Development of Students Result Management System: A case study of University of Uyo

Akpasam Joseph Ekanem, Simeon Ozuomba*, Afolayan J. Jimoh

*Corresponding Author: simeonoz@yahoo.com

Department of Electrical/Electronic and Computer Engineering, University of Uyo, Akwa Ibom, Nigeria

Abstract

In this paper, an automated platform for managing result of all categories of students in a seamless and interactive manner is presented. The system was developed using PHP, CSS, HTML, MYSQL and was hosted locally using Apache web server. The software development methodology adopted is a participatory incremental process model (PIP Model). The data used were obtained from the Department of Electrical/Electronic and Computer Engineering, University of Uyo. Functional decomposition of the system and its key modules are provided to explain the major functionalities proffered by the system. Also, use case diagram is presented to show the different categories of the system users and the various functionalities associated the different system user. Screen shots of various system functionalities are presented from the test run of the functional system.

Keywords: Result Management System; Participatory Incremental Process Model; Functional Decomposition; User Access Management System; User Access Privilege

1 Introduction

As universities in developing countries like Nigeria embrace information and communication technologies, they employ it in transforming their record keeping and information management operations. Particularly, students result management system (RMS) is one of the key information systems that readily attracts the attention of most universities [1,2,3,4]. While the universities in the developed countries already have well established students result management systems, their counterparts in the developing countries are having series of adhoc and standalone result management systems that provide only some basic elements of students result management system. Equally, majority of the students RMS do not have web interfaces and others do not integrate well with the other university-wide information system. The major challenge is that the universities outsource the development of such RMS without proper understanding of the essential functionalities and compatibility issues the RMS should afford to the university [5,6,7,8].

In this paper, a sample student RMS design is presented. In the system, the department of Electrical/Electronic and Computer Engineering University of Uyo is used as a case study. A detailed functional decomposition of the system is presented...
along with use case diagram. Also, the participatory software development methodology [9,10,11,12,13] used in the system development is presented. Such development approach will afford the system stakeholders in the university to participate effectively in the design process even when the software is outsourced. Finally, numerous screenshots of the developed student RMS are presented to further demonstrate the possible interface features the need to be built into such RMS.

2 Methodology

2.1 Participatory Incremental Process Model

The thrust of this paper is the development of Student Result Management System (SRMS) tailored to organize and manage students’ result and transcript in the various departments in the University of Uyo, using Electrical/Electronic and Computer Engineering department as a case study. Participatory incremental process model (PIP Model) in Figure 2.1 is used in the system development.

Figure 2.1: Participatory incremental process model (PIP Model)

The system stakeholders’ include, the Software developer, supervisor, students, exam officers, school management and lecturers. The PIP model begins with requirement engineering which embodies all activities carried out by the developer in understanding, analyzing, validating and properly replicating the user’s mental model of the system. The activities in requirement engineering stage include analysis of existing system, requirement elicitation, requirement analysis, requirement validation and verification, as well as requirement specification, which gives rise to requirement documentation and requirement management. Interviews, questionnaire, stakeholders’ group discussion, social analysis, evaluation and inspection of documents are used to elicit the user and system requirements. The requirements are analyzed to ensure that the requirement specification is free of inconsistencies, omissions, redundancy, ambiguity etc. Stakeholder’s feedback and comments are
sought for at each stage of the requirement engineering process to facilitate a complete requirement specification that meet stakeholders’ needs. Every conflict in the requirement specification is reconciled through the process of negotiation. Also, measures are put in place to identify, control and track requirements and requirement changes at any time in the system lifecycle.

2.2 Functional Decomposition of RMS
The functional decomposition \([14,15,16,17]\) of the result and transcript management system is presented in Figure 2.2 and the modules are further decomposed into its respective sub-modules as shown in the subsequent figures.

![Figure 2.2 Functional Decomposition of the system](image)

The system can be accessed by both students and staff. The staff can play three different roles in the system, namely, the system administrator, as the exam officer or as the course lecturer. Each of these roles constitute system module that is further decomposed into sub-modules. The staff must ‘registered on the RMS platform and then login with the staff username and password. The staff access privileges depend on the role the staff plays in the system. The the system administrator role has the highest access privileges.

Also, the students can access he system as ‘student’. The student must ‘registered on the RMS platform and then login with the student username and password. In the system, the student will have student privileges which will enable the student to register for course, drop registered course if required, view the result pertaining to the student alone, change the student’s settings like the photo, password and student’s contact address a, email and phone number.

2.3 Registration and Login Module
Whether a student or staff, the system policy requires that every user should register to create user account and then use the user account to login each time the user want to access the system functionalities. Using the students’ module as an example, the registration process is done by obtaining a registration code from the department and filling a form of personal information shown in the “user registration” page of the
result/transcript management module of the portal. If the user already has an account, the user can login using the user username and password. Figure 2.3 shows the registration and login process. For security reasons, a registration code is used during the registration process. This code is an automated code integrated by the system developer and can be generated by the portal administrator to ensure that only authorized persons can register in the portal.

![Diagram of Registration and Login Module]

**Figure 2.3 Registration and Login Module**

### 2.4 Administrator Module

The administrator module (Figure 2.4), also referred to as the admin module, is the module meant for the back-end system administrator that manages the control panel of the software. The admin module has functionality for generating and managing the registration code which is used to ensure that only bonafide students and staff can register in the system. The admin module also manages the courses, semesters, academic session and other key things the users can access on the system.
Apart from the admin module the staff can play the role of the course lecturer and the exam or result officer. Each of the roles has different user privileges. The functionalities associated with the course lecturer module and the exam officer module are shown in Figure 2.5 and Figure 2.6.

