Design and Implementation of Students' Information System (SIS) for Mulungushi University Based on Spring Framework

Douglas Kunda
School of Science, Engineering and Technology
Mulungushi University
Kabwe, Zambia
Email: dkunda [AT] mu.ac.zm

Victoria Chama
School of Science, Engineering and Technology
Mulungushi University
Kabwe, Zambia

Abstract—**Student Information Systems (SISs) are computer software applications for learning and education institution to manage student records such as registering courses, assessment monitoring, storing examination results and managing many other student-related information needs in a school, college or university. This paper reviews three popular SISs namely OPENSIS, Fedena and OPUS. Among these SISs, there are many common features, as well as distinct features. This paper discusses the design and implementation of a student information system for Mulungushi University based these reviewed system and their unique requirements. The SIS for Mulungushi University supports the management of student information through key business activities such as admission, registration, invoicing, accommodation, progression and graduation using the spring framework with MySQL as a database and incremental developmental approach. The paper presents the design and implementation of the system and the challenges in developing the system and the tests conducted as well as the lessons learnt.**

Keywords-Student Information Systems; Spring Framework; MySQL database; Model View Controller (MVC)

I. INTRODUCTION

Learning institutions such as universities and colleges use student information systems for the management of student records. There are a number of different student information systems that exist some are proprietary and others are open source. With the constant development of countrywide higher education, in order to resolve the trouble of "Information Island", the teaching branch of many universities established correspondingly a series of information system to manage teaching and administration better, and so on [1].

According to [2] “issues related to student academic record managing include improper course registration, delayed publication of students’ exam results, inaccuracy due to manual and tedious computation of fees and retrieval of information problems and inefficiencies” .She goes on to add that there are a wide collection of existing information systems that have been developed to address these problems but vary in implementation, target group, level of integration and customizations capabilities.

Such systems are referred by many names however have a common goal which is to improve the efficiency and effectiveness in which universities deliver its services to students or how management performs administrative tasks. Names of these systems include Campus Information System (CIS), Student Information System (SIS), Student Administration System (SAS), University Classroom Management Information System, Education Management Information System (EMIS) and Student Information Management System (SIMS)

According to [3] student information management system should be able to offer users with satisfactory information and efficient query tool. For existing systems and system review, this paper focuses on three examples of student information systems. These are OPUS, OPENSIS and Fedena.

OPUS-College is a free and open source information system for the registration and management of student, staff and course data within a university or other institute of higher learning. OpenSIS is a free, open-source platform designed for K- 12, regardless of if they are charter, private, or state-run schools. Fedena is “open Source Software” for school management. The technology behind the development of Fedena is Ruby on Rails, a powerful web application development framework.

Reference [4] defines incremental development as “developing systems through incremental releases, first providing essential operating functions, then providing system users with improved and more capable versions of a system at regular intervals. For the design and implementation of the student information system first identification of the core modules to be developed will be done in the initial phase. There after a general over view of how these modules are expected to work together in the system to provide certain functionality. Next priority is assigned to each module which will give the order in which the increments will be developed.

System testing is intended to ensure that the implementation of the system is indeed as the user wants. To achieve this, the designed system was tested in three iterations following each increment (i.e. after the development of each module).

II. Reviewed Student Information Systems

For existing systems review, this paper focuses on three examples of student information systems. These are OPUS, OPENSIS and Fedena.

A. OPUS

OPUS-College is a free and open source information system for the registration and management of student, staff and course data within a university or other institute of higher learning. The general or generic name of the system is "OPUS-College", where OPUS stands for "OPen University Systems", indicating the open source character of the system. The Mozambican or Portuguese instance or implementation of the system is called "eSURA", which in Portuguese means: (electronic) system for academic registration [5].

OPUS-College was initially developed as part of a development project for Mozambican universities, founded by NUFFIC, the Dutch governmental organization for university development cooperation. Given the relative success of the Mozambican project, NUFFIC decided to also choose OPUS-College as the Student Information System to be implemented at the Zambian universities, within the framework of a new development project, which started early 2010 [5].

