UML model of a Grade Point Average Calculator Software

Nuvadga Christian Tete
University of Buea
mpara.faith@gmail.com

Henry Ngatchu
University of Buea
henry.ngatchu@ubuea.cm

Ngatchu Damen
University of Buea
ngatchu_nn@yahoo.com

ICT4AFRICA 2014 Conference Papers
Paper No. xx, pp. xxx-xxx

Conference Paper

(Will be set by ICT4AFRICA Committee)

UML model of a Grade Point Average Calculator Software

(*Received xxx 2011, accepted xxx 2011, will be set by the editor*)

Abstract.*As enrollment of students into tertiary education in Cameroon state universities continues to increase, the need for an efficient and speedy method ofmeasuring academic performance of students has become more significant. Thispaper attempts to discuss at length a UML model for a Grade Point Averag(GPA) calculator software. UML (Unified Modeling Language) is a language for visualizing, specifying, constructing and documenting artifacts of software intensive systems. The UML diagrams used in this paper helps us in the analysis,design and simulation of possible scenarios, and to implement the results. TheUML diagrams were designed with Visual paradigm Community Edition. Theharmonization of two standards of computing student's GPA in Cameroon's state universities has also been discussed.*

Keywords: Unified Modeling Language, Grade Point Average, Object-Oriented, Cameroon State Universities, Harmonize GPA Computation.

INTRODUCTION

In recent years, there has been a paradigm shift in the manner in which higher education is delivered and supported. Thanks to innovations in the areas of Information and Communication Technology that has helped institutions to deliver higher education in a more effective manner while transacting its processes even more economically. Computation of students GPA is of paramount importance couple with the fact that it serves as a yard stick for measuring the academic performance of students in tertiary education all over the world. This phenomenon of grading student's work quantitatively was developed by William Farish and first introduced in the academic milieu by Cambridge University in 1792 [What is CGPA, 2011 http://www.goggle.com/m?q]. Since then, several changes have been made on this concept.

Sampling popular opinions, it is revealed that an approximately a large number of students do not even understand the concept of GPA computation.

Majority of students go through university studies up to the point of graduation without ever having the opportunity to track their GPA progress. The presentation in this paper provides a step towards eliminating this myth that exists amongst students in the university community in Cameroon. A GPA calculator will become handier for students in higher institutions. Anaesthetically well design GPA calculator will save time and energy that students spend trying to figure out how various grade scenarios will influence their final GPA. The GPA calculator will serve as a tool for students who would want to see how their grades are coming along as the semester progresses. The GPA calculator is important for the fact that it turns the student's credit hour and letter grade to a full flesh GPA. The flare a GPA calculator brings is answering thequestion "if I can only have 'C' grade in course 'X' would an 'A' grade in course 'Y' make a difference in my final GPA?"

Though, there are a good number of electronic devices developed to serve the purpose of evaluating student performance by grading their work, most of them do not provide a unified standard for the concept of GPA computation.This paper presents a designed of a UML model for GPA calculator software that attempts to standardize the concept of GPA computation in Cameroon state universities, providing a deep overview of the design and implementation techniques. UML, being the de-facto industry standard for modeling object-oriented (OO) softwares (Muhammad Usman et al. 2009) [6] makes it possible with thediagrams it provides to model structural and behavioral aspects of an object oriented system (software applications). UML diagrams possess sufficient capabilities to handle almost all of the implementation details. A UML model for a GPA calculator software comes handy since programming languages are not at a high enough level of abstraction to facilitate discussions about design. UML as a blue print is about completeness complete specifications of the system. Well-developed UML diagrams for a GPA calculator software will be compiled directly into executable code, and UML becomes the source code.

The purpose of this paper is to present an object oriented model for GPA calculator software that attempts to harmonize the overall concept of GPA computation in the academia (Cameroon State Universities). The general objectives of this paper are to:

1) Present a UML design that will serve as a backbone for harmonizing the two different GPA computation standards that are used in Cameroon state universities.

2) Outline the UML diagrams for a GPA calculator software application development processes.

