iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** IX **Month of publication:** September 2022

DOI: https://doi.org/10.22214/ijraset.2022.46760

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue IX Sep 2022- Available at www.ijraset.com

Design and Implementation of an Online Enrolment System for Higher Education Institution amidst covid-19 Pandemic in Zamboanga Peninsula, Philippines

Zenon A. Matos Jr[1]
School of Engineering and Technology, J.H. Cerelles State College

Abstract: ***The purpose of the study is to design and implement an enrolment system through web based application intended for higher education institution in Zamboanga Peninsula amidst covid-19 pandemic. The functionalities of the system are guided using Use Cases identified during requirement phase. The existing system encountered several constraints on the process of the enrolment, especially in detecting conflict of course schedules and the availability of slots of the courses offered, handling large number of data, and in cases where modifications or errors in the program that need to be fixed. The methodology used was prototyping to take advantage of the limited experience of users in using computerized systems. It was implemented using Hypertext Pre-processor (PHP) programming language and MYSQL database along with JavaScript, Cascading Style Sheets (CSS), JQuery and Macromedia Dreamweaver as integrated development environment (IDE). The functionalities of the system get the approval of the school administrators allowing the program to be use in an actual enrolment setting.***

Keywords: **Online Enrolment System, Implementation of Enrolment System, Web-based System, Higher Education Institution**

I. INTRODUCTION

Higher Education institutions are looking for methods and techniques to speed up transactions and improve operations. One of these processes is registration. Enrolment systems play an important role in all educational institutions as they manage information about students, track courses, and create enrolment lists. It provides a means of storing information and quickly looking up records by storing files in something called a database.

Web-based applications are now widely used due to their ubiquity. Given its current and potential growth, the Internet has become a way for e-commerce to improve and implement processes given existing technology. Some colleges and universities have implemented online registration support, allowing students to register for courses over the internet without having to go to the school campus. Inspired by these online services, the design and implementation of an admissions system for higher education institutions in Zamboanga Peninsula is a practical step during the pandemic. This study looks at common school procedures and how these procedures can be transformed into a better web-based application system.

The Southern City College is a private non-sectarian institution of higher learning that envisions the advancement of an ideal multi-cultural Filipino society in the Philippines, with a vision of “Triumph in tenacity”.

The existing enrolment of the school for all year levels takes two days. Based on the observation of the previous enrolment, the process may extend from one to three days as the students were not able to enlist on the specified date remain un-enlisted. Enlisted courses to be enrolled for the students will be deleted if the students do not pay the initial down payments within the given period of the three (3) days.

II. OBJECTIVE OF THE STUDY

The main objective of the study is to design and implement a web-based enrolment system for Higher Education Institution amidst Covid-19 pandemic. The specific objectives are the following:

1) To model the enrolment system using UML.
2) To design a normalized database that minimizes data redundancy.

©IJRASET: All Rights are Reserved

3) To develop a module that detects if the class size reaches its limit, determines if the schedules of the students are in conflict, assesses the student's fees and accounts, records the payment of the students and provides necessary printed reports needed with regards to the enrolment process.
4) To test the functionality of the system guided by Use Cases
5) To deploy the developed system

III. METHODOLOGY

Prototyping was a methodology used in system development. This allows service providers to understand customer requirements in the early stages of development. It helps you get valuable feedback from your customers and helps software designers and developers understand what to expect from the product they are developing (Sommerville, 1997). One of the key principles of the prototyping model is user involvement throughout the development process. This makes the final implementation more likely to be accepted by users (Maheshwari and Jain, 2012). Prototyping was chosen as the method of developing the system because the user is involved in the early stages of development. Maheshwari and Jain (2012) found that users know the problem domain better than anyone on the development team, so prototyping allows users to view/manipulate prototypes to provide better and more complete feedback and specifications. says it can. More interaction means better product quality.

The researchers surveyed the current system and conducted a series of interviews, starting with admissions officers, department heads, cash registers and registry offices. After collecting all registration information, researchers organized the collected data to define the scope of the project.

A. Requirement Documentation

The goal of the requirement identification is to identify the requirements of the system. This is to arrive at the agreement of the stakeholder and to know the objectives of the project. The researcher interviews the different designated positions about the current enrolment system of Southern City Colleges to establish the business cases for the system and delimit the project scope.

