On-line Programming Course Registration System (OPCRS)

A thesis submitted to the College of Arts and Sciences in Partial in
Fulfillment of the requirement for the degree Master of Science
(Information Technology)
Universiti Utara Malaysia
October 2009

By

Saleh Sulaiman Al-qatan

PERMISSION TO USE

In presenting this thesis of the requirements for a Master of Science in Information Technology (MSc. IT) from Universiti Utara Malaysia, I agree that the University library may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by my supervisor or in their absence, by the Dean of Graduate School. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Request for permission to copy or make other use of materials in this thesis, in whole or in part, should be addressed to:

Dean of Graduate School
Universiti Utara Malaysia
06010 Sintok
Kedah Darul Aman

Abstract

Recently, we noted that the students in the College of Arts and Sciences (CAS) in University Utara Malaysia (UUM) are going into specializations that do not require programming skills to avoid the necessity of learning programming languages as they do not have backgrounds in programming languages such as ICT students. In this proposal we propose a prototype using Java server Pages to serve as a mediator between students and the training center which is located in FTM (Computing professional Enrichment & Development Division (CoPEN) Center). The main objective of this prototype is to improve interaction between students and the Center by providing them with the latest information about the center such as; allowing the students to know if there are any new courses they could start in the near future by sending E-mails(electronic mail) from the system to inform the student about these courses. On the other hand, this system could allow the students to register and pay for these courses online which will improve and facilitate the process of registration and payment. Moreover lecturers could use this system to inform students if he is not able to attend a certain class or if he wishes to change the timetable and the lecturers will able to upload files such as PDF, document to his students Online. By using this prototype, it will increase efficient interaction between students and staff.

ACKNOWLEDGEMENT

Praise and gratitude to Allah, the Almighty, for bestowing me with great strength, patience, and courage in completing this project.

My gratefulness to my supportive and helpful supervisor, Ms. Rafidah Bt Abd Razak for assisting and guiding me in the completion of this research. With all truthfulness, without him, the project would not have been a complete one. He has always been my source of motivation and guidance. I am truly grateful for him continual support and cooperation in assisting me all the way through the semester. I am grateful to my brother (Dr.Mamduh sulaiman Alqatan), Dr.Mohammad Al-muiet, Bilal Al-khateeb for their help in making my project successful.

I would like to present my thanks to my father, my mother and all my family who has always been there for me. Finally, I would like to express my appreciations to all my friends, colleagues, FTM staff, and everyone who has helped me in this journey.

CHAPTER 1

INTRODUCTION

CHAPTER TWO

LITERATURE REVIEW

CHAPTER THREE

RESEARCH METHODOLOGY

CHAPTER FOUR

ANALYSIS AND DESIGN

CHAPTER FIVE

EVALUATION

CHAPTER SIX

CONCLUSION

LIST OF FIGURE

LIST OF TABLE

CHAPTER ONE

INTRODUCTION

1.0 INTRODUCTION

Computer programming still remain an important part of most Information Systems courses (McCarth, 2008). It is not easy for academic institutions to provide relevant and engaging computing courses that students need to expand their life skills and enhance their employment opportunities. Education institutions that have that benefit provide students with a rich and resonant learning experience, using not only the best available textbooks, but also e-learning and software so that students garner hands-on experience. In various fields, there are so many specialized institutions. They should follow the guidelines and specific plans to learning programming language (Holden & Weeden, 2003).

There are institutes of specialized programming languages such as Java, ASP.NET and Oracle. It is aiming to improve the software experience and the level of student performance of software (Hyland & Clynch, 2002). These institutes came to assist students who face difficulties in learning programming languages.

In recent years, there is a clear development of information technology including the development of the universities and educational institutions that contribute to solve many difficulties that are facing students in their universities like online courses registration system which if done could save a lot of precious time (Tunçkan, 2007) .

In university Utara Malaysia, there are many educational institutes such as Computing Professional Enrichment & Development Division (CoPED) center and Job Key center. There is also a newly established Institute specializing in programming languages at the Faculty of Information Technology, Computing professional Enrichment & Development Division (CoPED). This center gives many programming courses such as Cisco, Oracle, Open source, Sun, Microsoft, soft skill and seed software, but the Institute lacks its own website. The process of registration, payment etc are still done manually which sometimes leads human errors. In addition, the institute still announces the available programming courses manually through posters which sometimes lead to a bridge in communication gap as many students who desire to take the courses are not even aware of the availability of such courses.

The online programming courses Registration System is aiming to make it easier and more flexible for students in the College of Arts and Sciences (CAS) to register for programming courses and payment for such courses. Many websites are designed to take care of the online registration. These sites are technologically advanced to facilitate and simplify procedures and improve the level of online registration.

1.1 PROBLEM STATEMENT

The center professional Enrichment & Development Division (CoPED) faces challenge to provide effective interaction with students. The center provides specific programming courses, but to begin these courses we need a certain number of students. In most cases, the courses were cancelled due to the low number of registered students as against the number required for the course. Although, there are many students intending to register for these courses, the communication gap makes it impossible for them as they are not even aware such courses are being taken in the centers.

On the other hand, when some of these course start, the lecturer may be absent on that different occasions without ever informing the students about his absence, so when the students get to the lecture venue and discover there is no lecture on that particular day, they consider as time and effort wasting and may not even turn up for the next class.

According to Gunawardana et al (2008). Filling applications by manual method has become obsolete since we are in the computer age. This also causes a lot of wasted time and effort. The online method saves time and effort.

An interview with staffs of Computing Professional Enrichment & Development Division (CoPED) Center staff –Mohd Samsu Bin Sajat (Chief Coordinator) and Hairuhniza Biniti Abd Rahman (Program Manager).

We found that, there is no effective interaction between students and CoPEN center. There were difficulties in making announcements for any new programming course and in registering the students for the courses. In the staff opinion that the a system will enhance the workings of the centers by helping to inform the students – class members- if the lecturer will be absent or if he wanted to change class timetable, increase the number of registered students in the courses, improve the process of the courses registration, enhance the performance of the staff, enhance the interaction between the center and the students and will make it easier for students to make payment for the programming courses as it would be done online thereby saving time and energy.

1.2 RESEARCH QUESTIONS

The research questions of this study are:

- How to improve the interaction between the students in the College of Arts and Sciences (CAS) in UUM and the professional Enrichment & Development Division (CoPED) Center ?
- What are the basic requirements to build a online registration system?
- How to propose an online Programming Courses Registration System using Server Page (JSP). ?
- How would user accept the new system?

1.3 OBJECTIVES

The main objective of this study is to improve the interaction between the students of FTM College in UUM and the professional Enrichment & Development Division (CoPED) Center. In order to achieve that the following objectives will be pursued:

- To identify the basic requirements for building an online registration system.
- To propose model of online registration system by using Java Server Page (JSP).
- To evaluate the user acceptance of the prototype.

1.4 SCOPE OF STUDY

The scope of this study is concentrating over programming languages institutes, specifically in College of Arts and science, represented by Computing professional Enrichment & Development (CoPED) center. The center provides many courses in programming languages, such as CISCO, Oracle, Open source, Sun, Microsoft, soft skill and seed software, providing details about subtractive courses for the trainee. The students in college of arts and sciences (CAS) will be having the opportunities to register in available programming courses and pay the fees for those courses online.

1.5 SIGNIFICANCE OF STUDY

This prototype will improve interaction between students and computing professional Enrichment & Development Division (CoPED) center. Through this system, students can know new and available programming courses in the Computing professional Enrichment & Development Division (CoPED) Center, they can also register for the courses and pay fees online. Students are encouraged to get details on these courses through a visit to the website of the center. The system will also send E-mail to all group members if the lecturer will be absent or if there is a change in time-table, thus reducing effort and saving time. The lecturers will also able to upload files such as PDF, document to students online.

1.6 Report Organization

The structure of this study is divided in to six chapters and three appendixes as follows: In Chapter one, we discussed based on topic line the question of research, problem statement, research objectives and scope and significant of the study. Chapter Two, we presented a theoretical framework to the online programming registration as well as discussion of the literature relevant to this study. Chapter Three, we described and discussed methods and techniques which will implement in order achieve the objectives that are used in the four or five chapters are focused on. Chapter Four, we provided discussion of the requirements of system and describe the design for the system which includes use case diagram, sequence diagrams and snapshot of the system. Chapter Five, we provided the proposed system discussion and evaluation using the questionnaire to measure the user acceptance. Chapter Six, we described the research Conclusions and Future Work.

CHAPTER TWO

LITERATURE REVIEW

2.0 Introduction

This chapter presents the literature review on the area of project studied. It conceptually gives an insight or reviews on the previous and existing works that have been conducted on the same area. This chapter is organized into four sections of subtopics. The first subtopic reviews on the web-based technology. Meanwhile, the second subtopic will shed light on the online registration and booking systems. Thereafter, the third subtopic will focus on the Internet payment systems and the tools for optimizing these services. The fourth subtopic reviews on Java Server Page (JSP), MySQL and the related and previous works.

2.1 Web Applications

A Web application is a software application that delivers its functionality to a user from a Web server, through a network such as the World Wide Web or an intranet. The user views and manipulates the application through a Web browser (Carat, 2002). There are too many application used in many fields, like education, e-commerce, and other web-based facilities (Nijaz, 2000).

2.2 Webs-Based and Communication

According to Kerner (2006), the new rapid the integration in various fields, especially in the creation of modern and demanding study material becomes more and more important. Many studies tried to explore the alternative methods by using the interactive tools in the material, user-friendly, and available on the World Wide Web (Schmitt, 2006). The main aim for enhancing the Web services is to build some general guidelines for the design and development of interactive Web-based that make it friendly and flexible for the users.

2.3 Advantages of Web-based Applications

The progress in the business fields these days became the real changing for the trade methods, the Web-based occupy a big area in the growth. Otherwise the growth of the internet technology makes it more flexible way for the business that can be available for the small business education and organization (Ahn et al, 2004).

Some are referring to the new Internet evolution, which refers to organizations new approach to using the Internet to increase efficiency, productivity and customer satisfaction, while at the same time, substantially decreasing administrative and personnel costs. There are many other advantages for the web-based application:

- Web-based applications are custom designed to meet the specific need
- Web-based applications provide 24/7 access from any web-browser in the world to the data stored in the custom system.
- Web-based applications are secure, fast, and extremely reliable.

2.4 Principled Design of the Web Architecture

The WWW has succeeded in a big part because its software architecture has been designed to meet the needs of an Internet-scale distributed hypermedia application. The advantage of the Web-based becomes more useful for many fields, the Web-based architecture emphasizes scalability of component interactions, generality of interfaces, which used to reduce interaction latency, the integration of the security, and encapsulate legacy systems (Chan, 2008).

The general architecture of Web-based applications relies on three-tier architecture (Figure 2.1): The client, the web server and the database. These general architectures are applicable across technologies (e.g. Microsoft and Java). There are two types of web architectures:

- Architectures suitable for larger Enterprise Applications.
- Architectures that might be used for smaller Web Applications.