The paper is limited to addressing issues patterning to the two above mentioned objectives. Considering the case of Cameroon state universities standards, University of Buea (UB) which is the lone state Anglo- Saxon University in Cameroon at the time of this publication and University of Bamenda (UNIBA) follows a grading standard that uses letter (A, B+, B, C+, C, D+, D, F, and I) whereas the rest of the state universities (Universities of Yaounde I, II, Dchang, Douala and Ngaoundere) follows a score/20 standard. However many school of thoughts believe that the two standards mentioned above does not provide a balance when tracking and comparing performance (in this case quantitatively) of students from these different state universities therefore beckoning for a review of a GPA calculator model that answers to this problem which is what this paper attempts to provide.

The rest of this paper is organized as follows. Section 2 presents a background on software systems or applications relating to this paper. In section 3, we substantially describe the system's architecture. This section demonstrates the design efforts including structural design and behavioral design. Section 4 provides discussions on the design approach, and model constructed. Finally, Section 5 concludes the paper

LITERATURE REVIEW

A great deal of work is being carried out as far as developing software systems and applications using UML as a tool kit is concern. The key issue reviewed in this paper is presenting a UML model for a Grade Point Average (GPA) calculator that attempts to standardize the concept of GPA computation in Cameroon state universities as opposed to the following related works which mostly deals with aspects such as using UML as a standard tool to model, measure performance, and estimate the volume of systems. The papers which I have reviewed in this section looks at how UML is use to model, evaluate and estimate a cross section of systems including multi-threaded processes on dualcore processor, database systems and computer applications, decisional processes, JAVA code generation, and web applications. Also works relevant to computation of grade point average has also been reviewed in this section.

Software systems or applications modeled with UML are easy to understand and implement. Com+HDM being UML-based approach, Azrul Hazri Jantan (2012) [8] in his paper in which he implemented SLEX-Web as case study found that no parts in the design processes seem to be difficult and hard to model the end result of each design process was not too complex and easy to understand. He wrapped up his paper by saying that models constructed from design processes (conceptual, navigation, and user interface) makes it easy web pages layout – and yet they are all expressive and intuitive to implement.

Talking of UML as standard tool for modeling object oriented software systems and processes, Dr. Vipin Saxena et al. (2009) [1] publish a study inwhich UML was the main tool used in modeling the system to be tested. The findings from this study was tried and approved (The objective of their paperwas to evaluate the performance of multithreaded programs developed in C#and Java on Intel's Dual Core processors. The performance comparison in term of execution time was reported. It was concluded that C# takes less execution time as compared to Java over similar processor architectures. It was therefore concluded that the performance of C# is better than JAVA for multi-threaded programs and therefore, recommended for large number of threads computations.). Credit was attributed to the modeling technique (i.e. UML) used to model the system. From their study one can point that UML is special tool for modeling system processes.

Outlining the use of Unified Modeling Language (UML) as a standard notation of realworld objects in developing object oriented design methodology for computer applications and database systems, Sunguk Lee (2012) [2]discussed UML as a tool for specifying software systems that include standardized diagrams to define, illustrate and visually map or model a software system's design and structure, laying emphasis on the following UML diagrams: the use case diagram, class diagram, sequence diagram, state chart diagram, activity diagram, component diagram, and deployment diagram. These diagrams are used to provide models for several different systems and processes.

Theoretically, there are different models for the same system. UML supports object oriented representation of the entities defined in organization and allows the design of a model that can be analyzed by theoretical approach.Udrica Mioara, et al (2012) [3] extrapolated the above mentioned conclusions to the real system and it did paid the dividend of improving on the activities of the organization. The main objective of their paper was to present how UML diagrams can be used to improve the decisional process in which they revealed that decision makers or specialists in financial analysis were the last mediators,who, depending on the context, accept or change parts of model or even the entire model. Here we notice that the different UML diagrams in conjunction with UML elements and relationships combine to paint the big picture.

Of eight the UML diagrams used in the paper ―Implementation of Function Point Analysis in Measuring the Volume Estimation of Software System in Object Oriented and Structural Model of Academic System‖ Dian Pratiwi (2013) [4] depicted the behavioral and structural models with only two of them; Use case diagram, and the class diagram respectively. Meanwhile six of the remaining eight diagrams were considered complementary. UML can also be used in modeling database systems.