The researcher used the following diagrams:

1) Use Case Diagrams helps to model the system functions and the actors involved in the enrolment system as illustrated in Figure 4.1.
2) Activity Diagrams was used to identify the sequence of actions from the initial state to the final state. A simple diagram was then used to establish the baseline architecture of the system which provide a stable basis for the designed in developing the initial prototype.

B. Developing the Initial Prototype and Modelling the Enrolment System

After identifying the requirements, the researchers developed a first prototype of a registration system in which the requirements were presented and a user interface provided. The intended prototype is a web-based application implementation of a registration system. Researchers used UML modelling tools to model the registry system while defining software requirements. These modelling tools are Use Cases and activity diagrams. A use case diagram was used to identify the actors involved and the main functions associated with the actors. Actor identification was based on the existing roles of those involved in the old registration system. These roles also formed the basis for use case scenarios. Activity diagrams were used to guide the user through the operation of the system modules and show the user the sequence of actions from the initial state to the final state. It was also used to guide the user in performing the necessary actions in each case, especially when validating prototypes.

C. Process of Normalization

The process of normalization was followed in coming with a normalize database form. First, data from the existing database were extracted and plotted. Each table from the existing database was evaluated for redundancy. The first normal form was used to check if the data which concerns with identification of repeating group of information. Second and third normal forms were also applied for the succeeding evaluation of each table. The objective was to minimize data redundancy in the storage of data. For instance, some attributes of the tables in the old database design were removed in order to avoid data duplication in storing the enrolment information. For example, studentname and subjecttitle appear three and two respectively in the old database design. Along with the normalization process, the entity relationship diagram has been used to provide the quick view of entities in a database so it will be well understood in designing the new database.

This was also helpful in illustrating the overall relationship of each identified entity or table. The Entity Relationship Diagrams (ERD) is shown in Figure 4.3.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue IX Sep 2022- Available at www.ijraset.com

D. Implementation

The Hypertext Preprocessor (PHP) was used as a programming language for the development of dynamic web content and MYSQL as database storage.

The researcher decided to use MYSQL database because of its open source software that can store a large bulk of data compare to MS Access.

The Macromedia Dreamweaver was the selected integrated development environment (IDE) for the development of the initial prototype and user's interfaces. Furthermore, the Cascading style sheets (CSS) was also used for the webpage design, JavaScript and JQuery to make it more interactive. The following section will discuss important modules develop as part of major functionality support of the enrolment system.

1) Process In Avoiding Conflict Of Schedules

The scheduling scheme of the school was divided into two sessions per semester. This is to reduce the number of loads taken by the students simultaneously to avoid failing or incomplete grades.

The considerations in detecting the conflict of schedules is to check if the time is equal or between the existing time schedule in a given room and day within a session per semester as shown in the implementation code of detecting the conflict of schedules in Figure 4.5.

2) *Monitoring The Slots Availability Of A Subject*

Monitoring was done by checking the allowable size specified in the courses. This is done every student that will enroll in a subject. If an attempt to add a student in a subject that reaches the limit of allowable size, a message will be displayed informing the user of the allowable number of students for that particular section.

Moreover, the color of the design will change it into color yellow if the remaining slots are almost full and if the schedule are full it will disappear in the choices of the courses.

3) Assessing Student's Fees And Account

The assessment module is a functionality that assesses the tuition fees and other miscellaneous fees. For tuition fee, the assessment was based on the number of units taken multiplied by the amount per unit. The remaining fees were retrieved from the specified assessment entries in the database along with other fees.

The total assessment per student is the total of miscellaneous and other fees and tuition fee. This assessment was recorded as part of student record in the tbalance (total balance) attribute of the database. In cases where the students availed of discount (percentage), a percentage of the discount will be deducted from the total assessment depending on the type of discount or privileges. This will then be recorded as total assessment.

4) Recording Of Payment

The payment module is a feature that will record the payment of the student and other fees. The student's may choose the mode of payment in their tuition fee to pay partially or in full payment.

If the student's choose to pay their tuition fee partially, the total account balance was divided into four (4) exams in every semester and that will be the amount to be paid by the student's before they take the exam.

The payment was recorded and deducted in their account balance as part of the student's record in the "tamount" (total amount paid) attribute of the database.

Aside from the student's tuition fees, it will also record other payments like assessment card replacement, authentication fee, certification fee, diploma, application exam, laboratory fee, TESDA assessment fee and transcript of records.