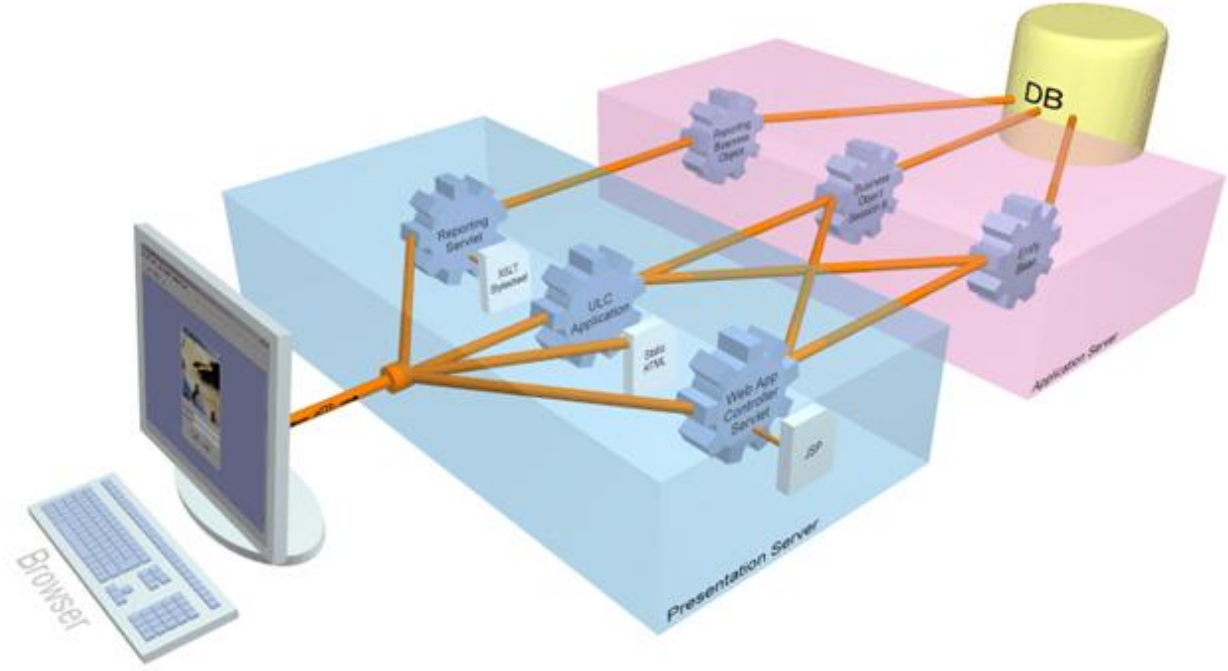


Figure 2.1: Web Based Architecture (Chan, 2008)

2.5 On-line Registration and Booking System

On-line registration is an example of Web-based applications is custom designed to meet the specific need**.** The different facility for providing users with the booking or registration system is easy to use web-based system designed especially for an organization which requires facility management (eg. seminar rooms, recreation rooms, sports facilities). Online registration system is a complete solution: hardware and software are integrated seamlessly resulting in great convenience for the user as well as flexibility for the system manager. One significant advantage of it is the modular design which meets the organization needs can be easily customized (Treiber, M., 2007).

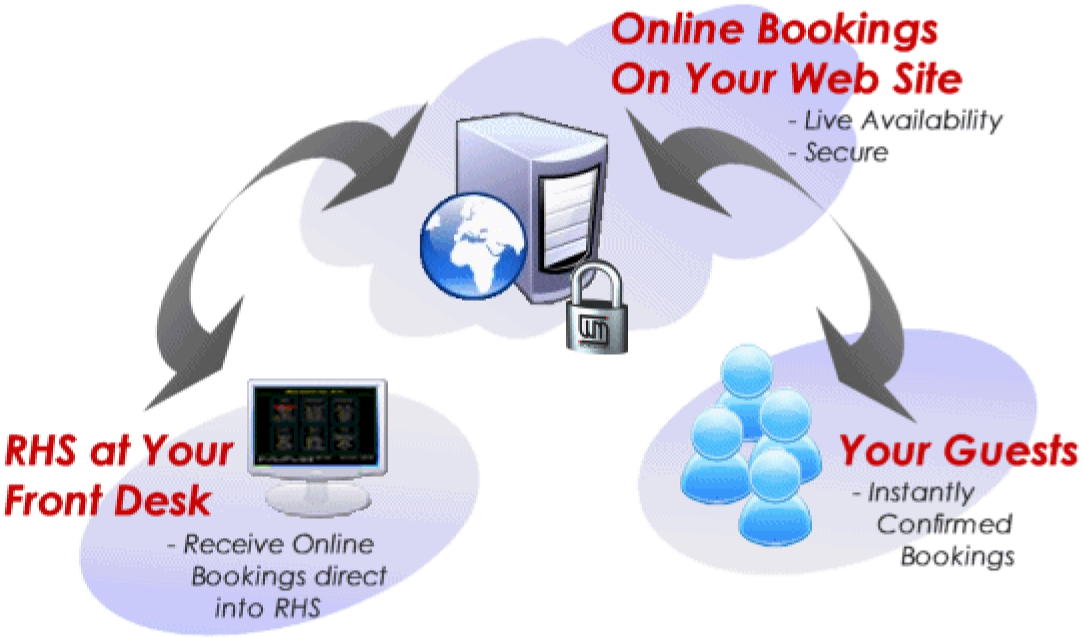


Figure 2.2: Online Booking System Architecture (Treiber, M., 2007)

On-line registration is considered as a contract just like other types of traditional registration. The idea that a registration or booking of this kind is an agreement has long been established. While electronic registration are of fairly recent vintage, the same principles apply as those of, say, telephone booking.

Figure 2.3 demonstrated how online registration system help registered campus users who need anytime-access to documents where electronic reserves comprise digital files, mostly H-TML formats. Even more resources placed in electronic reserves are convenient and relatively accessible for researchers: students and other users. A lot of people wish electronic booking or registration facilities to be deployed because they can easily login to obtain information from work, university , or any place on a 24/7 (Lankton, 2007).

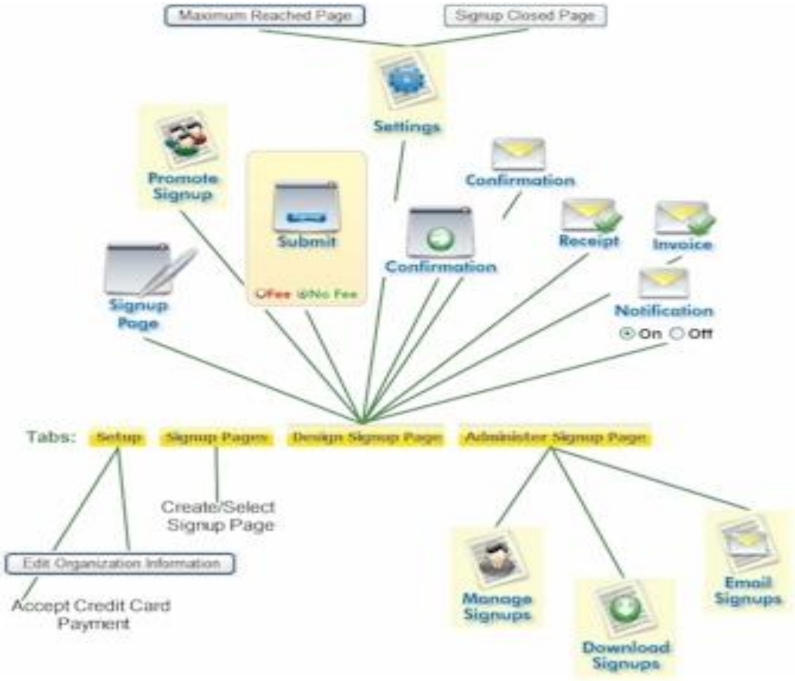


Figure 2.3: Online Booking System (Lankton, 2007).

According to Ciebiera, Mincer-Daszkiewicz and Walen (2004), registration is may be the most used and important service of training centers. The service that every student will use when they want register in centers. So, designing an effective registration system deserves significant attention performing study. As with online application, online registration is becoming very common in institutions because of the available technologies. The registration operations should be detailed information and very easy for students to use.

According to Boroson (2003), Course Registration System is a web-based program aimed to make easier and more convenient the registration process. Course Registration System (CRS) attempts to alleviate these hassles by providing several services to students through the internet. Most of the aspects in online registration systems are verified automatically and therefore could reduce human mistakes.

2.5.1 Benefits of Course Registration System

The benefits of course registration system are as the following:

(1) Easy to access and get information, (2) Information is stored securely and privacy (Blowers & Bryan, 2004), (3) 24 hour access to information and using it, (4) provide fast and effective registration process and (5) saving of money and saving of time (Srivastava, 2008).

2.5.2 Internet Payment Systems

According to Jing (2009), in many respects, online payment is the foundation of systems for electronic commerce. The ability to take payment distinguishes an electronic commerce system from one that provides only advertising or other communication capabilities. This system will use online payment system as a important part of it. To make the payment process is easy.

Electronic payment system, and the ability to pay electronically for goods and services via the Internet, is an integral part of e-business models and infrastructure for electronic commerce. One of the main reasons for the spread of e-commerce transactions, and perhaps the rapid development and growth in various electronic payment systems. In developed countries, and credit cards had been used before the advent of the Internet. (Singh Sumanjeet, 2009).

A key component of doing business is to accept the payment. Electronic commerce means that the virtual machine. As such, the electronic device is prohibited and encourages the use of cash transactions which do not require tangible goods, but rather include data transfer only. (Rayport and Jaworski, 2004).

Internet payment systems refer to the various methods by which individuals and companies doing business online collect money from their customers in exchange for the goods and services they provide (Junping, 2007).

According to Lee and Jae (2008) the online payment options are (Figure 2.4):

- **Credit cards**, credit cards remained the most common means of online payment. They were also among the easiest payment methods, for consumers as well as for merchants.

- **Smart cards**, smart cards are similar to credit cards, except they store information on an imbedded chip instead of on a magnetic strip on the back. Consumers can load money into an account on the card by using an Automatic Teller Machine (ATM) or by placing the card in a slot in a specially equipped computer.
- **Digital cash**, digital cash is a form of electronic currency that functions similarly to a debit card. Customers can transfer money from savings and checking accounts into an online cash account, from which they withdraw to make purchases over the Internet.
- **Electronic checks**, small businesses can also allow customers to pay for online purchases by accepting personal or business checks online.

- **Secure third parties and online banks**, secure third parties may be banks or other institutions that act as middlemen in financial transactions between merchants and customers. For small businesses, conducting transactions through a secure third party eliminates the need and expense of setting up a secure Web site. Secure third parties also provide consumers with added protection from fraud, since the merchants never handle their credit card numbers.

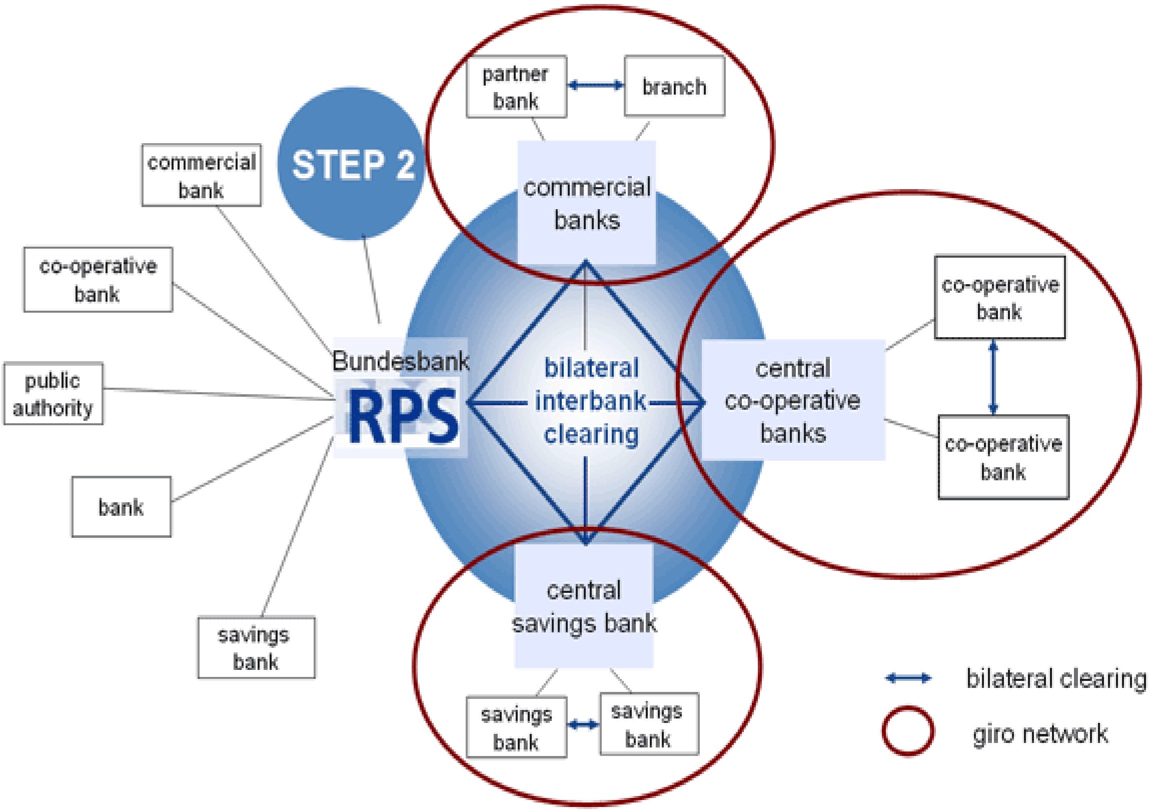


Figure 2.4: Online Payment System (Lee & Jae, 2008)

2.6 JAVA SERVER PAGE (JSP)

For this project JSP language has been used. Java Server Pages (JSP) is a technology that combines HTML / XML markup languages and elements of the Java Programming Language to return dynamic content to the Web client, which is normally used Presentation to face the logic of the application on the Internet, although it may be business logic (Sun,2007).