According to [5] OPUS-College covers a broad range of functionality concerning academic administration issues and tasks in an integrated way: full registration and update functions for student, staff, studies, courses and exam information in one system.

- Opus-College is a web-based information system for the registration and consultation of information on:
- Students (personal data, study plan, previous educational career, absence registration, etc.).
- Studies (structure and content: programmes of study, courses, exams, tests).
- Lecturers (staff members involved in the academic education process).
- Organizational units (Schools, Departments, Institutes).

The structure of OPUS-College comprise of a Core Module or Kernel common for all institutions implementing OPUS-College and dealing with all data registration and update functions. Additional Modules, which can be specified and tailored to the needs and situation of a country or even an individual institution. Currently the following additional modules are under development for OPUS-College:

- An Online Registration Module allowing students to register/subscribe through the internet.
- A Reports Module which holds all the output functions of OPUS-College and which can be tailored to the needs and (style) requirements of each individual university.
- A Scholarship Module for registering information about student's scholarship (Mozambican situation).
- A Fees Module for registering information on the fees paid (to pay) by students.
- An Accommodation Module for registering information on housing of students on campus.

B. OPENSIS

OpenSIS is a free, open-source platform designed for K-12, regardless of if they are charter, private, or state-run schools. OpenSIS is best suited for small and medium schools with a knowledgeable IT staff. While OpenSIS offers both free and paid versions of its software, its free version has substantial features. Schools can use this software to maintain transcripts, health records, attendance, demographic information, scheduling, grade books, and custom reports [6]. According to [5] OPENSIS is used in South Carolina and provides the following main components.

Core SIS features: OpenSIS is used to manage basic student information as well as the following:

- Scheduling – students can create single student schedulers or group schedulers.
- Health records - capture student's immunization, allergies, special medical conditions, physician contact information and nurse visits, and
- Special education functionality - track goals and progresses for students with special needs or attention.

Extensions: OpenSIS extends functionalities in the core features in order to enable teachers to create their own website to communicate with students and parents. It can also document and track disciplinary incidents, and notify parents of infractions. In addition, OpenSIS allows the integration of its SIS with other applications, e.g., WordPress, Moodle LMS and Google Checkout, to support customization.

Data integrations: OpenSIS enables its integration with state information system through the state reporting feature, which is based on the Extract-Transform-Load (ETL) concepts which consists of four data processing steps, i.e., collection, extraction, transformation, and load and submission [8].

Reference [8] provided a comprehensive review on the SISs developed in the US and the UK. The selected systems are chosen based on their outstanding features.

C. FEDENA

Fedena is "open Source Software" for school management. The technology behind the development of

Fedena is Ruby on Rails, a powerful web application development framework. Fedena due to its unique features and user friendliness is implemented by more than 40,000 institutions around the world.

The application is composed of a basic module which satisfies all the necessary requirements of a simple SIS. An e-mail system is also available with the Fedena through which mail can be sent and received by the users. Fedena provide following core modules with easy user interface that make working easy for all kind of users of software like Faculty, Staff and Students. These modules can be customized without difficulty as per the need of the organization. Fedena also provides local regional languages. One can devolve their own application which can be integrated with original Fedena [9].

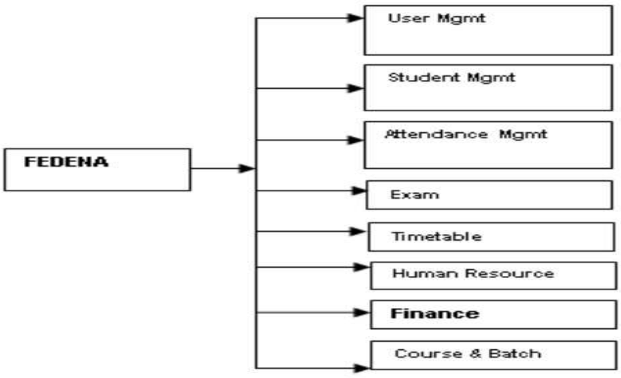

Figure 1. Core Modules of Fedena [9]

III. COMPARISONS OF REVIEWED SYSTEMS

This section will focus on comparing the three systems reviewed and compare some of the specification and functionalities with proposed system. Table 1 below presents the three systems reviewed and the proposed Mulungushi University SIS.