In their work presentation, Dr. Vipin Saxena et al (2011) [5] pointed out that UML can be easily applied in the area of modeling database systems with attempts made to represent the Life Insurance Corporation (LIC) of India database through UML. Database systems modeling with UML could also be extended to model database systems for hand held devices like mobile system,PDA, etc. Aesthetically well engineered UML designs depending on the UML tool used in modeling (e.g. Visual Paradigm) can be compiled directly into executable source code in any language that supports object oriented programming.

Automated Java code generation tool known as UJECTOR implementation and presented by Muhammad Usman et al. (2009) [6], takesUML class, sequence, and activity diagrams as an input to generate completely executable Java code. After an empirically evaluate the generated code from UJECTOR with different existing UML case tools. The results showed that the generated Java code was consistent with UML diagrams. The generated code was fully functional and understandable. Good software development depends also on how good the source is. Designing a UML model for a GPA calculator will better place us realizing a better end product which will go a long way to serve student track their grades.

G. N. Ezeh, et al (2012) [7] in their work highlighted the importance of concept computation of Cumulative Grade Point Average as a way of measuring the academic performance of students in tertiary education. They also pointed out the fact that students of higher institutions cannot continue to ignore the need for a portable, reliable, low cost and faster means of Computing Cumulative Grade Point Average of their students.

Their study designed and implemented a MicroController based Cumulative Grade Point Average (MICGPA) Software calculator for easy and speedy computation of CGPA of students. The MicroController AT895S2 base was programmed with Clanguage and compiled using Resonance Integrated Development Environment (RIDE). The MICGPA software calculator is relatively expensive and cumbersome in realizing the end product.

Our attempt in this paper is to use UML as modeling kit in amalgamation with it analytic and evaluative power revealed in the literatures reviewed aboveto build a model for a calculator software that will empirically computes and compare the two contrasting standards of computing grade point average in Cameroon State Universities. Note that in this paper we do not intend to invent another GPA computation standard but rather comparing and attempting to harmonize two standards that are mentioned in the introductory section of this document.

UML DESIGN MODEL FOR THE GPA CALCULATOR

Grading Standards

There is a great disparity on grading system among state universities in Cameroon. Since there is no globally accepted unified format for computing GPA. Individual grading systems has been adopted unique to their own institution. In Cameroon, for the eight state established universities, the grading system is divided into two categories. See index for sample Result transcripts from the University of Buea and Yaounde respectively.