Java Server Pages (JSP) technology provides a quick and easy way to create Web pages that display dynamically-generated content. JSP technology is designed to make it easier and faster to build web-based applications that work with a wide range of Web servers, application servers, browsers and tools development. (Sun.2008)

2.6.1 Java Server Pages technology work

JSP uses XML tags and allows scripts written in the Java programming language encapsulate the logic that produces the content. Through the format (HTML or XML) tags directly on the feedback page. Thus, JSP pages separate the logic from page design and posting.

JSP technology is part of the Java technology family. JSP, servlets compiled in JavaBeans components (beans), or project components, JavaBeans (enterprise beans) to be processed on the server. Thus, this is JSP technology is a key element in the structure quality levels scalabilities for Internet applications. JSPs are with unlimited on certain platforms or web server. JSP specification represents many inputs industry (Sun technology,2008).

2.7 DATA BASE

The simplest definition of a database is a collection of data items, typically describing the activities of one or more related organization, these items stored for later retrieval (Ramakrishnan and Gehrke, 2003).

Computer database is a structured set of documents or data that is stored in the computer system. Reliable data base program to organize storage data. In other words, the software models and the basic structure of data in what is called the models in the database (or data models). Form being the most common use today is the relational pattern. Other models such as the hierarchical model and the model using the network more explicit representation of relationships (Ramakrishnan and Gehrke, 2003).

2.7.1 MySQL

The type of database that has been chosen for development of this project is MySQL. MySQL is a multithreaded, multi-user SQL. The program runs as a server provides users access to a certain number of databases (JamesTurner,2002).

2.7.2 Advantages and Strength

MySQL have some of advantage competitors with other types of database as are PostgreSQl, Microsoft’s Server, and Oracle. The following advantages for MySQL:

(1) **High performance: -** There is no doubt promptly. You can see that developers Page mysql.com standard site on the Internet. Many of these standards and show MySQL to be orders of magnitude faster than the competition.

(2) **Low cost: -** MySQL Is available free under an Open Source license, or Reducing the cost under a commercial license, if required for your application.

(3) **Ease of use: -** most versions of database use SQL. As a new beginner, I did not have much trouble in using MySQL as it is rather directly forward.

2.8 PREVIOUS AND RELATED WORKS

Different studies have been addressed and reported the importance of online services for deploying other facilities.

According to Cui (2005), discussed the usefulness of using a web-based academic departmental community model is proposed to assist in the strengthening of departmental identity and community as well as advancing its mission. Concepts such as 'wiki' and anthill community underpin this model.

According to Naini (2008) reported the weakness in the registration process for graduate students in the Electrical and Computer Engineering Department is a paper-based process. The Masters of Science in Electrical Engineering (M.S.E.E) course plan is available as a downloadable Portable

Document Format (PDF) and Microsoft Word document at the Electrical and Computer Engineering Web site. The graduate students have to fill this course plan document and send it to their advisor as an email attachment for the advisor’s approval. The advisor then approves and signs the course plan. To proceed further with the registration process, the student should submit the approved document to the department.

According to Naini (2008), discovered the usefulness of using the WISRAS for simplifying the registration process of the graduate students by providing an electronic and interactive registration process. In the WISRAS process, the student fills the course plan page on the intranet site of the department and submits it online, which generates an email confirmation of the course plan submission to both the student and the advisor. The advisor then checks the course plan of the student on the intranet site and electronically approves it, generating an email confirmation of approval to the student and the department.

According to Yasuhiko (2006), which highlighted some problems regarding the student registration process, such as the difficulty of performing registration accurately due to the extremely complex rules in a registration system? In order to make such determinations with a high degree of precision, databases with a variety of information and programs that precisely describe the rules are needed.

A study by Ahmad and Yusoff (2001) reported the needs for a better integrated process design in higher education institutions that implemented e-learning to incorporate learner's expectation into the desired process, and propose a basic design of the courses registration process that could be integrated on-line. Also he recommended that the integration of productive processes from registration process to learning process should be created to ensure learners' expectations are consistently met.

According to Zainal & Hasibuan (2005), the Faculty of Computer Science (Fasilkom), University of Indonesia (UI) develops Student Centered E-Learning Environment (SCELE) for Graduate Program in Information Technology, which is developed using Enterprise Resources Planning approach as Learning Management System (LMS). Beside SCELE, Fasilkom UI also develops Contents that are conformant with SCELE, and also Digital Library and Online Academic Registration System to support learning process.

In The table 2.1 features and topics of some related works:

Table 2.1: Related Works

Author	Year	Topic	Features
Jing	2009	On-line Payment and Security of E-commerce	To increase more flexible and easier in payment process.
Gunawardana et al.	2008	An Online Course Registration System for the Faculty of Engineering in University of Peradeniya	To develop an electronic course registration system, where most of the aspects are verified automatically and therefore could reduce human mistakes.
John Jenq et al.	2004	Online Interactive Home Work Grading System	To facilitate the student and faculties in submitting and grading their home works, a new, interactive and adaptable approach has been implemented, and to allow the grader to mark over the assignment submitted and to add comments to it. To show how Web technology can be used to develop a software system for automatic management of assignments as well as paperless grading.
Zane	2004	Wireless Student Testing , Proceeding of the International Conference on Pervasive Computer and Communication	To improve registration process, where students enable on register from anywhere and anytime when they can access internet.

2.9 SUMMARY

The second chapter discusses the literature review of several related papers and applications that discuss the benefits of the internet, web services, and the information communication technology in the online services. Additionally, this chapter also identified the payment facilities types that web services provide. Finally, this chapter discussed the issues that faced the previous institutes, and the suggestions that may help to develop and improve the online course registration systems.

CHAPTER THREE
METHODOLOGY

3.0 Introduction

This chapter elaborates the study methodology which was adapted in this study. The research methodology in this study is a v method, a good selection, described and accepted in many researchers in the field of information, system design research (Vaishnavi & Lemoine, 2006).

3.1 Research Design

This research was done in several steps. Figure 3.1 below shows the main stages in the research, design methodology.

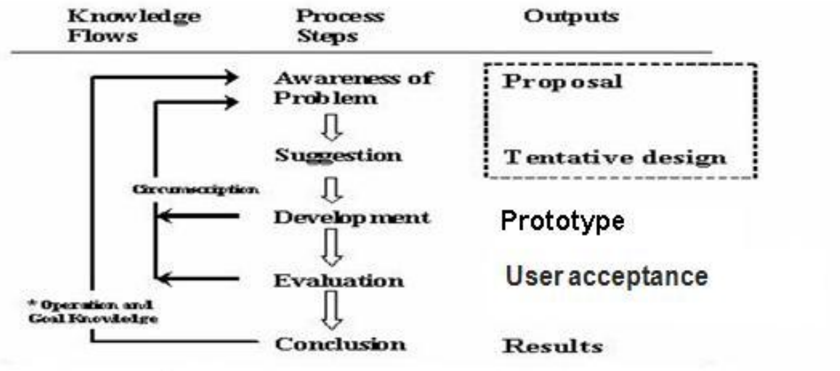


FIGURE 3.1: General Methodology of design research (adapted from Vaishnavi and Kuechler 2004)

According to Vaishnavi and Kuechler (2004), the design study methodology contained the major steps: Awareness of the Problem, Suggestion, Development, Evaluation and conclusion as shown in the previous figure.

3.1.1 AWARENESS OF PROBLEM

According to the information that has been gathered, the students and staff of (CoPED) center in college of Arts and Sciences(CAS) are facing difficulties with announcements about new programming coursers, registration and payment process. This phase of the methodology is in general, by a series of specific analysis and discussions with management and business users. It examines the development of procedures for the establishment of a mutual understanding of the objective, scope, user requirements and evaluation of the study feasibility. The information has gathered from interviews with chief coordinator and interviews and questionnaires with program manager in the center so as to get the real problem concerning this study.

3.1.2 SUGGESTION

Many approaches to the problem of this study were looked into with the view of selecting the best suggestion or approach for the accomplishment of this task. These are discussed over a period of time. Some of the alternatives that were discussed were development of a new system. Thus, this study, will present a prototype using Java server Page(JSP) to improve the website to Computing Professional Enrichment & Development Division(CoPED) . The center allowing students to register courses and pay fees for this courses and also enable them to know any new courses by sending E-mail. Moreover the lecturer will send E-mail to students in his class when he is absent or changes class time by using this system and he will able to upload files such as PDF, document to his students online . Analysis of the system was by UML diagram by using Use case, sequence diagram and class diagram.

3.1.3. DEVELOPMENT

The third phase of the methodology is concerned with converting the analysis that has been done in the previous phase into implementation, which will end up with the prototype. The completed design is now translated into program code. In this phase, the system was completely developed with Java Server Page (JSP) and database to store and retrieve all information, and phone emulator to show the results. The application process contains three main steps which were adapted from (Laudon & Laudon, 2000), as shown in (fig. 3.2). The purpose that approved by the users can be used as template to create a final application (Laudon & Laudon, 2000).

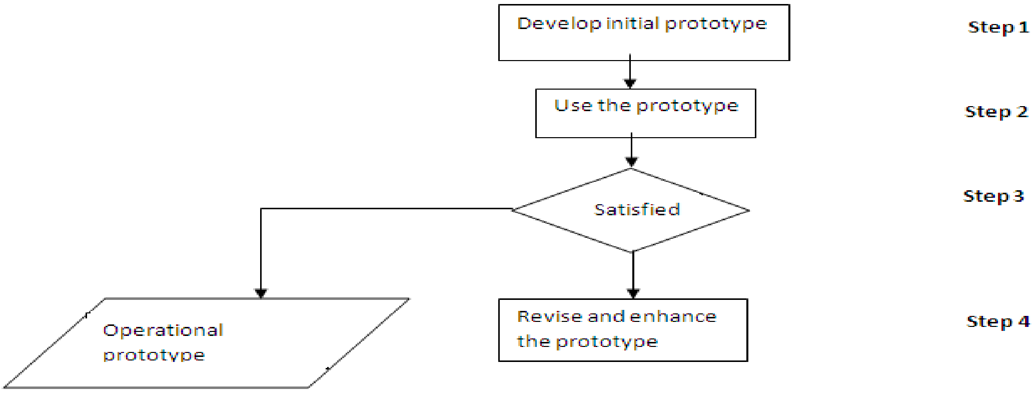


FIGURE 3.2: The Prototype Process Adapted from (Laudon & laudon 2000)

3.1.4. EVALUATION

The evaluation was conducted based on determining the usability and trust ability of the system which will be developed during this work as well as the satisfaction of the users.

Interviews and questionnaires were used to test this work. Students of IT analyzed and evaluated the web site and to find out what are the features of a web site.

3.1.5 CONCLUSION

This chapter discusses on the research methodology. The design has been made and linked with prototype. The project included functionality testing and report writing. Documentation was also prepared. The results for this study were based on the objectives of the study and the evaluation of the prototype by the students and staff.

CHAPTER FOUR

ANALYSIS AND DESIGN

4.0. Introduction

This chapter will introduce more details on the analysis and design of the system and the system development of online programming courses registration system prototype. First, discuss on the system requirements and the design. The system requirement will provide the functional and non-functional requirements of the proposed system. The UML diagram will presents in this chapter by use case diagram, use case specification (In Appendix C) and sequence diagram.