TABLE I. SYSTEM COMPARISONS

Characteristics	OPUS	OPENSIS	Fedena	Proposed System
Application Type	Web	Web	Web	Web
Frameworks	Spring, Ibatis		Ruby on Rails	Spring, Hibernate
Language(s)	HTML, JavaScript, JSTL, EL	Php	Php	HTML, JavaScript, JSTL, EL
Data Storage	PostgreSQL	MySQL	MySQL	MySQL
Operating Platform	All	All	All	All
Demographic	Management, Lecturers, Student	State, Management, Lecturers, Student	Management, Lecturers, Student	Management, Lecturers, Student
Student Management	Yes	Yes	Yes	Yes
Examination Management	Yes	Yes	Yes	Yes
Admission / Enrollment Support	Yes	Yes	Yes	Yes
Course Registration Management	Yes	Yes	Yes	Yes
Accommodation Management	Yes	No	No	Yes
Timetable	No	Yes	Yes	Yes
Attendance	No	Yes	Yes	No
Upload Exam Results Support	No	No	No	Yes
Parent/Sponsor Portal	Yes	Yes	Yes	Yes
Financial Management (i.e. PayPal/Bill Master)	Yes	Yes	No	Yes
Study Plan Support	Yes	No	No	Yes
Integrated SMS/E-mail Communication Support	No	Yes	Yes	Yes
Reports and Inquiries	Yes	Yes	Yes	Yes

The following are the reasons that affected the adoption of the reviewed student information systems for Mulungushi University:

A. OPUS

According to [10] an in-depth analysis of the specific Zambian (University of Zambia and Copperbelt) requirements was taken in the course of 2010, the results showed that quite a lot of adjustments and extra functionality where necessary to build the system appropriate for the Zambian situation. Planning of the implementation of OPUS-College at your institution and the organizational framework required for a sustainable functioning of the system requires an entire year and two months for it to be fully operational [10].

B. OPENSIS

To take full advantage of OpenSIS Community, Information Technology staff members will have to be familiar with PostgreSQL. The open-source version does not offer class portals, billing, discipline, or state reporting [6].

C. FEDENA

Fedena is the free version of another school management software of the identical name. When comparing the open-source version to the paid version, it becomes evident that the open-source version is lacking in a number of features, including accommodation, custom reports, registration, inventory and discipline [6].

IV. Design and Implementation of SIS for Mulungushi University based on Spring

A. Spring Framework

According to [11] Spring was created to address the complexity of enterprise application development, and makes it possible to use plain-vanilla JavaBeans to achieve things that were previously only possible with Enterprise Java Beans. But Spring's usefulness isn't limited to server-side development. Any Java application can benefit from Spring in terms of simplicity, testability, and loose coupling.

Spring simplifies Java development through:

- Lightweight and minimally invasive development with plain old Java objects (POJOs)
- Loose coupling through dependency injection and interface orientation
- Declarative programming through aspects and common conventions
- Boilerplate reduction through aspects and templates

Similarly according to [12] spring is described as a lightweight framework for building Java applications, but that statement brings up two interesting points. First, you can use spring to build any application in Java (e.g., stand-alone, Web, JEE applications, etc.), unlike many other frameworks such as Apache Struts, which is limited to web applications. Second, the lightweight part of the description doesn't really refer to the number of classes or the size of the distribution, but rather, it defines the principle of the spring philosophy as a whole—that is, minimal impact. Spring is lightweight in the sense that you have to make few, if any, changes to your application code to gain the benefits of the Spring core, and should you choose to discontinue using Spring at any point, you will find doing so quite simple.

B. System Analysis and Design

The aim of designing a new information system must be to produce something that meets the needs of the people who will be using it. In order to do this, we must have a clear understanding both of the overall objectives and of what it is that the individual users of the system are trying to achieve in their jobs [15].