IV. RESULTS AND DISCUSSIONS

A. Software Modelling Tools

Through observation, collection of documents used during enrolment, and interview with the different stakeholders, the functionality of the software has been identified. The Use Case diagram was used to document this process showing major stakeholders or users.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue IX Sep 2022- Available at www.ijraset.com

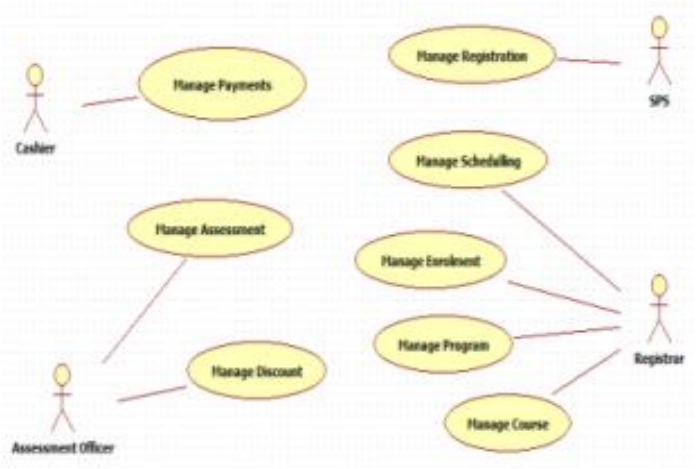


Fig. 1: Use Case Diagram

As illustrated in Figure 1, the diagram shows the major actors operating with the enrolment system. These are the Cashier, Registrar, Assessment Officer, and Student Personnel Services (SPS). As can be seen, the Registrar has the most number of functions with four major functions, followed by the Assessment Officer with two, and by Cashier and SPS with one functionality. After having identified the major functionalities, Use Case Scenarios were used to explicate the specific activities to be done in relation to the actor. An example of Use Case Scenario for Managing Scheduling is shown in Figure 1. Two major scenarios are used to illustrate each of the functionality.

The first is the main success scenario followed by the scenario extensions. The former details the normal flow of the events when the actor perform a particular function, while the latter is concerned with scenarios which do not conform to the normal flow of events. After defining the functionality and processes, data requirements are then defined. Entity Relationship Diagram (ERD) was used to define entities and the corresponding attributes representing the data requirements of the system. Given that there is an existing system, the starting point to determine the specific attributes are the existing ERD. Figure 2 shows the existing entity relations of the database at Southern City Colleges enrolment system.

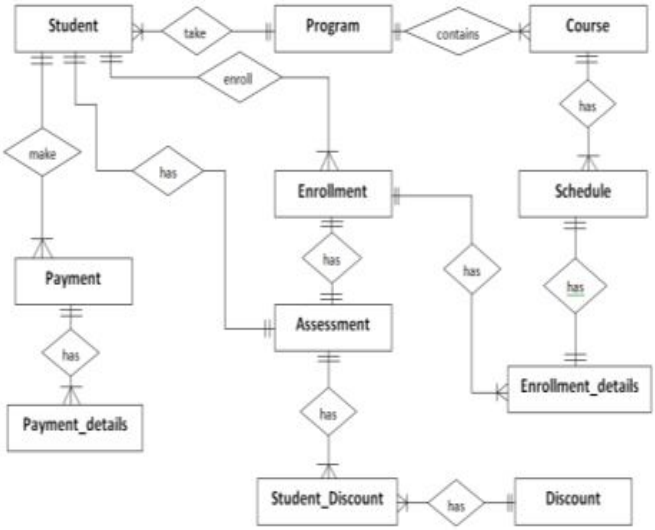


Fig. 2: Entity Relationship Diagram

The entities such as Student, Enrolment, and Payment contained the “studentname” attribute and the entities such as Course and Course Schedule contained the “subjecttitle” attribute that leads to data duplication in storing and retrieving the enrolment information. Furthermore, the use of Microsoft Access as a database application is not capable of handling large number of data which resulted to system lag. Given that the existing design of enrolment system has data redundancy, there is a need to revise it. Figure 2 shows the relationships of each entity in the new database design of enrolment system.