4.1 System Requirements

System requirements are important part to build the proposed system, which will presents in this step to give main components about system. System requirements used to explain the system needs, this step will provide the researcher to gather the research requirements that will be main aim to this research

System requirements consist two main types:

1. The functional requirements
2. The non-functional requirement.

4.1.1 Functional requirements:

The main functional requirements (Table 4.1) that the system provide are:

Table 4.1: Functional Requirements

List of Requirement	Description
Requirement1: Fill Application.	• The system should allow the students to update their information.
Requirement2: Update student information.	• The system should allow the students to update their information.
Requirement3: View Information	• The system should allow the students to view their information.
Requirement4: search course	• The system should allow the students to search about specific programming course.
Requirement5: Register Courses	• The system should allow the students to register programming course that want them.
Requirement6:View My course	• The system should allow the student to view his or her programming course.
Requirement 7: View My Email	• The system should allow the student to view his or her E-mail messages.
Requirement 8: Download File	• The system should allow the students to download his or her programming course files.
Requirement9: pay fees	• The system should allow the students to pay fees for programming courses that registered it.
Requirement10: Login	• The system should be allowed for students, lecturer and administrators to access their accounts through the insert correct user name, password and select user type.
Requirement11: Send Message	• The system should allow the lecturers to send message to his/her group members in class.
Requirement12: Loading files	• The system should allows the lecturers to upload files to his group members such as PDF.
Requirement13: manage courses	• The system should allow the administrator to manage programming courses (insert new programming course or delete course or update course or view course).

Requirement14:manage Advertisement	- The system should allow the administrator to manage Advertisement (insert new Advertisement or delete Advertisement or update Advertisement or view Advertisement).
Requirement15: Approval/ Reject Students.	- The system should allow the administrator to Approval/Reject student's application.
Requirement 16: Manage Student.	- The system should allow the administrator to manage Student (insert new Student or delete Student or update Student information or view Student information).
Requirement 17: Manage Lecturer	- The system should allow the administrator to manage Lecturer (insert new Lecturer or delete Lecturer or update Lecturer information or view Lecturer information).

4.1.2 Non-Functional requirement:

The main non- functional requirements (Table 4.2) that the system provide are:

Table 4.2: Non Functional Requirements

List of Requirement	Description
Requirement 1: Security	- The Students and Lecturers and administrators must have a username and password to log in the system.
Requirement 2: Usability.	- The system must be easy to deal with. - The system must provide easy information for the users. - The system must give easy instruction.
Requirement 3: Understandability	- The system should be easy to understand

Requirement 4: Performance	• The system must have a reasonable speed according to the technology used to access many of students to view courses information, to programming course register and pay fees for programming courses at the same time
Requirement 5: Availability	• The system should be available to all students and lecturers and administrators.

4.1.3 SOFTWARE REQUIREMENT

The software requirements, the system mainly needs these software products is shown in Table 4.3.

Table4.3: software Requirements

List of Requirement	Description
Requirement 1: Operating System	• Microsoft Windows XP Professional
Requirement 2: Database server.	• This component will be used to store the information such programming course information and student's information.
Requirement 3: JSP.	• This is the language used to develop this system.

4.2 Unified Modeling Language (UML)

Unified Modeling Language (UML) is used to draw the necessary diagrams such as use case, sequence. Defines UML as "Unified Modeling Language (UML) is a standardized visual specification language for object modeling. UML is a general purpose modeling language that include a graphical notation used to create an abstract model of a system, referred to as a UML model". UML is a suitable formalism to improve the understanding by both users and developers (Silva & Paton, 2003).

According to David, (2004) UML is industry standardization's graphic notation for specification, visualization, construction and documentation of item for software system. It is simplification of the complex process of software's design by creation of visual's models. Standard UML defines large set of resources, they use for development of products and for this project the following was used:

1. Use case diagram represents:
   - The functionality of system
   - Relationship between actors and system.
2. Sequential diagram or collaboration diagrams represents:
   - The dynamic structure of system.
3. Class diagram and package represent (Appendix E):
   - The static structure of system.
   - Relationship between elements such as interfaces, classes and control.

4.3 Use Case Model

The use case model specifies the functionality the system has to offer from a user's perspective and we define what should take place inside the system. This model uses actors to represent roles the users can play, and use cases to represent what the users should be able to do with the system. Each use case is a complete course of events in the system, seen from a user perspective. If appropriate, interface descriptions may also be developed. These will specify in detail what the user interface will look like when the use cases are performed. To give a conceptual picture and a better understanding of the system, we use objects that represent occurrences in the problem domain.

A use case is a specific way of using the system by performing some parts of the functionality. Each use case constitutes a complete course of events initiated by an actor and it specifies the interaction that takes place between an actor and the system. A use case is thus a special sequence of related transactions performed by an actor and the system in a dialogue. The collected use cases specify all the existing ways of using the system.

Use cases describe the behavior of the system when one of these actors sends one particular stimulus. This behavior is described textually. It describes the nature of the stimulus that triggers the use case; the inputs from and output to other actors, and the behavior that convert the inputs. The text of the use case also usually describes everything that can go wrong during the course of the specified behavior, and what remedial action the system will take.

4.3.1 Actor

Actor is person outside the boundary of the system but this person represent important part of system environment, where the actor can represent other system (the actor do not must to be person only) or other devices such as printer. The actor is person who has interacted with computer system (Satzinger, Jackson and Burd, 2002). Based on the information requirement we have the following Actors:

1. Students.
2. Lecturer
3. Administrator

This is the main use case diagram model the functionality of students, lecturers and administrators as actors of the system. The first actor, the functionality of the students is the ability to interact with the system by view programming courses information, and then the students can register those courses and pay fees it online. The second actor, the functionality of the lecturer is the ability to interact with the prototype through entering correct username and password, and then the lecturer will be able on loading files which relate in his/her class such as PDF .In addition, the lecturer will be able on sending message(E-mail) to all his/her class members. The third actor, the functionality of the administrator is the ability to interact with the system by sending message(E-mail) to all students to inform about any new programming course as well as he will be able to do manage (Insert ,Update ,Delete) for programming courses in Computing professional Enrichment & Development Division(CoPEN) Center.

ONLINE PROGRAMMING COURSES REGISTRATION SYTEM

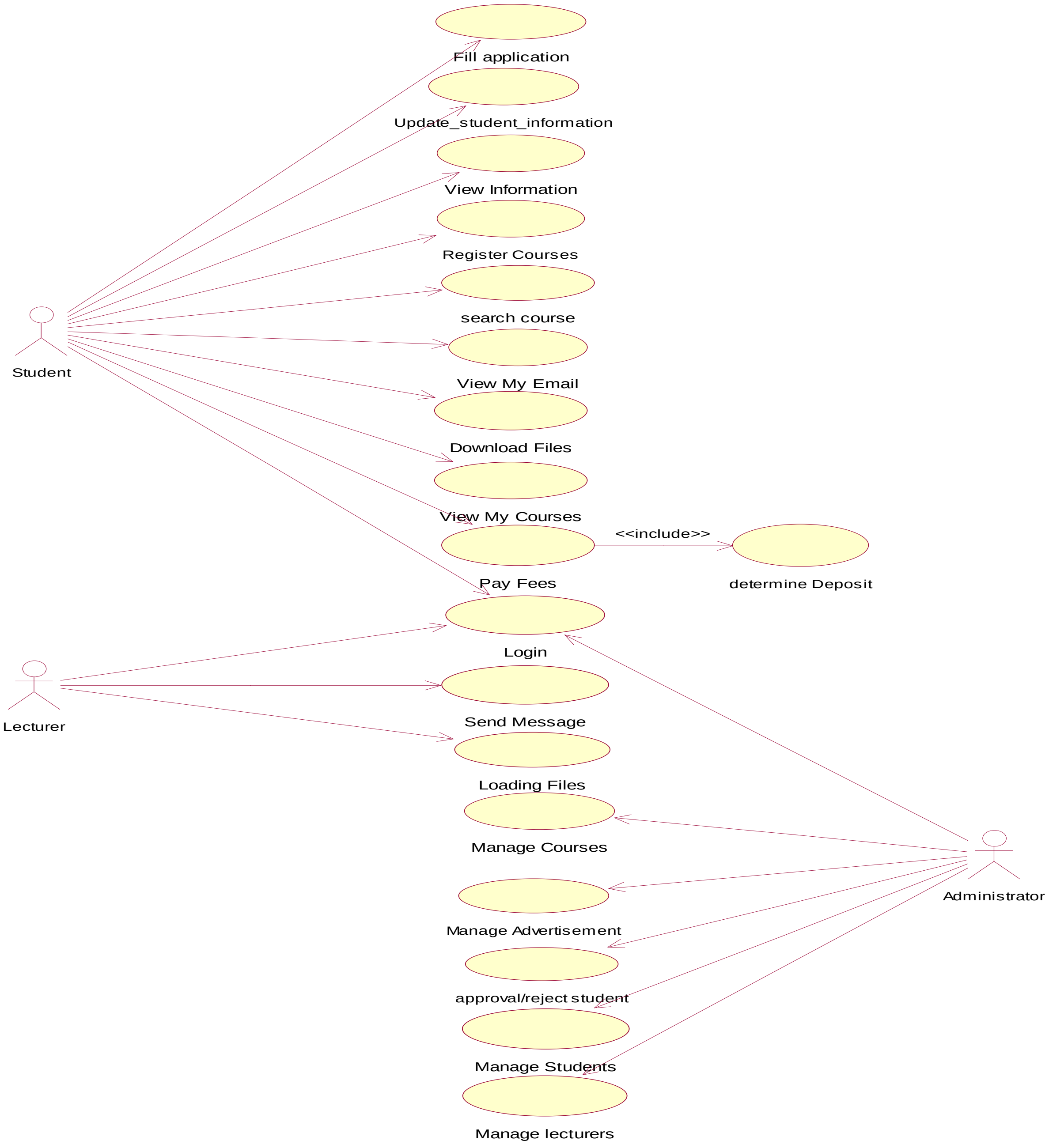


Figure 4.1: Main Use Case

4.4.SEQUENCE DIAGRAM

In this step of UML diagram, that shows the processes that execute in sequence, the sequence diagram shows the sequence of message, which are exchanged among roles that implement the behavior of the system, arranged in time, it shows the flow of control across many object that collaborate in the context of a scenario.

However the sequence diagram (interaction) it captures the behavior of single use case showing the messages passed between those object of the case and describe the sequence of operation in that use case.

4.4.1. Fill Application

The student in this sequence diagram should open student registration page to fill and apply his or her information.

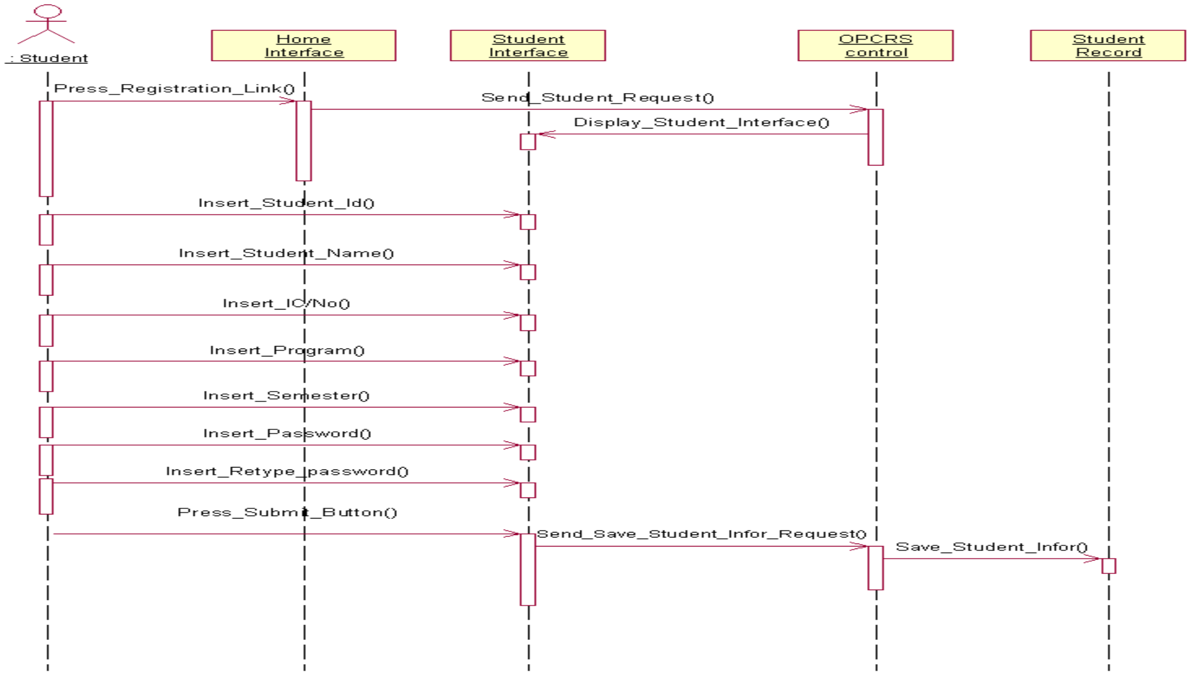


Figure 4.2: Fill Application

4.4.2. Update Student Information

The student in this sequence diagram should open update student information page to update his or her information.