To collect requirements for the Student Information System, observations and interviews were the techniques considered and used. After a deeper understanding of how the system is to carry out functions and tasks. The main areas that the project was set out to achieve are:

- Invoicing: Provide quotations per semester to students as well as invoice students.
- Course Registration: Provide an automated complete course registration process
- Examination: Provide an examination module that allows lecturers to manage student exam results

The method used for data analysis included use cases and scenarios. The use case diagram for the requirements gathered and analyzed for the student information system is provided in figure 1.

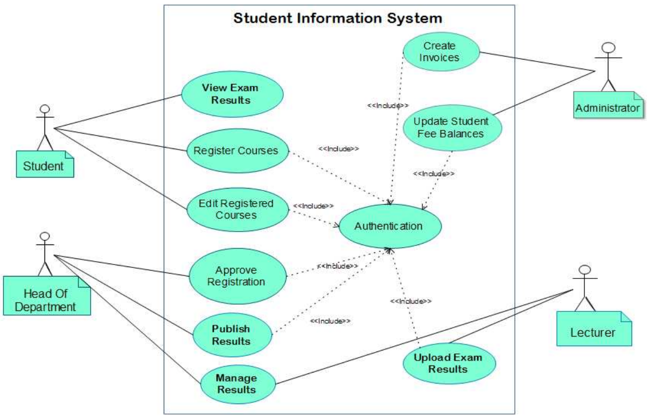

Figure 2. Use case diagram for Mulungushi University SIS

C. Database Design

The proposed system used MYSQL because it is a robust and versatile object relational database management system that is an enormous environment with unlimited potential that delivers information with very high precision and dependability.

According to [14] MySQL is the most popular "free" RDBMS in the world, is not an open source product, but it may be used without charge for non-commercial use on non-Windows platforms under the terms of the GNU Public License. MySQL is known for its clear and uncomplicated administration, which is also valuable in the academic environment. Reasons for adoption include:

- MySQL is also less demanding on resources [14].
- A clear advantage of MySQL is that it may be obtained free of charge for most Unix-based systems. Code can be easily downloaded from the MySQL web site, and supporting materials are freely available online [14].
- MySQL also has many Application Programming Interfaces (APIs) to give the developer to access and shape the database via programs in various languages [15].

D. System Architecture

According to [16] the model view controller (MVC) design pattern can weaken the coupling among the different application tiers and make the development and maintenance become simpler, and it has been popularly used in designing Web-based J2EE application. Reference [17] also agrees with this stating that the MVC is a fundamental design pattern for the separation between user interface logic and business logic. Since applications are very large in size these days and the MVC design pattern can weak the coupling among the different application tiers of application. Figure 3 shows the system components based on the Spring MVC pattern.

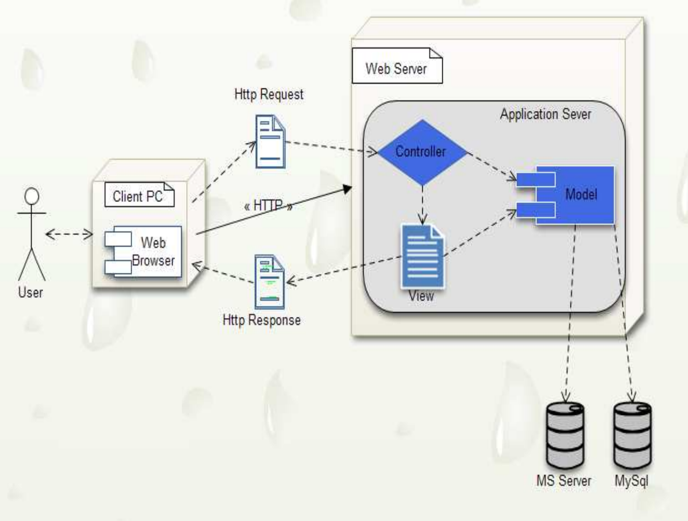

Figure 3. Mulungushi University SIS System Architecture

E. Challenges Encountered

There were two major challenges faced in designing and implementing the Student Information System. These include: 1) The learning curve in the use and application of spring and hibernate framework. 2)The constant change in requirements by the users.eg the criteria on how to invoice student changed from the enrollment year to according to programme and finally to the highest academic year attained or what to include in the exam result template to be used by lecturers when uploading results and which method to use. (Whether to use a dropdown menu or simple links to courses)

V. TESTING OF THE SYSTEM

System testing is intended to ensure that the implementation of the system is indeed as the user wants. Test cases are specifications of the inputs to the test and the expected output from the system (the test results), plus a statement of what is being tested, these cases help the developers equip themselves in discovering the program imperfections [18]. The results of the test case are what are known as test result.