©IJRASET: All Rights are Reserved

This is to reduce the duplication of data encountered in the implementation of new database design of the enrolment system. Figure 3 shows the entity relations of the new database design. It is then implemented using MYSQL database which stores large number of data compared to the existing MS Access application. Further, some of the attributes in the new database design has been removed to eliminate data duplication.

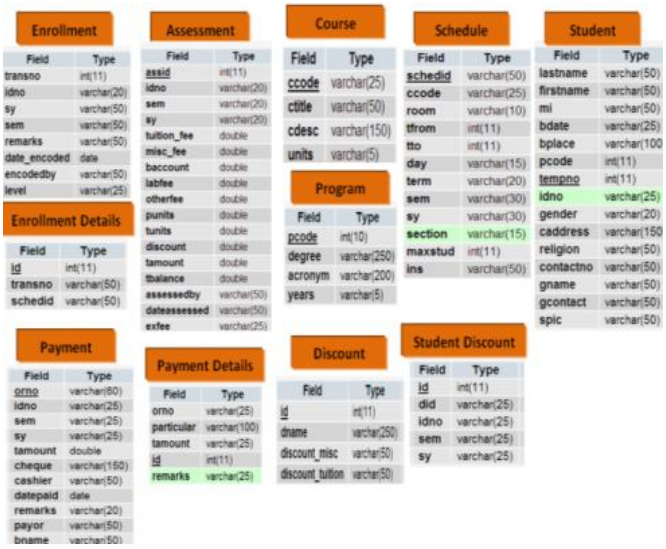


Fig. 3: Entity Relation with corresponding attributes

B. Development Of The Prototype

The researcher started developing the initial prototype after the enrolment system requirements has been identified. An online enrolment system has been chosen to be implemented in order to address the problem in cases where modifications or errors in the enrolment system need to be fixed, the system administrator will not go to each workstation to update the system, and to cater students' needs without going to school physically.

With a web-based application, the systems are accessible anytime, anywhere, via PC, tablet or mobile phones putting the user in charge of where and when they access the application. The installation and maintenance becomes less complicated. Once a new version or upgrade is installed on the host server and all users can access it straight away and there is no need to upgrade each client PC. The researcher used PHP programming language to implement a dynamic web content of the enrolment system and MYSQL database application capable of handling a large number of data. In order to create interactive effects of the web pages, the JavaScript, Cascading Style Sheets (CSS) and JQuery has been utilized. The development of the system functionality was based on the Use Cases identified as illustrated the Use Case Diagram. The implementation of the use cases started from: (1) Managing the student registration by the SPS Director; (2) Managing the program offered of the school; (3) Managing the program Courses; (4) Managing the Course Schedules offered in the current semester of the school year; (5) Managing the Student enrolment; (6) Managing student discounts; (7) Managing the student assessment of fees and account; (8) Managing the student payments.

C. Scheme In Avoiding Conflict Schedules

The conflict of course schedules happens when two or more courses have the same time, room and day within a term, semester and school year. In the enrolment system process, one of the dilemmas is the conflict of course scheduling and the courses schedule for the students. The current enrolment system cannot detect the conflict of course schedules and cannot monitor the availability of slot if it exceeds the maximum class size in current semester and school year. Additionally, the conflict of courses schedule enrolled by the students in current semester and school year was not determined.

The new system has the ability to determine the conflict of course schedules encoded by the dean/department heads. The considerations in determining the conflict of schedules are: (1) the system check the time, if it is equal or in between the existing time schedule; (2) the system check the room/day, if it is equal to the existing values (room/day) after it satisfied the number one consideration; (3) the system check the term, semester and school year if it is equal to the existing values after it satisfied the number 1 and 2 considerations; (4) if the rules 1, 2 and 3 considerations are satisfied a message will be displayed "Course schedule created is in conflict". Figure 4 below is the implementation code in detecting the conflict of course schedules.

```
$dconflict=mysql_query("Select * from college_schedule where (room='".strtoupper($_POST['room'])."' and
day='".strtoupper($_POST['day'])."') and ((".$_POST['sfrom']." > tfrom and '".$_POST['sfrom']."' < tto) or ('".
$_POST['sto']." > tfrom and '".$_POST['sto']." < tto) or ('".$_POST['sto']."=tto and '".$_POST['sfrom']."=tfrom))
and (term='".$_POST['term']."' and (sem='".$_POST['sem']."' and sy='".$_POST['sy']."'))");

if (mysql_num_rows($dconflict)>=1){
$cf=mysql_fetch_array($dconflict);
$m=$cf['ccode'];
    echo "<script language=javascript>
    alert('Cannot Proceed! Course in Conflict with $m');
    location.href='javascript:window.history.go(-1)';
    </script>";
    exit;
}
```