UML Use Case Diagram

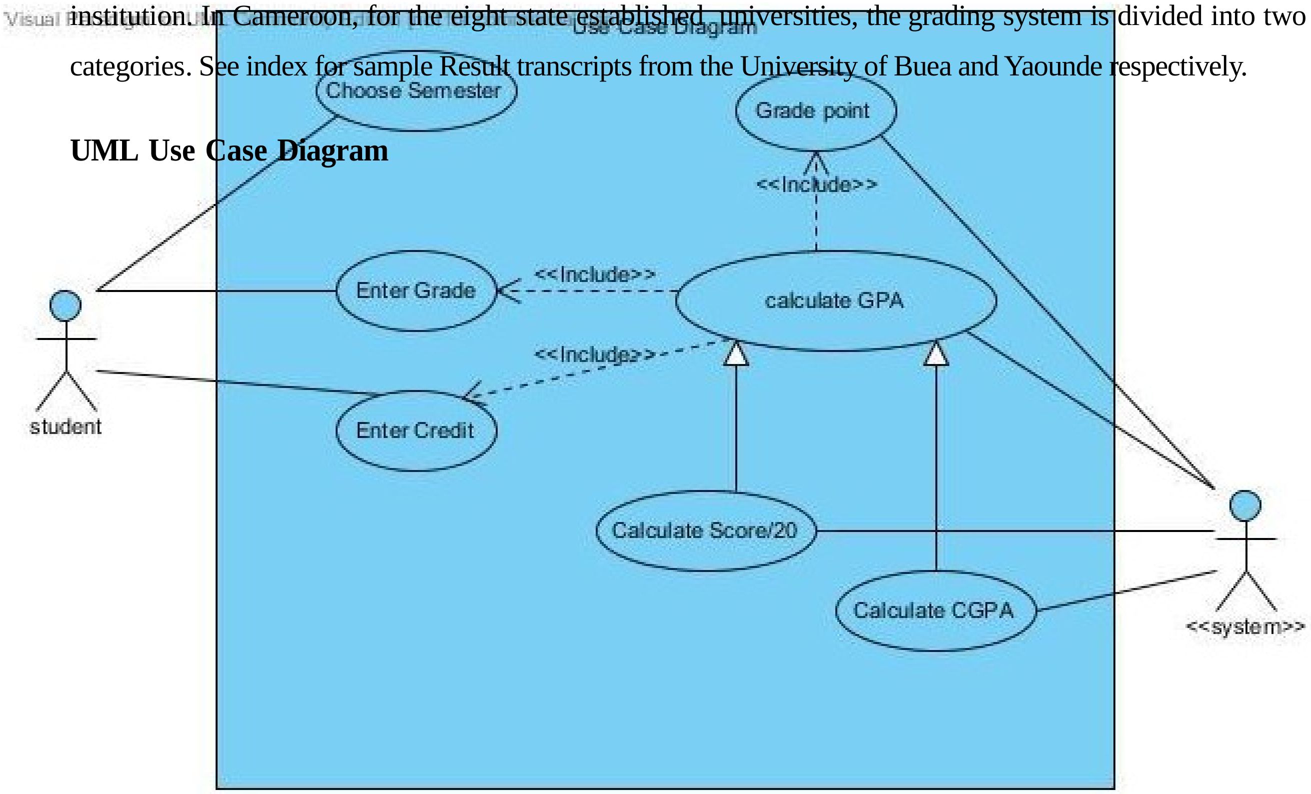

Figure 1. UML Use Case Diagram for the GPA Calculator

This Use case diagram captures the required behavior of the system from the user‘s viewpoint. This diagram is behavioral static in nature. We have aesthetically arranged the use case diagram in such a manner that readers can getsome idea of the process flow. From the above diagram we can see that the system comprise of seven different use cases. The use cases are represented by the oval shaped circles. The calculate GPA use case must include the enter grade,enter credit, and grade point use cases in order for it to be complete. This scenario has been demonstrated with dotted arrows leaving from the calculate GPA use case pointing to the use cases it include. They include stereotype is used and represented with double angle brackets as shown in figure 1 above. We also see from our use case diagram that there two actors involve in the system (i.e. the student actor and the system actor). The association of each of the actors with the system is demonstrated by straight lines that leave from the use cases to the actors on the system. The arrows that leave from the calculate score/20 and calculate CGPA use cases to the calculate GPA use case represent a generalization relationship. That is the calculate score/20 and calculate CGPA use cases inherits some of their attributes and operations from the calculate GPA use case. This means that the "calculate GPA" use case is parent and the "calculatescore/20" and "calculate CGPA"are the children.