The following are the Test Cases performed for the system:

- If all users are able to be authenticated using credentials.
- If students can get quotations
- If students can pre-register per semester
- If students can see all academic information in single login.
- If lecturers can approve courses
- If lecturers can upload results
- If system is able to generate invoices
- If system is able to update student balances
- If admin is able to manage (add, edit, delete) system records.

The result of the Mulungushi University SIS system test cases is presented in Table II. It will be noted that the developed system passed all the test cases.

TABLE II. STUDENT INFORMATION SYSTEM TEST CASES

Input	Code / Method	Expected Output	Actual Output	Result
Input of username and password by user (student, lecturer. administrator)	Login()	Login failed. Check that your username and password are correct	When user enters in correct credentials. The system displays the correct user view	Pass
Get quotation per semester	doGetQuotation(studentNumber)	Field must not be Empty	The quotation is presented to the student with an expected total amount	Pass
Pre-register	getCourseRegistrationCourses(studenrNumber)	You have already registered for this semester. To make changes edit your course registration	Complete course registration process including invoicing.	Pass
View Student Details	viewDetails()		Displays student details	Pass
Approve Courses	approveCourseRegistration()	A list of registered students	A list of all approved registrations	Pass
Upload Results	uploadExams()	Please check that the format of csv file is in the approved format	A summary of successfully uploaded results	Pass
Generate invoices	createCSV()	No invoices to create	Zip file containing student invoices	Pass
Update Student Balances	updateStudentBalance()	Student balances have no begin updated	A confirmation message of successful update	Pass
Manage records	manageRecords()	Error in add/updating or deleting record	A list of added/edited or deleted records	Pass

VI. LESSONS LEARNED

The design and implementation of the system and the student information system using the spring framework brought out a number of lessons. These lessons include:

1. The use of incremental approach when developing a system with a large number of modules will minimize the risk *of project failure.* According to [18] the incremental approach in designing and implementing is beneficial because with the incremental approach changes can easily be made and users of the system are able to interact with the system which helps refine requirements. Furthermore, with incremental approach it is easier to test and debug during a smaller iteration and easier to manage risk associated with the design and development.

2. Close consultation with the users of the system and representatives from the organization helps build ownership and commitment. Extensive user participation in systems development and testing is essential for a viable end product. When developing applications for enterprise use, testing is an important way to ensure that the completed application performs as expected and fulfills all kinds of requirements (architectural, security, user requirements, and so on) [12].

3. Were it is applicable use as many existing reusable components to develop applications faster. Reusable software components have the obvious advantage of reducing the amount of software to be developed and so reducing cost and risks. It usually also leads to faster delivery of the software [18]. This project used hibernate technology that accelerated the development process through re-use of existing components.

4. Integration and use of the spring and hibernate frameworks. According to [12] spring is described as a lightweight framework for building Java applications, but that statement brings up two interesting points. First, you can use spring to build any application in Java (e.g., stand-alone, Web, JEE applications, etc.), unlike many other frameworks such as Apache Struts, which is limited to web applications. Second, the lightweight part of the description doesn't really refer to the number of classes or the size of the distribution, but rather, it defines the principle of the spring philosophy as a whole—that is, minimal impact. According to [19] hibernate is an ambitious project that aims to be a complete solution to the problem of managing persistent data in Java. It mediates the application's interaction with a relational database, leaving the developer free to concentrate on the business problem at hand.

VII. CONCLUSION AND FURTHER WORK

In general, student information systems are created with a purpose to hold details of quality and accurate information and make them accessible as an integrated system. The system helps the administrators, lecturers and policy makers make accurate, fast analysis and decisions about university and students by enabling them to carry out their task easily, efficiently, and timely manner.