Fig. 4: Implementation code in avoiding conflict schedules

D. System Architecture and Actual Set-up

After the initial prototype has been developed, the developed system is ready to be implemented as first iteration of deployment. The purpose of this initial prototype is two folds. First, it will serve as validation of the user requirements as specified in the requirements analysis; and to evaluate the functionalities on the user side. In order to make this purpose a reality, an actual set-up has been formulated. The system architecture composed of web server and system's database. It shows the flow on how the clients/users access the system through the central server. Figure 5 shows the system architecture used of Southern City Colleges enrolment system. The system used Apache web server that handles the request of the users/clients when they open a web pages using the Hypertext Transfer Protocol (HTTP). The apache supports MYSQL database that manages the storing and retrieving of information by the users/clients in every transaction they made. It also supports PHP scripting language for the dynamic web development of the system. The JavaScript, Cascading Style sheets (CSS) and JQuery used to create interactive effects within web servers.

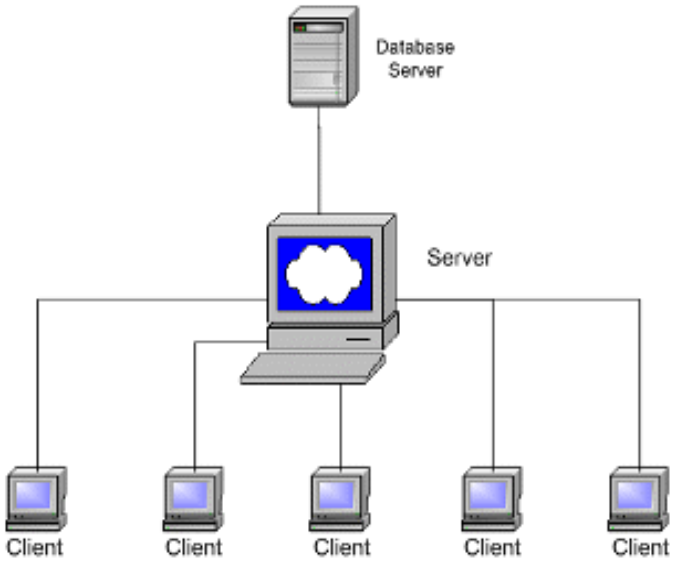


Fig. 5: System Architecture

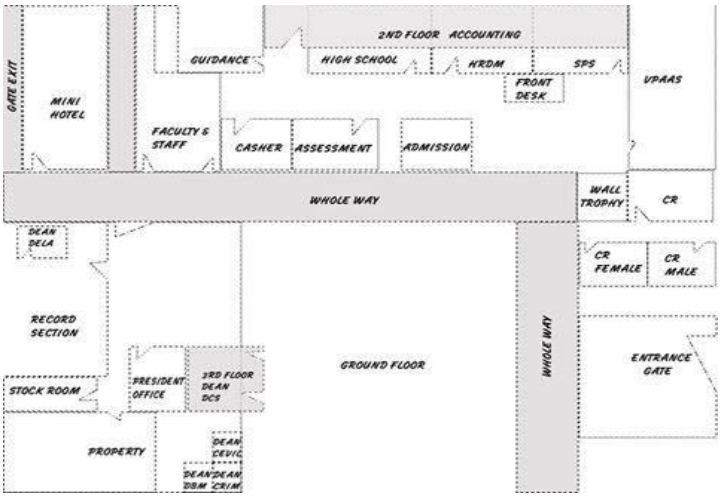


Fig. 6: Actual Setup

Figure 6 shows actual setup of the enrolment system in Southern City Colleges. The central server is located at the Department of Computing Studies, 3rd floor FMC IT building, Southern City Colleges, Pilar Street Zamboanga City. The estimated number of clients connected to the central server is thirteen (13) starting from the school cashier, registrar, deans of the different departments, Assessment, and the Student Personnel Services (SPS).