UML State Diagram

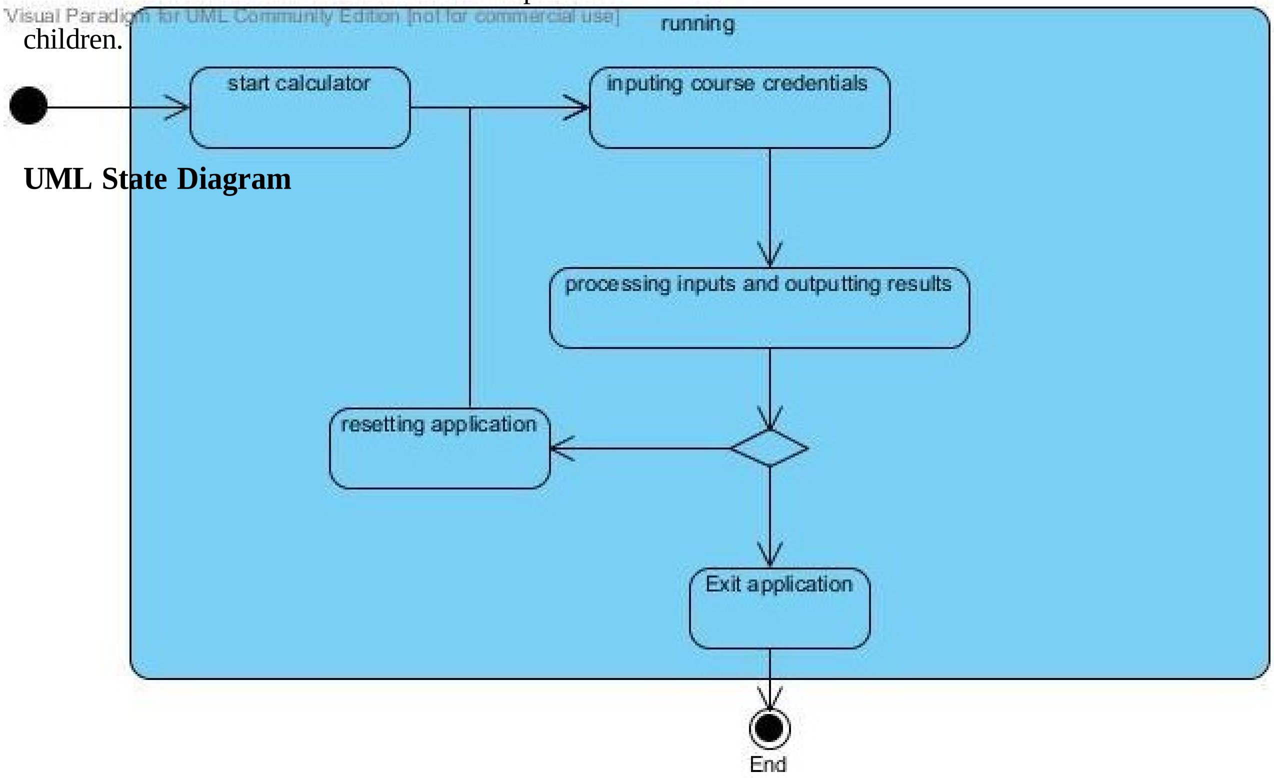
Visual Paradigm for UML Community Edition [not for commercial use]
running
start calculator
inputing course credentials
processing inputs and outputting results
resetting application
Exit application
End

Figure 2. UML State Diagram for the GPA Calculator

This UML state diagram is used to indicate the various states in which the use cases or the entire system could be in at an instance. It shows the ―instance version‖ of the system. From the initial point represented by the boldcircle, thecalculator is started after which it immediately goes to the input coursecredentials state. This is the state were the user will be allowed to input their grade and credit value. The processing input and outputting results state were the system process the grades and credit value to obtain student GPA then follow. It is also at this state that the calculator conversion from GPA to score/20 and viceversa. After which we may choose to close the application or reset the system. Resetting the application will take us back to the "inputting course credentials” state. Running is composite state with all the other states being sub states. Thenature of the state diagram is dynamic behavioral. This diagram also show theentire behavior of one objectdepicting the life cycle of an object as it changes its state in response to the messages it receives.