To implement the Mulungushi University SIS, it was decided to apply incremental developmental as a software development methodology, java for programming language, Spring framework and MySQL database server for the storage of data. The project was completed and the following modules namely student course registration, invoicing, interface with ERP Accounting software, examination system are operational.

Further work will include development of the following modules:

1. Admissions module

The admission module is responsible for the admission of students into the university. Admissions are done both manually and automatically. If students apply online their details are to be automatically entered into the university database and linked to the accounting software Sage 300. This module will have a notification system which should be able to send emails or text messages to approved or rejected admissions to applicants.

2. Automatic course registration for all first year students.

If all payments have been made by newly admitted students, they should be automatically registered by the system.

3. Integration of Student Identity cards with catering and access to examination room. Currently the University provides meal cards to all student who eat from the cafeteria and administration of meal cards is complex and therefore needs automation. course registration and examination. With the use of student cards that have bar codes students can use their ID's for getting meals at the cafeteria can print registration information as well as use them as an entry into the examination room.

REFERENCES

[1] Liang, J, S Guo, L Li, S Yang, The Research and Design of the University Classroom Management Information Platform Based on .NET. Advances in information Sciences and Service Sciences, 4 (21), 2012, pp. 422-427

[2] Vecchioli L. A process for evaluating student records management software. ERIC Clearinghouse on Assessment and Evaluation; 1999.

[3] Yang, C., Design and Implementation of Students' Information Management System (SIMS) Based on ASP. International Conference on Economic Management and Trade Cooperation (EMTC 2014). Atlantis Press

[4] Basili, V R, and Turner, A J,, Iterative Enhancement: A Practical Technique for Software Development, IEEE Trans. Software Engineering, 1(4), 1975, pp. 390-396.

[5] OPUS [online] Available at: from < The official site for OPUS-College. Available from: http://www.opuscollege.net/OPUS.php?page=docsENG> Accessed on [Accessed September 14, 2016]

[6] Burger, R., Capterra School Administration Blog. [Online] Available at <http://blog.capterra.com/thetop6freeschooladministration-software/> [Accessed September 29, 2015]

[7] OPENSIS [online] Available at: from < The official site for OpenSIS. Available from: http://www.opensis.com/features.php> Accessed on [Accessed September 14, 2016]

[8] Kusakunniran W, Thongkanchorn K, Kongdenfha W. Review of Student Information System for Basic Education. InBUU-2014 2014 Jul 3.

[9] Saxena, V et al., Study of School Based Service Software: With Reference To Fedena. Journal of Engineering, Computers & Applied Sciences, 2(6), 2013, pp. 66-71

[10] Simons, E., OPUS-College: Student Information and Registration System. Radboud University Nijmegen (NL), 2011

[11] Walls, C., Spring In Action. (3rd Edition ed). Dreamtech Press. Manning Publications, 2011.

[12] Ho C and Harrop R, Pro Spring 3. Apress, 2012.

[13] Benneet, S et al,. Object-Oriented Systems Analysis and Design. (3rd Edition ed.). McGraw-Hill, 2010.

[14] Deaton, JW and Peace, AG., Journal of Information Systems Education, 14(4), 2003.

[15] Suehring, S.. MySQL Bible. Wiley Publishing, Inc. New York

[16] Huang, S and Zhang, H. (2008). Research on Improved MVC Design Pattern Based on Struts and XSL. International Symposium on Information Science and Engineering, 2002

[17] Gupta,P and Govil, M.C, International Journal on Computer Science and Engineering. 2(4), pp. 1047-1051, 2010.

[18] Sommerville, I., Software Engineering (9th Edition ed.). Boston: Pearson Education, 2011.

[19] Bauer, C and King, G.. Hibernate in Action. Manning Publications Co., 2005.

[20] Amin, A. and Cohendet, P., Architectures of Knowledge, Oxford University Press, Oxford, 2004

[21] Star, S.L. and Ruhleder, K., “Steps toward an ecology of infrastructure: design and access for large information spaces”, Information Systems Research, 7, 1996, pp. 111-135.