E. Assessment Of The System

The researcher developed the initial prototype of the enrolment system and was presented to the users involved in the project. The user's reviewed the prototype and gave feedback for further enhancement of the developed prototype. There are seven (7) designated positions who reviewed the prototype; these includes the Deans and Department Heads, Assessment officer, Cashier, Student Personnel Services Director (SPS), Accounting Officer and School Registrar as it shown in Figure 1.

First Iteration, the users tested the functionality of the developed prototype guided by Use Cases. The testing transitioned of system functionality for the enrolment system during first iteration is shown in Table 1.

Table 1: First Iteration – Use Case Testing Transitioned

Use Case	Description of Testing	Remarks
Manage Payment	Test wild characters.	OK
	Add Payments.	(Not yet OK) Additional attributes (cheque # and bank name), add section for other payments.
	Cancel Payments.	(Not yet OK) Has error
	Print official receipts (OR)	OK
	View Payments.	OK
Manage Assessment	Test wild characters.	OK
	View list of students (Assess/Not Assess)	OK
	Assess student fees and accounts.	(Not yet OK) Additional attributes (excess payment and other payment).
	Add discounts and privileges.	(Not yet OK) Has error
Manage	Test wild characters.	OK

Scheduling	Add / detect the conflict of course schedules.	(Not yet OK) Error in detecting conflict of schedules.
	Assign class size	OK
	View list of schedules (open/closed).	(Not yet OK) List of closed schedules not working.
Manage Enrolment	Test wild characters.	OK
	Detect student conflict schedule.	(Not yet OK) Error in detecting student conflict schedules.
	Print certificate of registration (COR).	OK
	Monitor schedule class size.	OK
Manage Program	Test wild characters.	OK
	Add, Update, Delete.	OK
	View list of programs.	OK
Manage Course	Test wild characters.	OK
	Add, Update, Delete.	OK
	View list of courses.	OK
Manage Discount	Test wild characters.	OK
	Add, Update, Delete.	OK
	View list of discounts.	OK
Manage Registration	Test wild characters.	OK
	Add, Update, Delete	OK
	View list of students.	OK
Enrolment Statistical Report	Report by Semester and School Year	(Not yet Created)
	Report by Program and Year Level	(Not yet Created)
	Report by Department	(Not yet Created)

The user's reviewed the prototype and specified some comments and recommendations like: button positioning, adding enrolment statistical reports, additional fields for excess payment of the student in the assessment module, filtering the students in the payment collections module should automatically appear once they type the keyword on the search area without clicking the search button, and system bugs. Second iterations, the researcher again presented the prototype to the users involved in the project in which their comments and recommendations in the first iteration was taken care of. The testing transition of the prototype is shown in Table 2.

Table 2: Second Iteration – Use Case Testing Transitioned

Use Case	Description of Testing	Remarks
Manage Payments	Add payments.	OK
	Cancel payments.	OK
Manage Assessment	Assess student fees and accounts.	OK
	Add discounts and privileges.	OK
Manage Scheduling	Add / detect the conflict of course schedules.	OK
	View list of schedules (Open / Closed).	OK
Manage Enrolment	Detect student conflict schedules.	OK

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Enrolment Statistical Reports	Report by Semester and School Year.	OK
	Report by Program and Year Level.	OK
	Report by Department.	OK

F. User Observation during the implementation

This section provides the observation and insights of the users during the enrolment system implementation. According to the school registrar, the new implemented enrolment system is better compared to the old system especially in determining the conflict of schedules, monitoring the slots availability of the courses offered and student course enlistment compared to the old enrolment system. This is because of the functionality supported by the implemented system. On the other hand, the assessment officer commented that the new system helps them a lot especially in assessing the student fees and accounts, managing the student discounts and privileges compared to the existing enrolment system. It lessens the number of students who fall in-line in the assessment area when they start to use the new system.

On the cashier level, the recording of the payment is better compared to the old system and she feels comfortable in filling the payment form. They can immediately generate the financial status of the school and determine of how many students who pay their tuition fees. According to student personnel and services (SPS) director, the system helps them in managing the information of the students especially in retrieving student's records compared to the old enrolment system implemented.

The new system lessens the work of the system administrator in managing the system and installing system updates. This is because the system administrator will not go to each workstation; instead modification to a server will be reflected to each workstation. Additionally, they also said that the new system was better in terms of accessibility because they can now access the system through their Personal Computer (PC), Laptop and Tablet as long as they are connected to the system server. Compared to the existing enrolment system, they can only access the system in specific workstations where the system administrator installed the system.