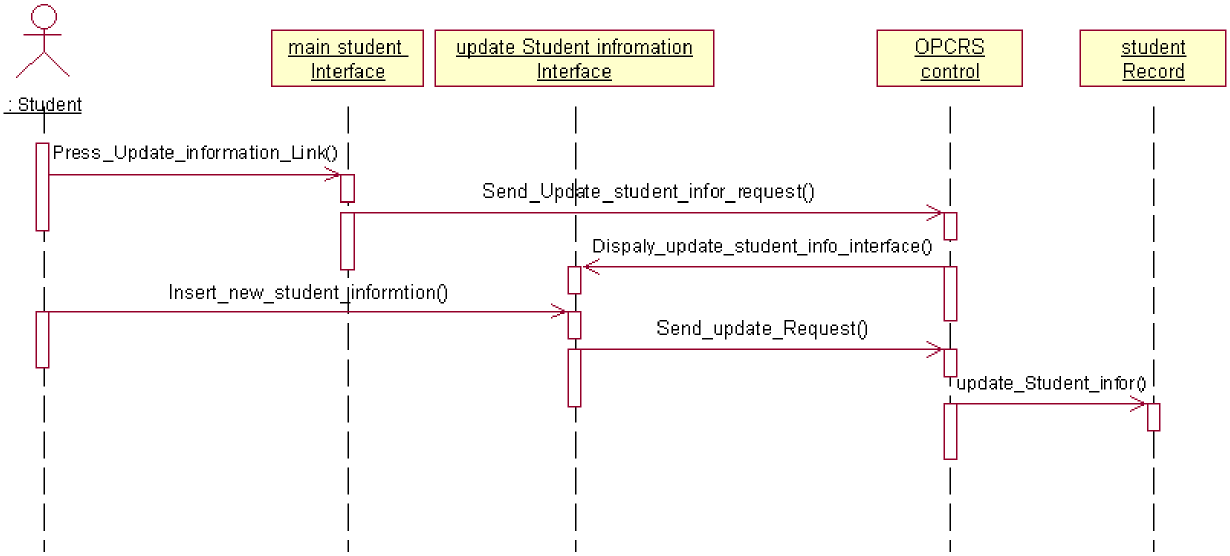

: Student
main student Interface
update Student infromation Interface
OPCRS control
student Record
Press_Update_information_Link()
Send_Update_student_infor_request()
Dispaly_update_student_info_interface()
Insert_new_student_informtion()
Send_update_Request()
update_Student_infor()

Figure 4.3: Update Student Information

4.4.3. View Information

The student in this sequence diagram should open View student Information page to view his or her information.

.

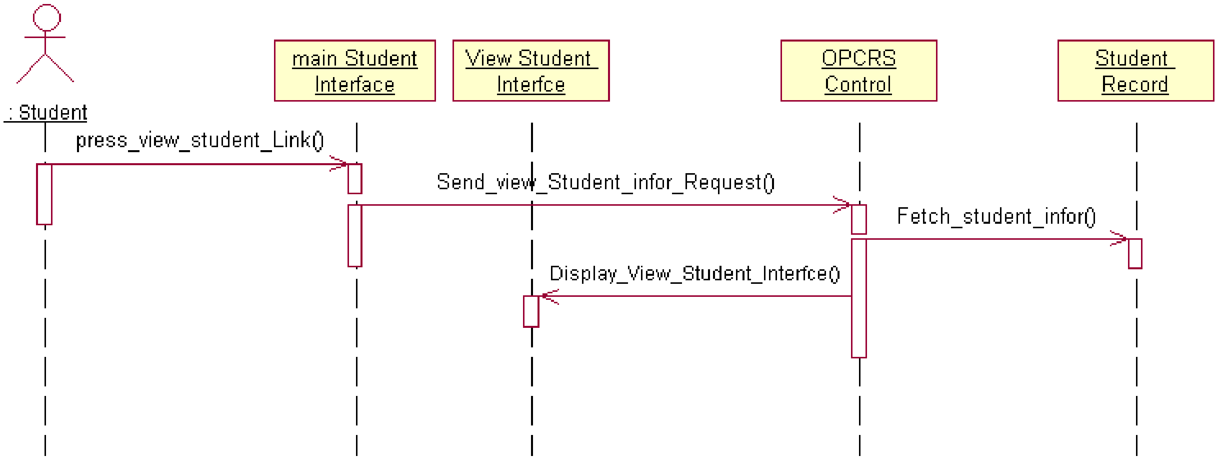


Figure 4.4: View Information

4.4.4. Register Courses

The student in this sequence diagram should open register programming course page to register his or her programming course that he want register it.

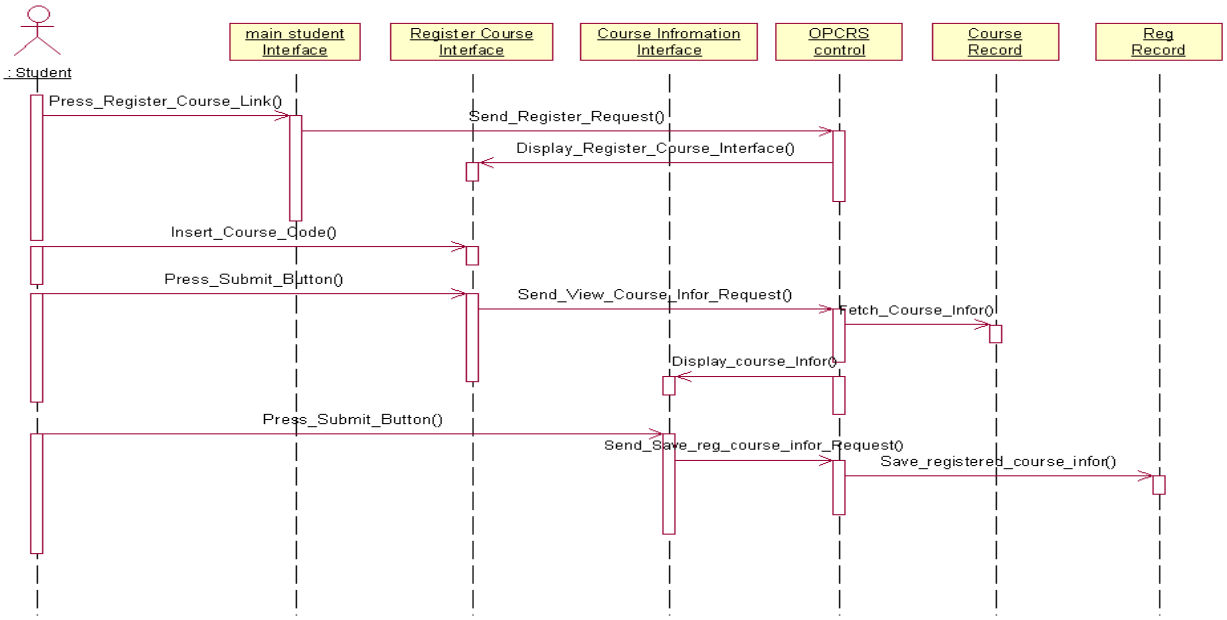

: Student
main student Interface
Register Course Interface
Course Infromation Interface
OPCRS control
Course Record
Reg Record
Press_Register_Course_Link()
Send_Register_Request()
Display_Register_Course_Interface()
Insert_Course_Code()
Press_Submit_Button()
Send_View_Course_Infor_Request()
Fetch_Course_Infor()
Display_course_Infor()
Press_Submit_Button()
Send_Save_reg_course_infor_Request()
Save_registered_course_infor()

Figure 4.5: Register Course

4.4.5. Search Courses

The student in this sequence diagram can open Search Course Page to make search about specific programming course.

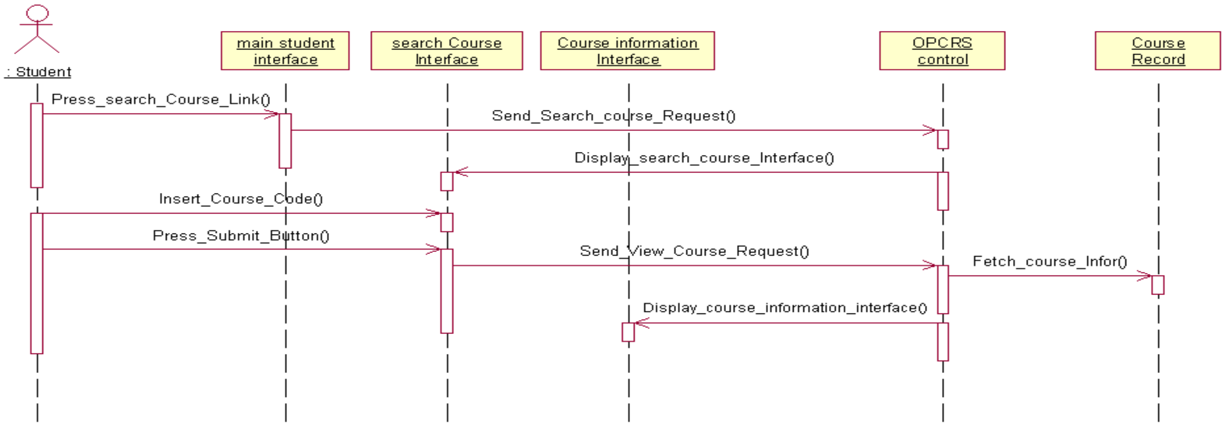

: Student
main student interface
search Course Interface
Course information Interface
OPCRS control
Course Record
Press_search_Course_Link()
Send_Search_course_Request()
Display_search_course_Interface()
Insert_Course_Code()
Press_Submit_Button()
Send_View_Course_Request()
Fetch_course_Infor()
Display_course_information_interface()

Figure 4.6: Search Course

4.4.6. View My Email

The student in this sequence diagram can open View My Email Page to view his or her Email messages.

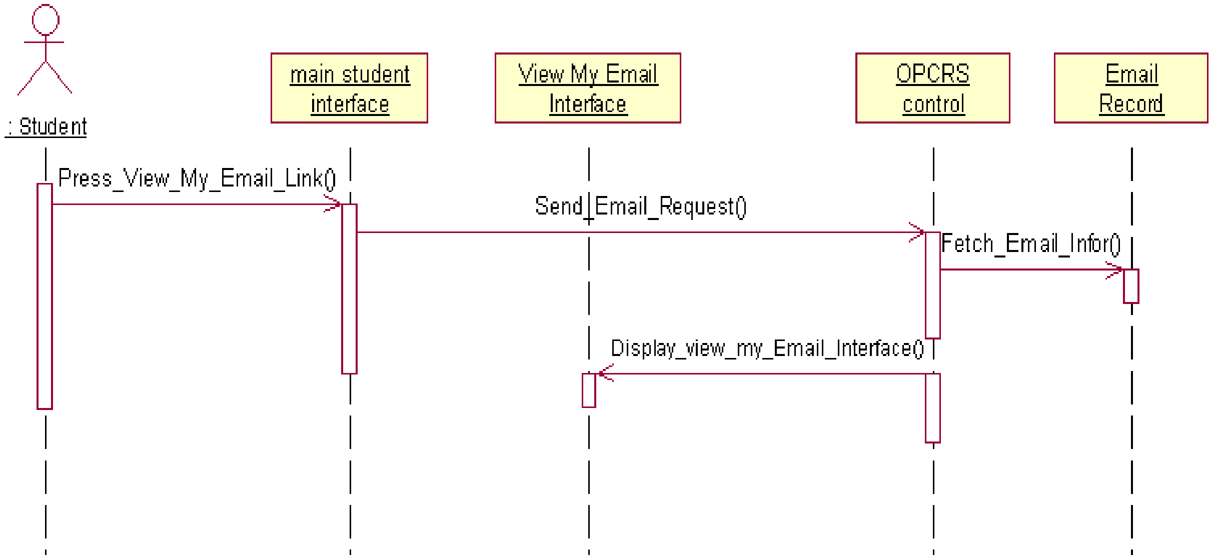


Figure 4.7: View My Email

4.4.7. Download Files

The student in this sequence diagram can open Download file page to download his or her programming courses files.

