;
    String semester;
    Scanner keyboard = new
Scanner(System.in);
    System.out.println(" enter level
");
    level = keyboard.nextInt();
    System.out.println(" enter
semester");
    semester = keyboard.next();
    System.out.println(" enter
maximum credit load for the semester ");
    maximumCreditLoad =
keyboard.nextInt();
    System.out.println(" enter
number of carry over courses offered ");

    numberOfCarryOverCoursesOffered =
keyboard.nextInt();
    CarryOvercourseNames = new
String [numberOfCarryOverCoursesOffered
+ 1 ];
    CarryOvercourseCodes = new
int [numberOfCarryOverCoursesOffered];
    CarryOvercreditUnit = new
int [numberOfCarryOverCoursesOffered];
```

4766

```
        carryOverUnits = new int
[numberOfCarryOverCoursesOffered];
        for (int x = 0; x <
numberOfCarryOverCoursesOffered ; x++) {
            System.out.println("
enter carry over course name ");
            CarryOvercourseNames
[x] = keyboard.nextLine();
        }
        for (int x = 0; x <
numberOfCarryOverCoursesOffered ; x++) {
            System.out.println("
enter carry over course code ");
            CarryOvercourseCodes
[x] = keyboard.nextInt();
        }

        for (int x = 0; x <
numberOfCarryOverCoursesOffered ; x++) {
            System.out.println("
enter course credit unit ");
            carryOverUnits[x] =
keyboard.nextInt();
            if
(carryOverUnits[x] > 3) {

                System.out.println("re-enter
course credit unit ");

                carryOverUnits[x] =
keyboard.nextInt();
            }

            sumOfCarryOverUnits
= sumOfCarryOverUnits +
carryOverUnits[x] ;

        }

        System.out.println(" enter
number of regular courses offered ");
        regularCoursesOffered =
keyboard.nextInt();
        RegularcourseNames = new
String [ regularCoursesOffered];
        RegularcourseCodes = new
int [regularCoursesOffered];
        RegularcreditUnit = new int
[regularCoursesOffered];
        for (int x = 0; x <
regularCoursesOffered ; x++) {
            System.out.println(" enter
regular course name ");
            RegularcourseNames [x] =
keyboard.nextLine();
```

```
        }
        for (int x = 0; x <
regularCoursesOffered ; x++) {
            System.out.println(" enter
regular course code ");
            RegularcourseCodes [x] =
keyboard.nextInt();
        }
        for (int x = 0; x <
regularCoursesOffered ; x++) {
            System.out.println(" enter
course credit unit ");
            RegularcreditUnit[x] =
keyboard.nextInt();
            if (RegularcreditUnit[x] >
3) {

                System.out.println("re-enter
course credit unit ");
                RegularcreditUnit[x]
= keyboard.nextInt();
            }
        }
        for (int x = 0; x <
regularCoursesOffered ; x++) {
            sumOfCreditUnit =
sumOfCreditUnit + RegularcreditUnit[x]
;

        }

        sumOfUnits =
sumOfCreditUnit + sumOfCarryOverUnits;
        if (sumOfCarryOverUnits >
maximumCreditLoad ) {

            System.out.println("you are on
probation see level adviser because sum
of carry over units is " +
sumOfCarryOverUnits+ " is greater than
the maximum credit load which is "+
maximumCreditLoad );
        }
        else if
(sumOfCarryOverUnits ==
maximumCreditLoad ) {

            System.out.println("you can
register only carry over courses ");
            for (int x = 0; x <
numberOfCarryOverCoursesOffered ; x++) {

                System.out.println(" * " +
CarryOvercourseNames [x] + " " +
```

```

CarryOvercourseCodes [x] + " " +
carryOverUnits[x]);
    }
    }
    else if (sumOfUnits ==
maximumCreditLoad) {
        System.out.println("
you can register all courses for " +
level);
        for (int x = 0; x <
numberOfCarryOverCoursesOffered ; x++) {

            System.out.println(" * " +
CarryOvercourseNames [x] + " " +
CarryOvercourseCodes [x] + " " +
carryOverUnits[x]);
        }
        for (int x = 0; x <
regularCoursesOffered ; x++) {

            System.out.println
(RegularcourseNames [x] + " " +
RegularcourseCodes [x] + " " +
RegularcreditUnit[x] );
        }
    }
    else {
        System.out.println("
you can register only regular courses
for "+ level);
        for (int x = 0; x <
regularCoursesOffered ; x++) {