V. CONCLUSIONS AND RECOMMENDATIONS

The enrolment system was implemented at Southern City Colleges, Inc. one of the higher education institution in Zamboanga Peninsula amidst covid-19 pandemic based on the functional requirements defined during data gathering. It was implemented in a web-based application which allowing the system administrator to maintain the systems server to reflect new updates or changes. The data normalization has been observed in the new database design to reduce the data redundancy in storing and retrieving of enrolment and student information. The following features are: Monitor the courses offered and the slots availability for the students; determine if the courses schedules of the students are in conflict; assist the Assessment Officer to assess the student's fees and monitor their accounts; record the payments of the students and provide necessary printed reports needed with regards to the enrolment process; and show the graphical presentation about the number of students enrolled using bar chart. The new implemented enrolment system helps the administrative and staff of the school to manage the enrolment processes. The prototyping methodology also supports possible enhancement in case there are changes needed based on the suggestion of the users. Lastly, it specifically addresses the problems on conflict of student's schedule as well as availability of slots especially during peak times of enrolment. Also, it resolved the data redundancy problem of the previous database by applying normalization. Moreover, modification of errors or any changes in the code makes update easier since only the server is being updated which automatically updates the entire system giving.

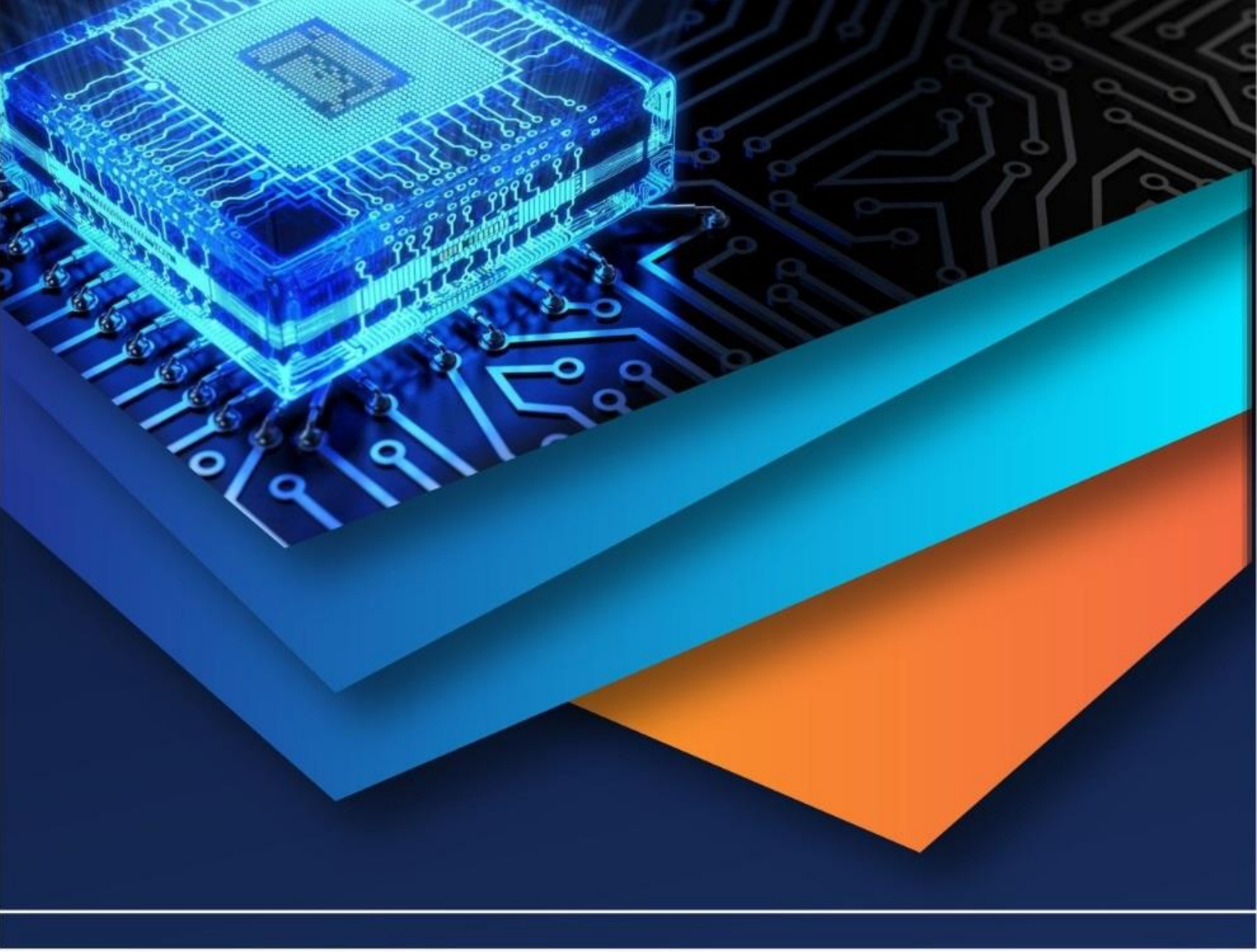
It is recommended that the adoption of the developed enrolment system through web-based application be done in a progressive manner to improve the enrolment processes. This progression coincides with the prototyping methodology. In this manner, sources of data to formalize the process need not to elaborate the entire process. That means that the initial system will be developed and further elaborated in the succeeding evaluation from the users. This approach addresses the limited idea of the user during initial data gathering especially for those who have limited experience in using an electronic information system to perform their function in the organization.