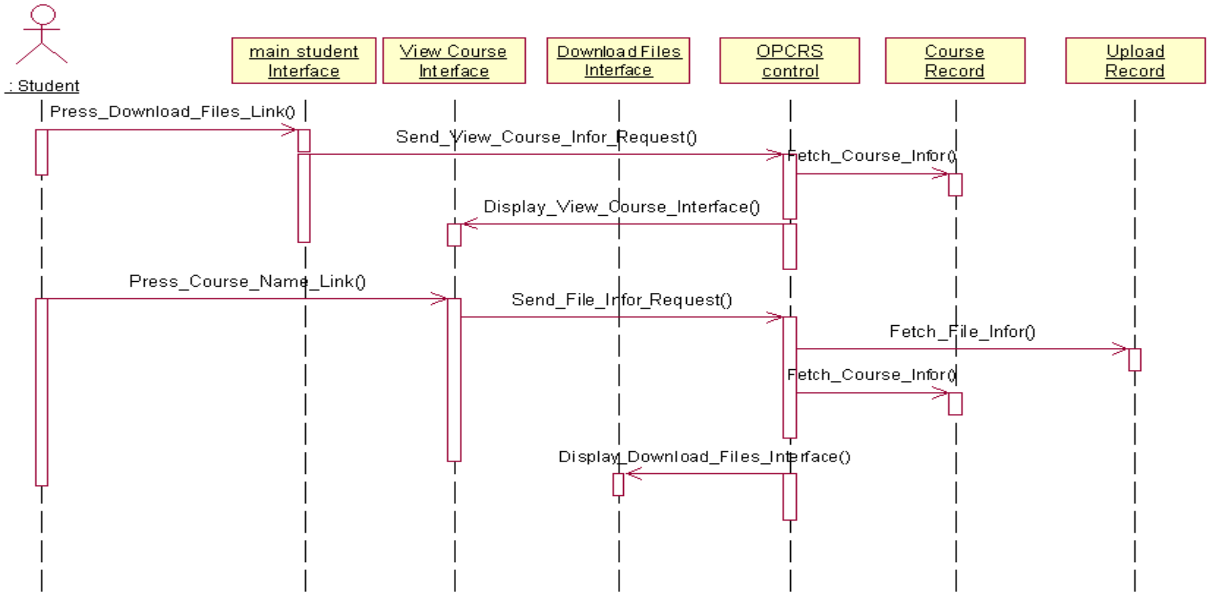


Figure 4.8: Download Files

4.4.8. View My Course

The student in this sequence diagram can open view my course page to view his or her programming courses.

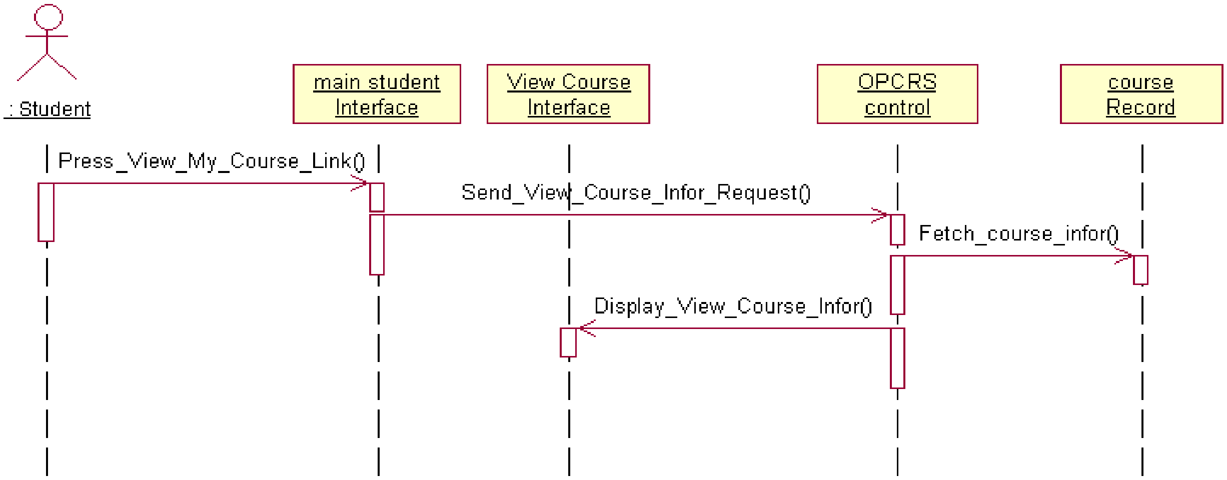


Figure 4.9: View My Course

4.4.9. Pay Fees

The student in this sequence diagram can open payment page to pay his or her programming courses fees to this service by inserting credit number and deposit value.

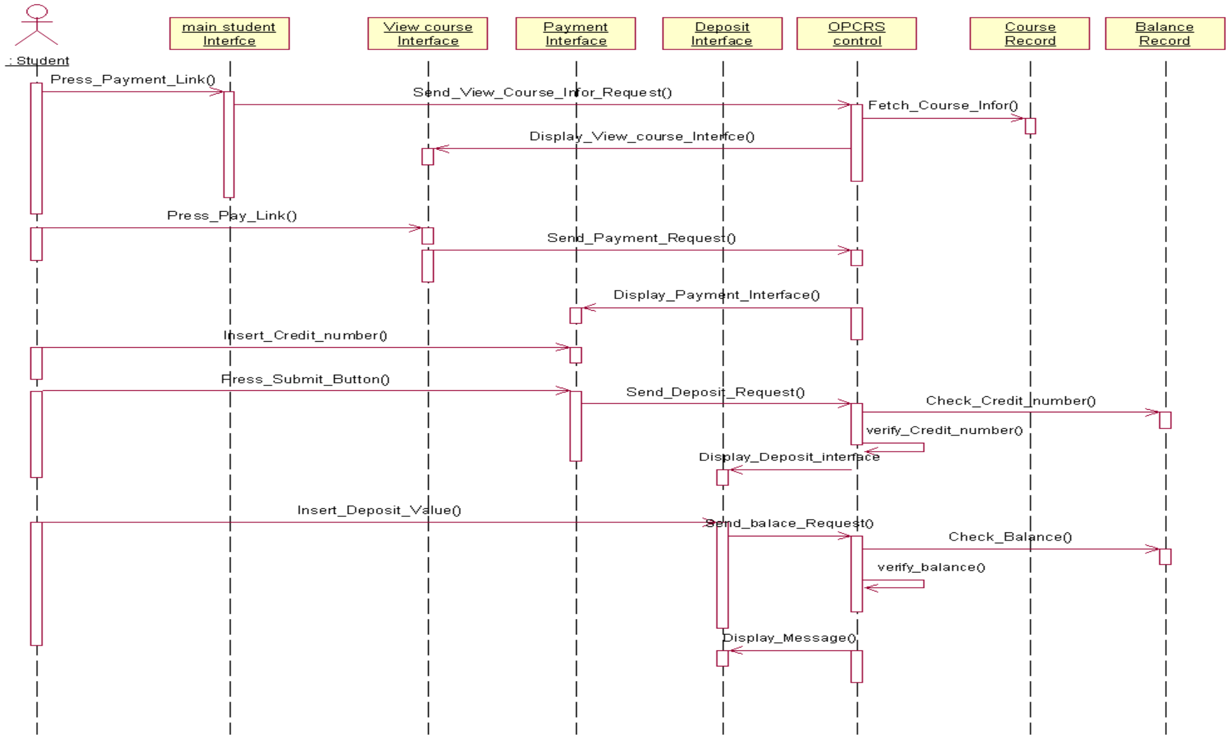


Figure 4.10: Pay Fees

4.4.10. Login

The (student or Lecturer or Administration) in this sequence diagram can open Login page to let User them login to enter his or her main page to this service by inserting Username, Password and Select user type