UML Sequence Diagram

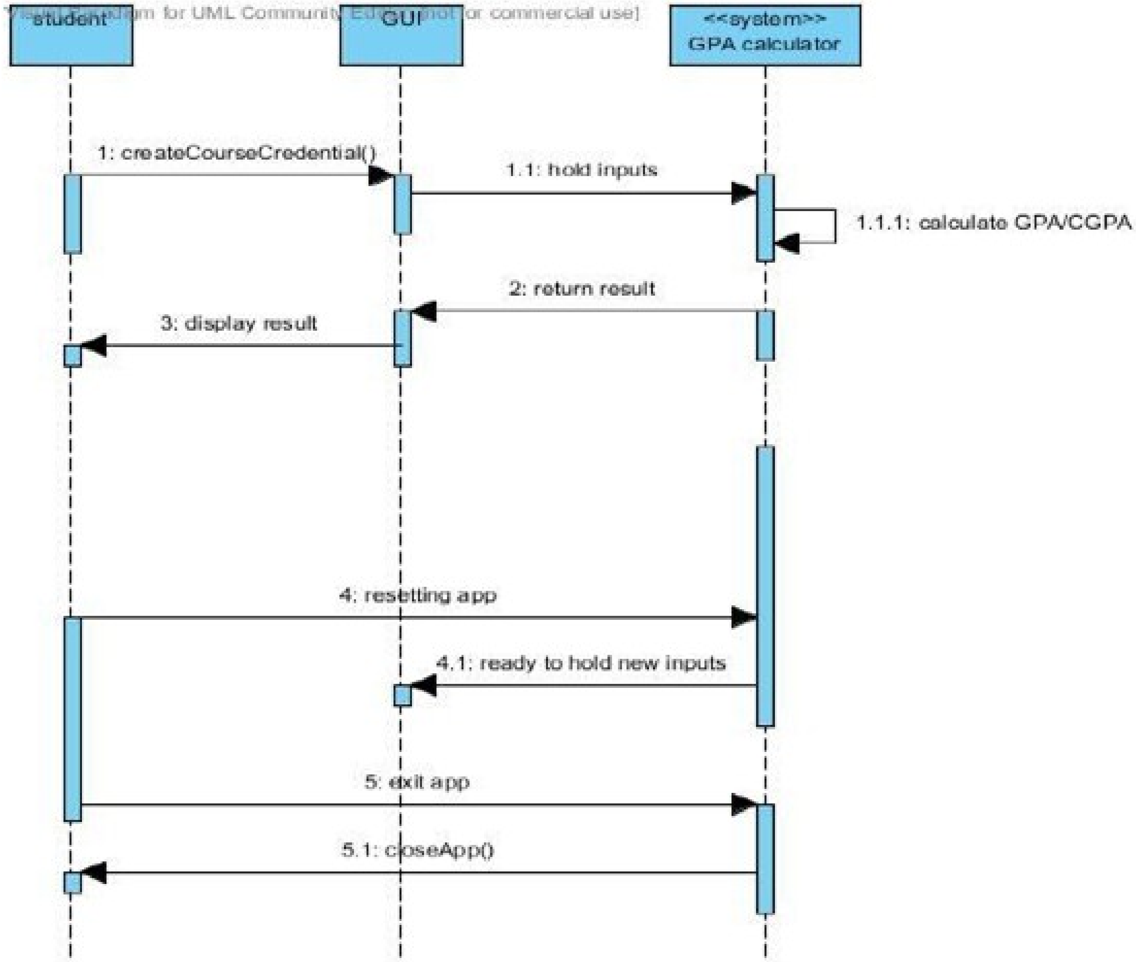
for UML Community Ed [not for commercial use]
student
GUI
<<system>>
GPA calculator
1: createCourseCredential()
1.1: hold inputs
1.1.1: calculate GPA/CGPA
2: return result
3: display result
4: resetting app
4.1: ready to hold new inputs
5: exit app
5.1: closeApp()

Figure 3. UML Sequence Diagram for the GPA Calculator

The sequence diagram has a practical ability to show what is happening in theuse cases. The horizontal dimension shows the classifier roles that represent theindividual actors and/or objects collaborating with each other. This diagram makes it possible see what happens between two messages (e.g. when the create createCourseCredentials() message is sent, the GUI object holds the inputs and the calculator (system) process the GPA and CGPA. The system then returns the computed results back to the GUI object which then displays them to the user(student)). The message direction is indicated by the arrows shown in figure 3 above. This sequence diagram also ascertains us of what happens as time progresses. This is demonstrated by the vertical dotted lines. Sequence diagram is dynamic-behavioral in nature.

DISCUSSIONS

Although a good number of software applications has modeled using UML, some developers still find reasons why they should not use UML in modeling their own designs. We have used UML in this model for several reasons;

It‘s the standard. UML is the de facto standard notation for software design.In other words, it‘s the only software design notation that you can expect your peers to be familiar with.

Tool support. Besides drawing diagrams, some of these tools including visual paradigm can generate code, apply design patterns, mine requirements, reverse engineer code, and perform impact analysis, refactoring, and complexity analysis. The ability to generate code from the design found in many UML tools alone is a good reason for our choice of UML.

UML is flexible. Stereotypes and profiles let us tailor UML to the required needs. In other words, it provided us modeling elements and relations that are specialized for the system we are building.