For those who want to study more about the enrolment system through web-based application, the researcher recommends to examine the integration of the following to further improve the enrolment system and its functionalities: (1) Online Payment through credit, debit card or even pay pal so that students may not go to school to pay their tuition fees and other fees; (2) Student Advising that automatically advised the student of what are the courses to be taken or not; (3) High School enrolment system of Southern City Colleges, Inc. should also be considered to improve the process of the enrolment in the school

REFERENCES

[1] Booch, G., Jacobsonm, I. & Rumbaugh, J. (2000). OMG Unified Model Language Specification. Boston, MA 02116
[2] Coughlan, P., Suri, J., & Canales, K. (2007). Prototypes as (Design) Tools for Behavioral and Organizational Change: A design-based approach to help organizations change work behaviors. The Journal of Applied Behavioral Science, 43, 1-13. doi: 10.1177/0021886306297722
[3] Caipang, C. (2013). Development of a computerized enrolment system in a rural-based higher education system. Academic Research International, 4(3), 142-147. Retrieved from www.savap.org.pk/journals/ARInt.
[4] Jones, C. (2011). Robert Gordon University. Online enrolment v2 and Agile Development, 2,1- 17. Retrieved from https:// www.ucisa. ac.uk/~/media /Files/members/awards/excellence/2011/Robert%20Gordon.ashx
[5] Lafon,M.&MackayW.(2003).Prototyping Tools and Techniques: Prototype Development and Tools,2, 1-37. Retrieved from https://www.kth.se /social l/upload /52ef5ee4f2765445a466a28a/
[6] Martin, R. C. (2003). UML for Java Programmers. USA: Prentice Hall Professional Technical Reference, 1, 2-27. Retrieved from https://www.csd.uoc.gr/~hy252/references/
[7] Maheshwari, S. & Jain, D. (2012). A Comparative Analysis of Different types of Models in Software Development Life Cycle. International Journal of Advanced Research in Computer Science and Software Engineering, 2(5), 285-289. Retrieved from www.ijarcsse.com
[8] O'Mahony, M.& Smyth, B. (2007). A Recommender system for online course enrolment: An initial study, 1, 33- 36. Retrieved from https: //www.macs. Hw .ac.uk/~dwcorne/ACMRecSys07/
[9] Sommerville, I. (1997). 'Software Process Models'. In Handbook of Computer Science and Engineering. 4(2). A. Tucker, CRC Press.
[10] University of the Philippines - Diliman Website. Retrieved August 21, 2022 at http://upd.edu.ph/
[11] De La Salle University –Manila Website. Retrieved August 21, 2022 at http://dlsu.edu.ph
[12] MasseyUniversityWebsite.Retrieved August 21, 2022 http://enrol.massey.ac.nz/massey/enrolment/webenrol/guide.cfm#whatdo
[13] University of the Philippines Website. Retrieved August 21, 2022 at http://www.upd.edu.ph
[14] University of Waterloo Website. Retrieved August 21, 2022 at www.uwaterloo.ca
[15] University of Sta. Isabel – Naga Website. Retrieved July 20, 2022 at http://www.usi.edu.ph

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)