4.4.10.1. Login for Student

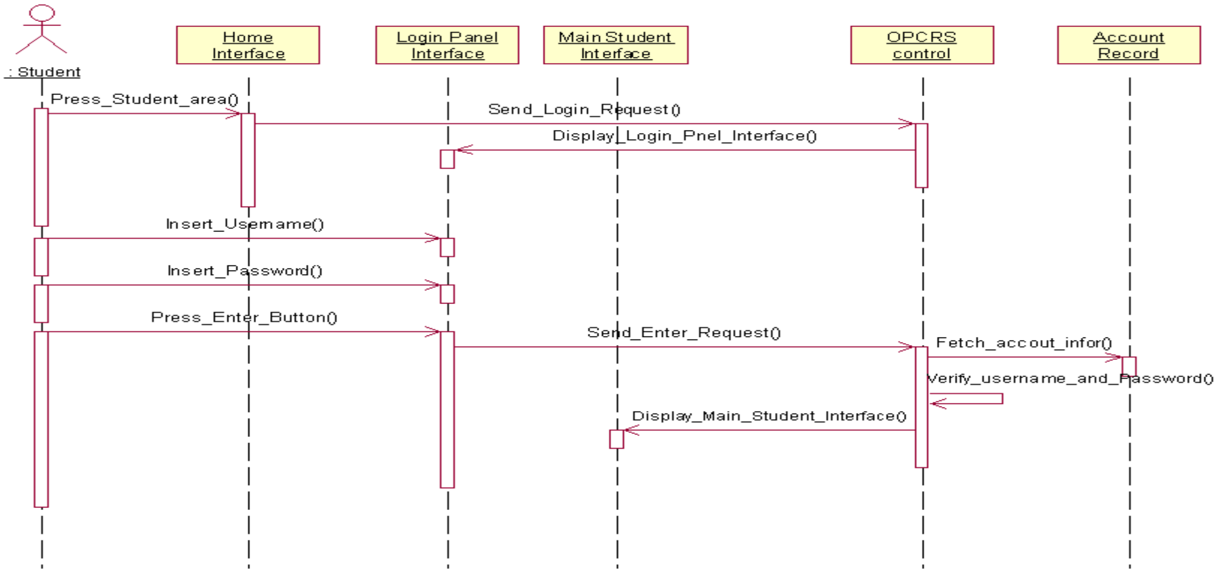


Figure 4.11: Login for Student

4.4.10.2. Login for Lecturer

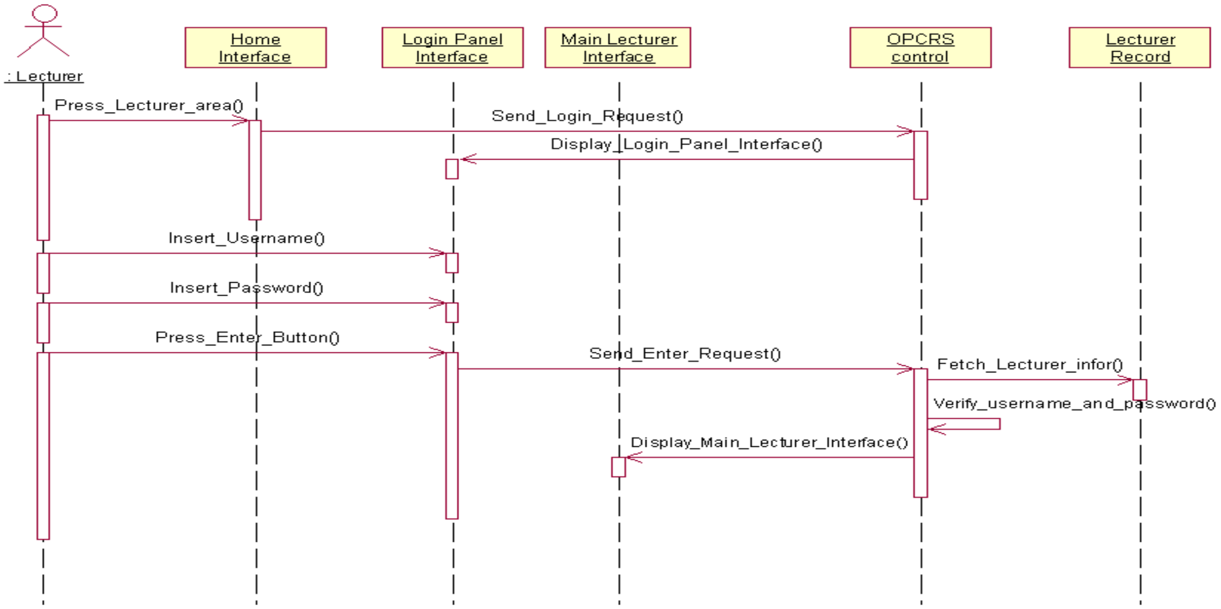


Figure 4.12: Login for Lecturer

4.4.10.3. Login for Administrator

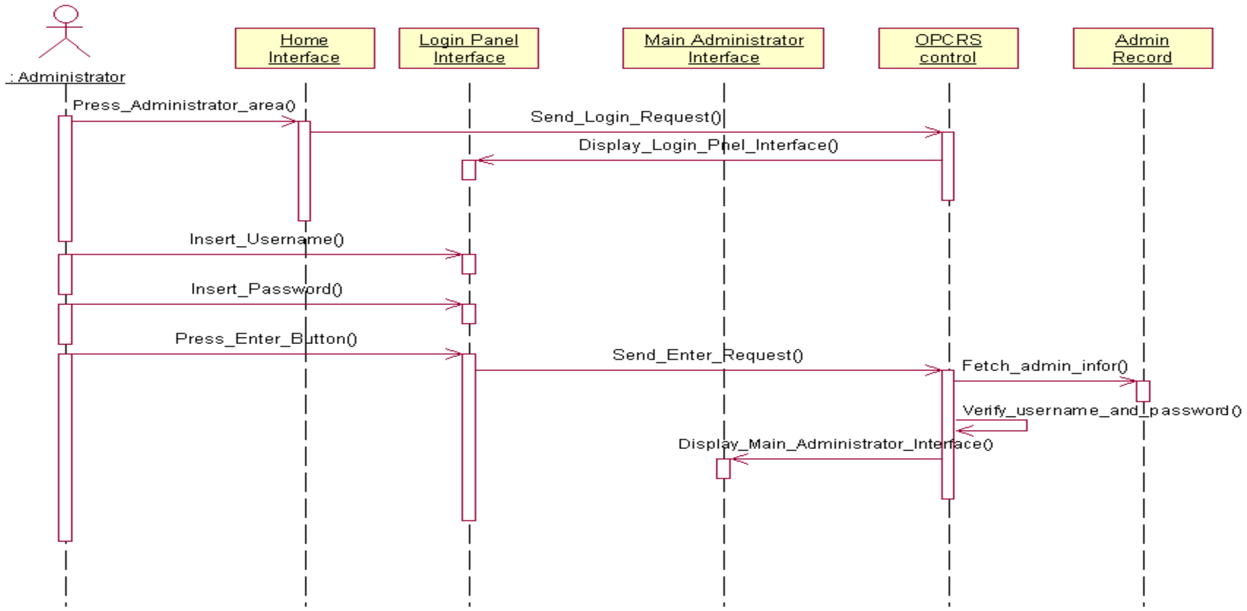


Figure 4.13: Login for Administrator

4.4.11. Send Message

The Lecturer in this sequence diagram can open send message page to send Email to his or her class members to enter his to this service by inserting message title, message description.

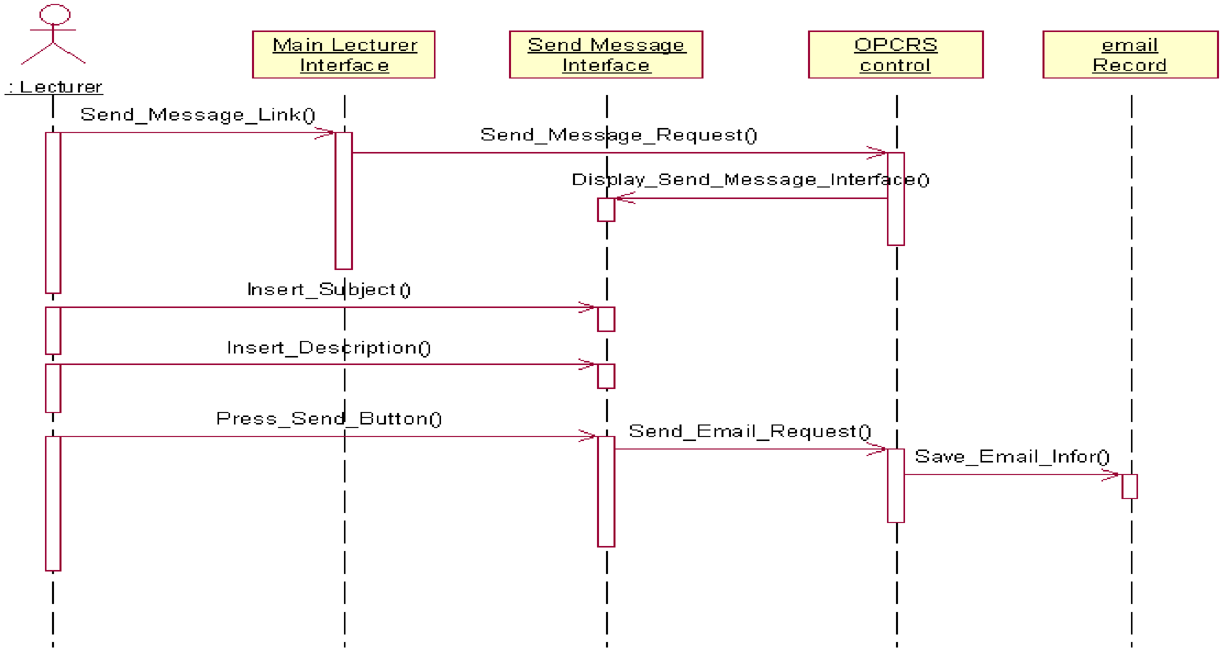
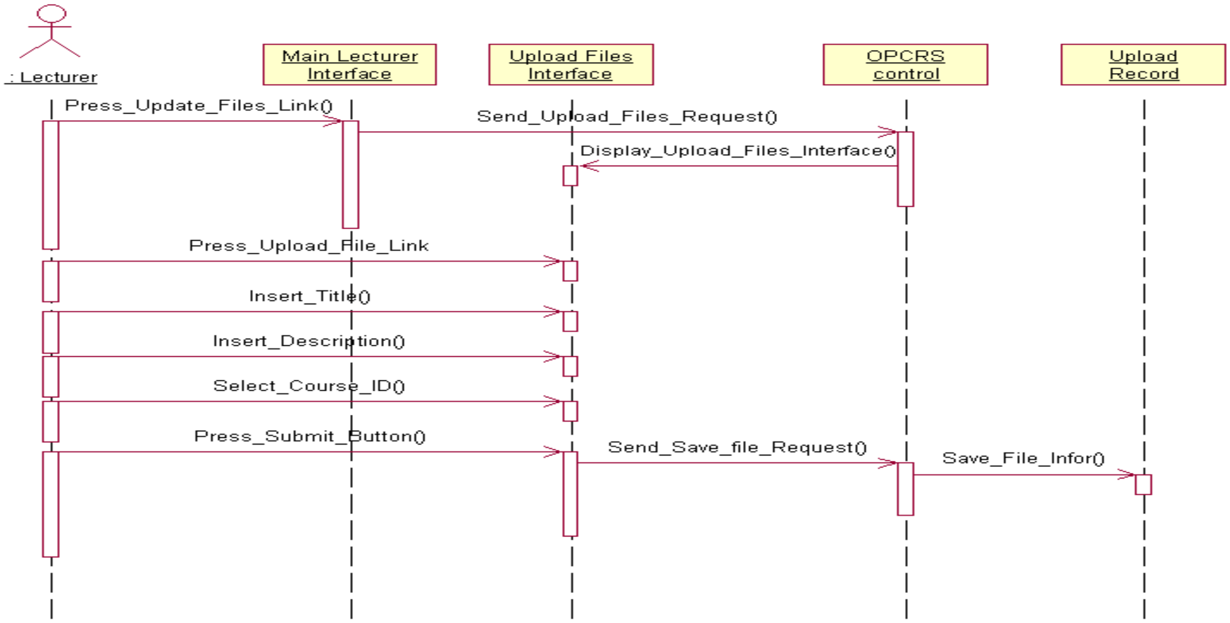


Figure 4.14: Send Message

4.4.12. Loading Files

The Lecturer in this sequence diagram can open upload page to upload files to his or her class members to this service by loading file , file title , file description and select course code.



4.15: Loading Files

4.4.13. Manage Course

The administrator in this sequence diagram can open course page to make manage for programming courses (Add programming course).

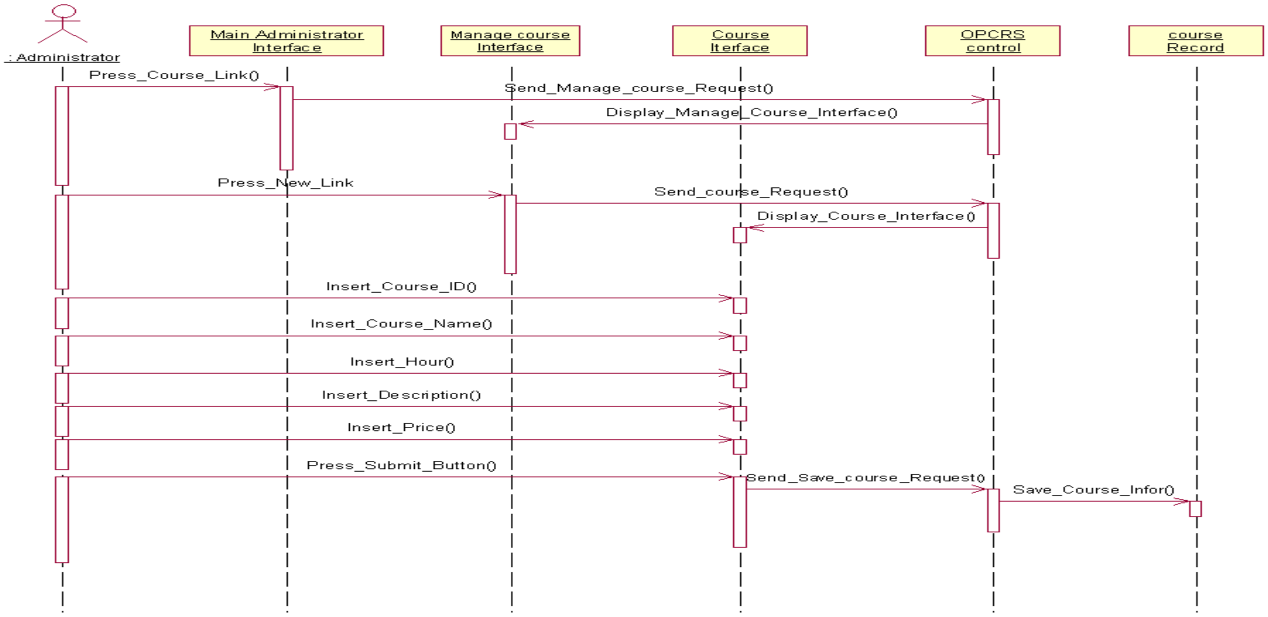


Figure 4.16: Manage Course

4.4.14. Manage Advertisement

The administrator in this sequence diagram can open Advertisement page to make manage for advertisement (Add new advertisement).