**Figure 2.4 Admin sub-module**

**Figure 2.5 The Exam Officers Module**

**Figure 2.6 The Lecturers’ Module**

### 2.5 Use case Diagram

The use cases of the system are identified to be “Registration”, “Register Courses”, “Generate/Print Transcript”, “Record Test/Exams scores”, “Generate/Print...
Mastersheet”, “Manage Semester/Sessions”, “Manage courses” “View/PrintResult”, “ManageDeadline”, and “GenerateRegistrationCode”. The use case diagram of the system is depicted in Figure 2.7.

![Use case diagram](image.png)

**Figure 2.7 Use case diagram**

### 3 Result and Discussion

A functional result management system was developed using PHP and MYSQL as the side-server and HTML and CSS as the client-side. The database used in the RMS was designed with MYSQL Database Management System (DBMS). All the information pertaining to the RMS are stored in the system database. The sample output/results are shown in the various screen shots presented in this section. Figure 3.1 shows the student registration and login page.
Figure 3.1 Student registration/login page

Figure 3.2 shows the registration module. For student or staff to register on the portal, the user must first enter a registration code given to the user by the portal administrative officer as shown in Figure 3.2.

Figure 3.2 Registration module

Student registration form is presented in Figure 3.3. A student must fill the form to be registered on the system.

Figure 3.3 Student registration form
Figure 3.4 shows staff registration/login page

Figure 3.4 Staff registration/login page

Figure 3.5 shows student dashboard after login

Figure 3.5 Student dashboard

Figure 3.6 shows student course registration module.

Figure 3.6 Student Course registrations
Figure 3.7 shows the “drop registered courses” module. Student can drop registered courses here but if result has been recorded by the lecturer for that course then the student cannot drop the course anymore.

Figure 3.7 Student drop courses module

Figure 3.8 shows “view result” module. Student can view result by selecting the session and semester and clicking the go button. Student can only view the result but cannot print as the print, right click, ctrl+x, print screen, ctrl+A, highlight and ctrl+p option has been disabled for the page.

Figure 3.8 Student view result module

Figure 3.9 present the functionalities of the administrative module. It shows the administrative dash board which provides information such as the total number of registered users, faculties, departments and courses.

Figure 3.9 Student view result module
Figure 3.9 Administrative dashboard module

Figure 3.10 manage session/semester module on the administrative panel. As shown in the diagram below, the administrative can add new session and activate new semester. Also, deleting a session would affect all activities of the university ranging from students’ level, result computing and others. Deleting a session should only be done when the session has not been used.

Figure 3.10 Manage session/semester module

Figure 3.11 Shows how the administrator can add new courses to the system

Figure 3.11 Manage courses module
Assigning of courses to lecturers is presented in Figure 3.12. Courses assigned to a lecturer may be reassigned if the lecturer is not around.

Figure 3.12 Assign courses to lecturer module

Figure 3.13 shows how to add departmental courses on RMS.

Figure 3.13 Add departmental courses module

Figure 3.14 shows how the administrative can manage deadline for result computing.

Figure 3.14 shows how the administrative can manage deadline for result computing.
Figure 3.14 Manage deadline module

Figure 3.15 shows how the administrator generates registration code for users of the system. The system users include Administrator, exams officer, lecturers and students. Users are to enter the registration code as the pre-condition for registering on the portal.

Figure 3.15 Registration code module

Figure 3.16 shows Record students’ result module. From the figure, the deadline for computing result is shown on the lecturer’s dashboard, list of departments offering his/her course. The lecturer will click on each department to view the list of students’ who have registered for the course. The record result button is also shown where lecturers can click and record students’ result.

Figure 3.16 Record student’s result module 1
According to Figure 3.16, a lecturer has clicks on the “record students’ result” button shown in Figure 3.16 and he is taken to the page found in Figure 3.17 where he can record students’ result.

Upon clicking on the “Record” button of Figure 3.17, a pop up window seen in Figure 3.18 will appear and the lecturer will be able to record the student test/exams score and the site will compute the total score and grade. However, if the student did not take the test, he will fail the course even when exams scores have been recorded because it is an incomplete result.

Figure 3.19 shows the lecturers dashboard displaying all the courses assigned to the lecturer.
Figure 3.19 Lecturer’s dashboard

Figure 3.20 shows manage course module on the lecturer’s panel. Here, the lecturer can assign himself the courses he is coordinating.

Figure 3.20 Manage courses module on the lecturer’s panel

Figure 3.21 shows a page where the lecturer can view student result. If there is any error on the student result, the lecturer can edit and correct the error.

Figure 3.21 Lecturers view result page
Figure 3.22 shows Exams officer’s dash board where he can view and print students’ semester and session result. Also the exams officer can also perform lecturer’s role by clicking on ‘perform lecturer’s role on the exams officer’s panel.

Figure 3.22 Exams officer dashboard

Figure 3.23 is exams officer’s module showing result master sheet.

Figure 3.23 Session result module

Figure 3.24 shows exams officer generating student transcript.

Figure 3.24 Transcript module
4 Conclusion
In this paper, development of students’ result management system (RMS) is presented. The software is meant to address the challenges facing students’ records management in the University of Uyo. The RMS is developed using PHP, MYSQL, HTML, CSS, JAVASCRIPT and was hosted locally with Apache web server. Also, a Participatory Incremental Process Model (PIP Model) is used as the software development methodology. Functional decomposition of the system and its key modules are provided to explain the major functionalities proffered by the system. Also, use case diagram is presented to show the different categories of the system users and the various functionalities associated the different system user.

References


design, and agile development. In *Companion Proceedings of the 36th International Conference on Software Engineering* (pp. 520-523). ACM.


Copyright © 2017 Akpasam Joseph Ekanem, Simeon Ozuomba, Afolayan J. Jimoh. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.