            System.out.println
(RegularcourseNames [x] + " " +
RegularcourseCodes [x] + " " +
RegularcreditUnit [x] );
        }
    }
}
}
}

```

4767

Results

Table 1: Output for 100 level first semester registration with a maximum credit load of 21.

SERIAL NUMBER	COURSE NAME	COURSE CODE	CREDIT UNIT
1	Operating system	101	3
2	System analysis and design	102	3
3	Mathematical methods	103	3
4	Statistics for sciences and engineers	104	3
5	Use of english	105	3
6	Calculus	106	3
7	Vectors and mechanics	107	3

Table 2: Output for 100 level first semester registration with a maximum credit load of 21 with a message showing credit unit exceeded by 2.

SERIAL NUMBER	COURSE NAME	COURSE CODE	CREDIT UNIT
1	Numerical methods	101	3
2	Statistical inference	102	3
3	Engineering mechanics	103	3
4	Engineering drawing	104	2
5	Electronics	105	3
6	Electricity	106	3
7	Programming	107	3
8	Database management system	108	2

Table 3: Output for 200 level first semester registration with a maximum credit load of 24 with a message showing only registration of carry over students because is equal to maximum credit load.

SERIAL NUMBER	COURSE NAME	COURSE CODE	CREDIT UNIT
1	*Numerical analysis	211	3
2	*Mathematical methods	212	3
3	*Statistics	213	3
4	*Artificial intelligence	214	3
5	*Mathematical estimation	215	3
6	*Calculus	216	3
7	*Electronics	217	3
8	*Electrostatics	218	3

Table 4: Output for 200 level second semester registration with a maximum credit load of 24

SERIAL NUMBER	COURSE NAME	COURSE CODE	CREDIT UNIT
1	* Statistical inference	212	3
2	* Engineering mathematics	213	3
3	* Real analysis	214	3
4	* Discrete structure	215	3
5	Cloud computing	312	3
6	Object oriented programming	313	3
7	Structure programming	314	3

NOTE. – The output also contains a message showing that all courses, including carry over and regular courses can be registered because the total credit for carry over and regular courses is equal to the credit load.

Discussion

The research solved the problem of student course registration in tertiary institution by computing and monitoring accurately the corresponding credit units of courses registered for a semester with respect to the maximum credit load. In the first scenario of the program output in table 1 if the total credit unit for intended courses for registration is less than or equal to the maximum credit load for the semester the entire courses for the semester are registered conversely if the number of credit units for the intended courses for registration exceed the credit load as shown in table 2 where the credit load exceeds by 2 unit, a prompt message will be displayed informing the student that there is need to drop a 2 credit unit course. The prompt will give the student a guide in terms of the next line of action seeing the level advisor. Adewuyi (2011), who showed that on-line registration will generate stress among the fresh students in Lagos State University, pointed out that admission procedures into Nigerian Universities generate stress and anxiety among students because of advanced Information Computer Technology (ICT) where students have to register on-line. This e-registration is strange to some students who have passed out from secondary school before the introduction of Computer Education.

This buttressed the need for a pre-registration system that is a non-web-based system that will aid students to perform pre-registration of computing credit units before performing the on-line registration of courses in the portal. Uka and Ekwonwune (2019) deployed Object Oriented Analysis and Design Methodology (OOADM) using Hypertext Markup Language (HTML), Hypertext Preprocessor (PHP), Cascading Style sheet (CSS), Java Script (JS), and My Structural Query Language (MySQL) to design a university portal. The system allows students carry out course registration once the student have paid fees and other payments by issuing them a user name and password. The system only allow students to

perform registration only when there have paid fees, the limitation is that the student can only know the courses to register for after payment and also the system is a web based system while in our research we developed a simple credit unit calculator which can be easily used by the student without a pre-condition. Anameje and Okeke (2021) used Structured System Analysis and Design Methodology (SSADM) approach to analyze and design information systems. The process includes the application of a sequence of analysis, documentation and design concerned with analysis of the existing system, the logical data design, logical process design with PHP and MySQL tools to developed a web-based departmental Portal. The portal registers students, generates blacklist for student, generates quick result for each student, provides upload and download of course material as well as allocation of departmental courses.

In our calculator the system neither perform registration nor blacklisting of students in as illustrated in table 3, the number of carryover courses must be registered first fresh courses are registered. This will aid the student to easily identify the courses that require registration priority and also aid the level advisors. Padia and Santani (2021) developed an online school Management System. The system has a course registration page which is a home screen displaying a scrolling welcome message with the school name. It also has a small general message stating information about the courses offered at a college, with radio buttons for course search and course registration and a footer which has a hyperlink to the college. The next is the course search option which has a radio button. When the button is selected for course search, the card expands and prompts the student to select their upcoming semester. Another component of the system is the course registration option which registers the student. The system also has an error handling implemented using java and a database created and managed using MySQL.