Just Need a Subset of the Notation. UML has 14 different types of diagrams.You won‘t find anyone who uses all 14 types to document a software system. In our model we have use just approximately 20% of the UML language to cover 80% of our modeling needs. Subsets of the notation effectively communicate thedesign using UML.

Architecture is important. The software architecture is the blueprint for the system; it allows reasoning about performance, availability, security, and otherattributes, allows planning for incremental development, and guides work assignments and tracking. However, even the best architecture, created based on years of experience and carefully selected design patterns, may be useless if not properly communicated to the people who need to use it. Designing architecture without documenting it is like winking a lady in the dark. You know what you‘redoing, but nobody else does. UML is a wellknown visual language that cancapture much of the information that one needs to communicate about the architecture.

We are looking forward to transforming the UML model of the GPA calculator software into a bona fide GPA calculator.
We will also extend the work on how to sensitizing the university community in Cameroon on how to effectively deploy and use the GPA calculator to track their GPA as theprogress in an academic year. We also hope to extend the study on how the GPA calculator application will be accepted and adopted in our state universities.

CONCLUSION

In this paper as a very suitable tool for modeling GPA calculator software. We have also highlighted the importance of a GPA calculator for student at the tertiary level of education. We explain in depth, the design UML design model, how the different UML diagrams are used to model a GP A calculator software system. We also show how these diagrams are used in conjunction to arrive at a final bona fide GPA calculator software system. It also tries to correct some of the common fallacies and myths about GPA computation in Cameroon state universities, attempting to harmonize two different standards of computing GPA in Cameroon state universities. Finally , we hope to have provided an adequate answer to the question —can there be a unified standard for calculating GPA in Cameroon state universities?

ACKNOWLEDGEMENT

The authors owe much thanks to Prof. Nalova Lionga, Vice-Chancellor, University of Buea, for providing excellent working facilities in the University campus. Thanks are also due to the Catholic University Institute of Buea for providing work space for this work.

REFERENCES

1) Dr. Vipin Saxena, et al (2009), UML Modeling and Performance Evaluation of Multithreaded Programs on Dual Core Processor, International Journal of Hybrid Information Technology
Vol.2, No.3B.B. Ambedkar University (Central University), Lucknow, India.

2) Sunguk Lee (March, 2012), Unified Modeling Language (UML) for Database Systems and Computer Applications, International Journal of Database Theory and Application Vol. 5, No. 1 Research Institute of Industrial Science and Technology Pohang, Korea.

3) Udrica Mioara, et al (June 2012), Issue 3, Improving The Decisional Process By Using UML Diagrams, Scientific Papers (www.scientificpapers.org) Journal of Knowledge Management, Economics and Information Technology.

4) Dian Pratiwi Trisakti (May 2013), Implementation of Function Point Analysis in Measuring the Volume Estimation of Software System in Object Oriented and Structural Model of Academic System, International Journal of Computer Applications (0975 – 8887) Volume 70– No.10,

5) Dr. Vipin Saxena, et al (Jan 2011): Object –Oriented Database Representation through UML, International Journal on Computer Science and Engineering (IJCSE) Vol. 3 No. 1

6) Muhammad Usman, et al (April, 2009), Automatic Generation of Java Code from UML Diagrams using UJECTOR, International Journal of Software Engineering and Its Applications Vol.3, No.2

7) G. N. Ezeh, et al (September 2012), Microcontroller Based Cumulative Grade Point Average (Micgpa) Calculator, Natural & Applied Sciences Vol. 3, No. 2

8) Azrul Hazri Jantan (January 2012), An Extension of UML Modeling for Web Hypermedia Design: A Case Study, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, International Journal of Information and Communication Technology Research, Volume 2 No. 1

9) S. R. Kodituwakkuand P Chamii'ara I (2009), Design and Implementation of the Course Unit Based Results Analysis System, Caylim Journal of Sciences: Physical Sciences, 15 27-39

10) Bhuvan Unhelkar, PhD, Verification and Validation for Quality of UML 2.0 Models, a Wiley-Inter science Publication, ISBN 0-471-72783-0, JOHN WILEY & S-ONS, INC.

INDEX