The system developed by Padia and Santani is a registration system while in our research we developed a pre-registration system which will guide the student in identifying the courses for registration from the credit load of the selected semester. Zhang (1999) developed a University Course Registration and Management System (UCRMS) implemented using Microsoft Visual C++ and Microsoft (DCOM). The UCRMS is a real time, three-tier (presentation-tier, application-tier, and data-tier), distributed application capable of running on number of PCs according to the configuration. It provides an easy to use graphic user interface for both students and school managers. It allows students to make a self-registration for their selected courses. It also allows university administrators to manage the student and course data, such as adding a student, adding a course, dropping a student and removing a course. In our calculator after all the courses to be registered for a semester are entered the calculator first of all computes the credit units for carryover courses if the credit unit have not been exceeded as shown in table 4, fresh courses are then entered and the entire maximum credit unit for the semester is computed and displayed. So, Zhang (2019) developed a self registration system in our research we developed a credit unit monitoring system. Muthia and Efendy (2021) created a web-based application to assist in the process of registration for new students in Eden Everyday English Course Institute adopting the waterfall technique which has been found portend in the field of information system development.

Muthia and Efendy system was used to register new students in while our calculator can be used by both old and new student to compute their credit load before registration. Tumasius (2022) developed a portal that manage the records of client either by using manual keying or adapting the Optical Character Recognition Technology using a scanner as an input device to scan the forms and convert directly to text input in the record management system without inputting manually

and individually for easy data storage and update. Kaewsuwan and Khwunnak (2022) designed, developed and implemented a web-based portal application that enables different parties working with higher education general teaching to benefit from.

The application also determines the courses to registration first especially when issues of carry over courses arises. The laid down rule is that carry over courses are registered first. This is captured in the application which the output in table 3 clearly depicts. Noraziah *et al* (2011) Online Registration New Student Intake System (ORNSIS) used integrated barcode technology to overcome the manual system weakness. This system assists the administrator to access the database, to add new students' information, to log in into the system, new students' can access the system and also update their information. The administrator can get the actual number of students register online and also the barcode scanner is used to read the barcode at the offer letter.

Other conditions where the sum of carry over unit exceeds the maximum credit load is also captured. Here the student is either advised to registered all carry over courses or proceed for probation after seeing the course advisor. Also, if the student has no carry over the student is told to proceed with the registration. The program in this research is a useful tool for both students and level advisors.

Conclusion

The research shows for algorithm development, flowchart and code written using java programming language. The product of most researches anchored on the design and implementation of portal systems to automate registration in tertiary institutions. All the systems developed are web application developed using web tools. Most of the designs did not show the modularization procedures in the sub-systems development. This research solved the problem of the development of an application that is not web based which can be used effectively by the

students and course advisors to assist the students to mitigate the bottlenecks that come with not being able to register courses especially for students who have carry over courses.

Conflict of interest

There was no conflict of interest.

References

- Adewuyi, T. D. O. (2011). Stress and on-line registration among fresh students in Lagos State University. *Procedia Computer Science*, 3, 654-658. doi: <https://doi.org/10.1016/j.procs.2010.12.109>
- Anameje, C. & Okeke, O. C. (2021). Design and Implementation of a Departmental Portal (Case Study of Computer Science Department). *Global Scientific Journal*, 9, 4, ISSN 2320-9186. www.globalscientificjournal.com
- Chaka, P. & Mungadzi, F. S. (2013). An Implementation Of An Online Based Registration System In Tertiary Institutions In Zimbabwe. *International Journal of Engineering Research & Technology (IJERT)*, 2, 10. doi: <https://doi.org/10.17577/IJERTV2IS100988>
- Muthia, D. & Efendy, Y. (2021). Online Course Registration Application at Eden Everyday English Bogor. *Journal Teknoinfo*. 15. 74. doi: 10.33365/jti.v15i2.832.
- Nadir, Bamo & Ali, Aree. (2021). Online Course Registration and Advisory Systems Based on Students' Personal and Social Constraints. *Kurdistan Journal of Applied Research*, 6. 11. doi: 10.24017/science.2021.2.8.
- Noraziah, A., Johari, M. S., Herawan, T., Sidek, R. M., Lee, H. C., & Abdalla, A. N. (2011). Managing Registration of New Student Intake using ORNSIS-Integrated Barcode Technology. *Procedia - Social and Behavioral Sciences*, 28, 144-150. doi:

<https://doi.org/10.1016/j.sbspro.2011.11.029>.

Padia, A., Santani, V. (2021). Course Registration System. *International Research Journal of Engineering and Technology (IRJET)* e-ISSN: 2395-0056 Volume: 08 Issue: 04 | Apr 2021 www.irjet.net p-ISSN: 2395-0072.

Tumasis, J. (2022). Web based management information system with optical character recognition technology for a Philippine accounting firm. *South Asian Journal of Engineering and Technology*, 12. 48-54. doi: 10.26524/sajet.2022.12.8.

Uka, K., K., Ekwonwune, E., N. (2019). Web Based Students' Record Management System for Tertiary Institutions. *International Journal of Advanced Research in Science, Engineering and Technology*, 6, 6. ISSN: 2350-0328.

Zhang, Qiang (1999) University course registration and management system: a distributed application using Microsoft distributed component object model. *Masters thesis, Concordia University*.