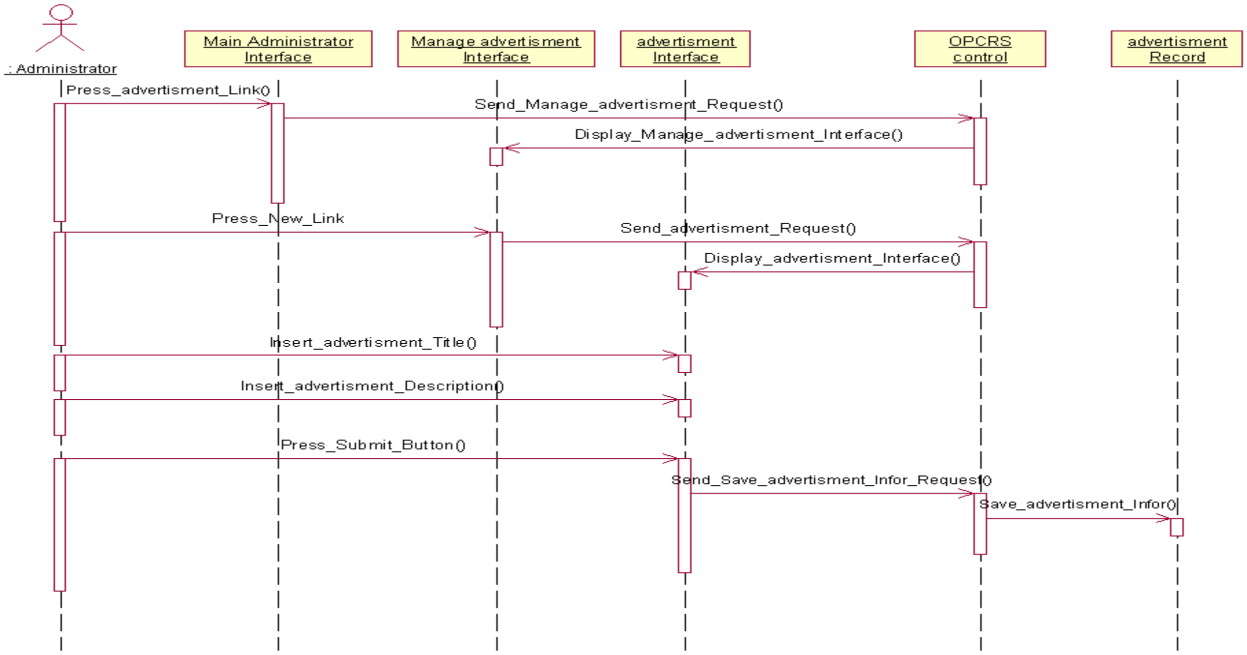

: Administrator
Main Administrator Interface
Manage advertisment Interface
advertisment Interface
OPCRS control
advertisment Record
Press_advertisment_Link()
Send_Manage_advertisment_Request()
Display_Manage_advertisment_Interface()
Press_New_Link
Send_advertisment_Request()
Display_advertisment_Interface()
Insert_advertisment_Title()
Insert_advertisment_Description()
Press_Submit_Button()
Send_Save_advertisment_Infor_Request()
Save_advertisment_Infor()

Figure 4.17: Manage Advertisement

4.4.15. Approval/Reject Student

The administrator in this sequence diagram can open Approval Page to approve or reject a student application to this service by press approval link or reject link.

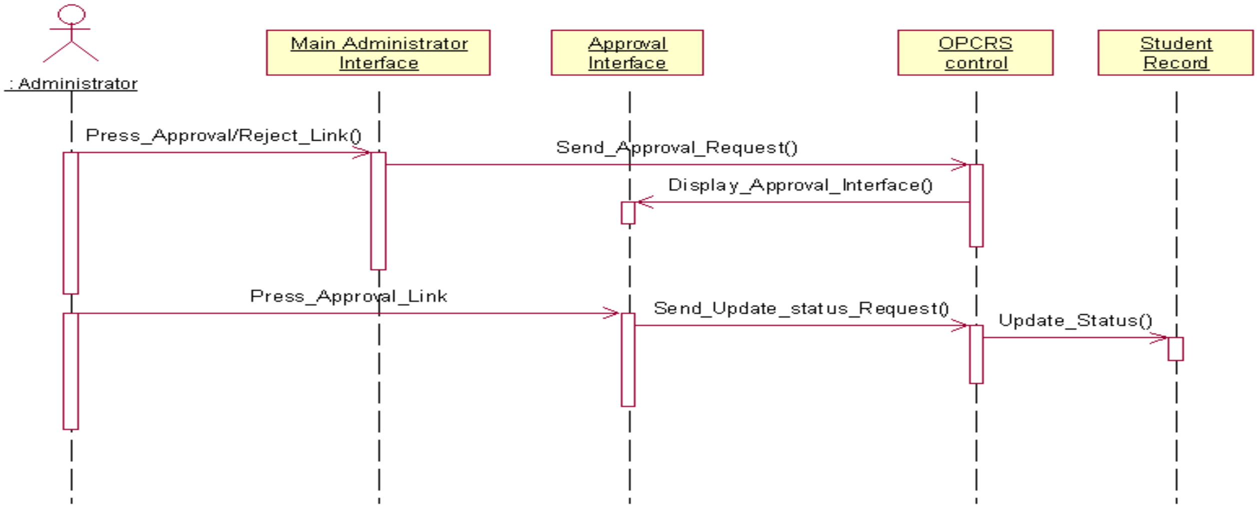


Figure 4.18: Approval/Reject Student

4.4.16. Manage Student

The administrator in this sequence diagram can open manage student page to make manage for student (Add new student).

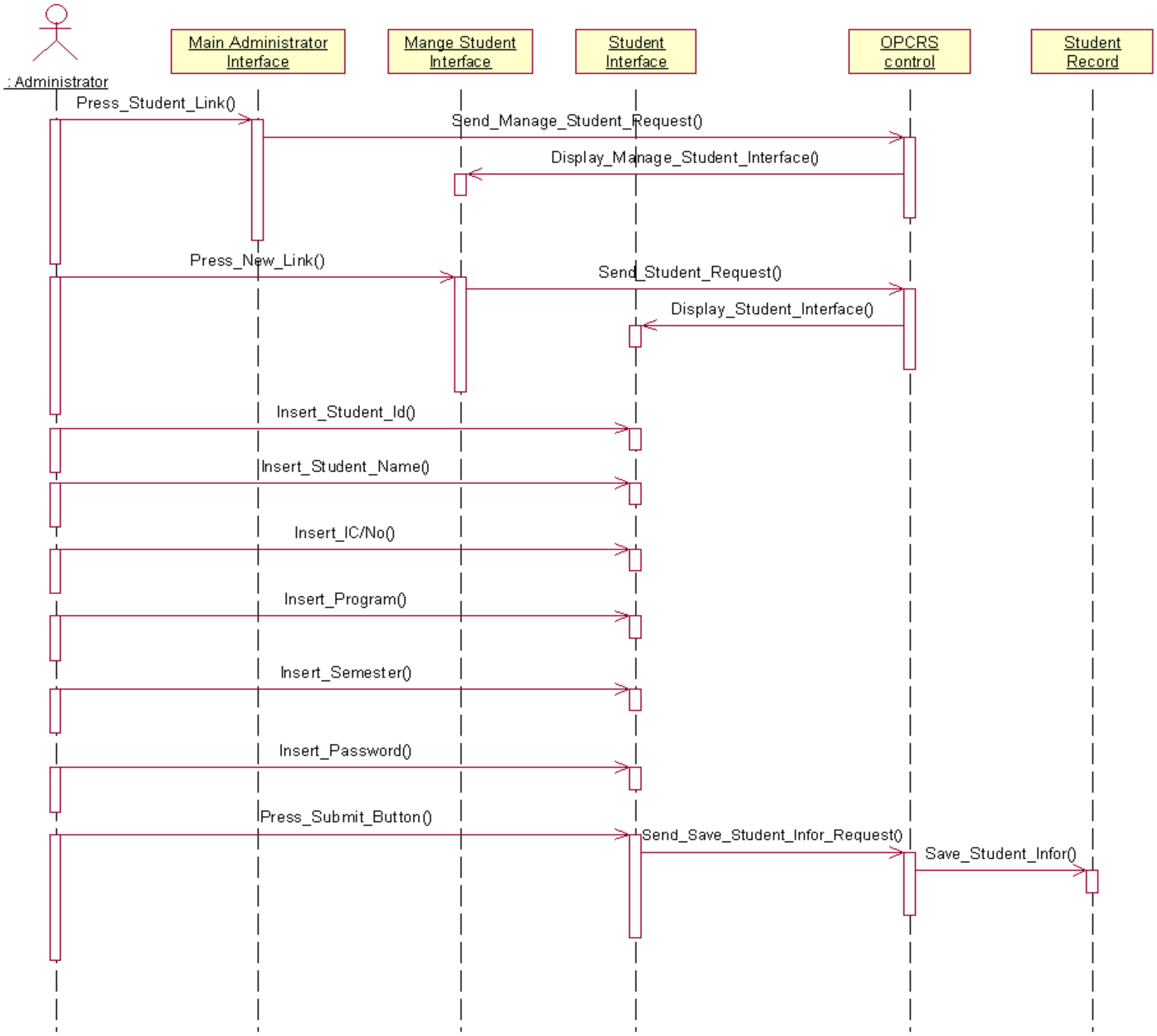


Figure 4.19: Manage Student

4.4.17. Manage Lecturer

The administrator in this sequence diagram can open manage lecturer page to make manage for lecturer (Add new student).

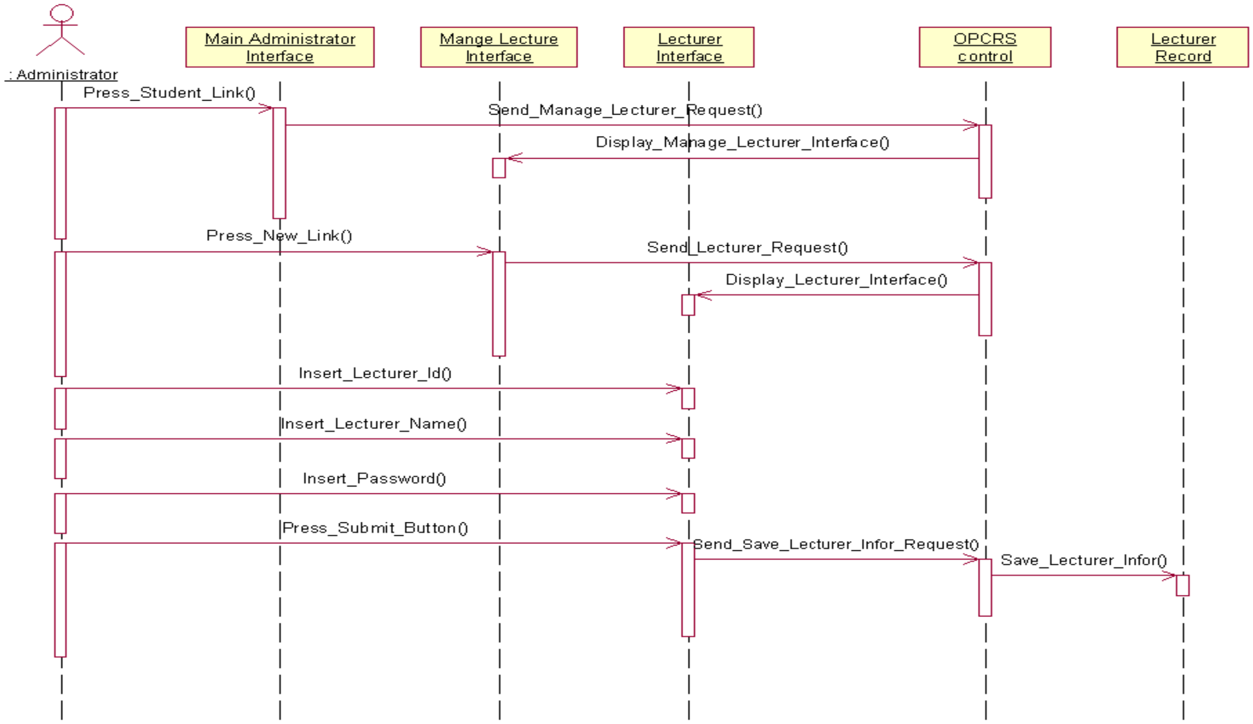


Figure 4.20: Manage Lecturer

4.5.System Development

The system development presented the language that used to build this application in order, the WML and Java Server Page (JSP) language used to develop this application.

4.5.1. Main Prototype Page

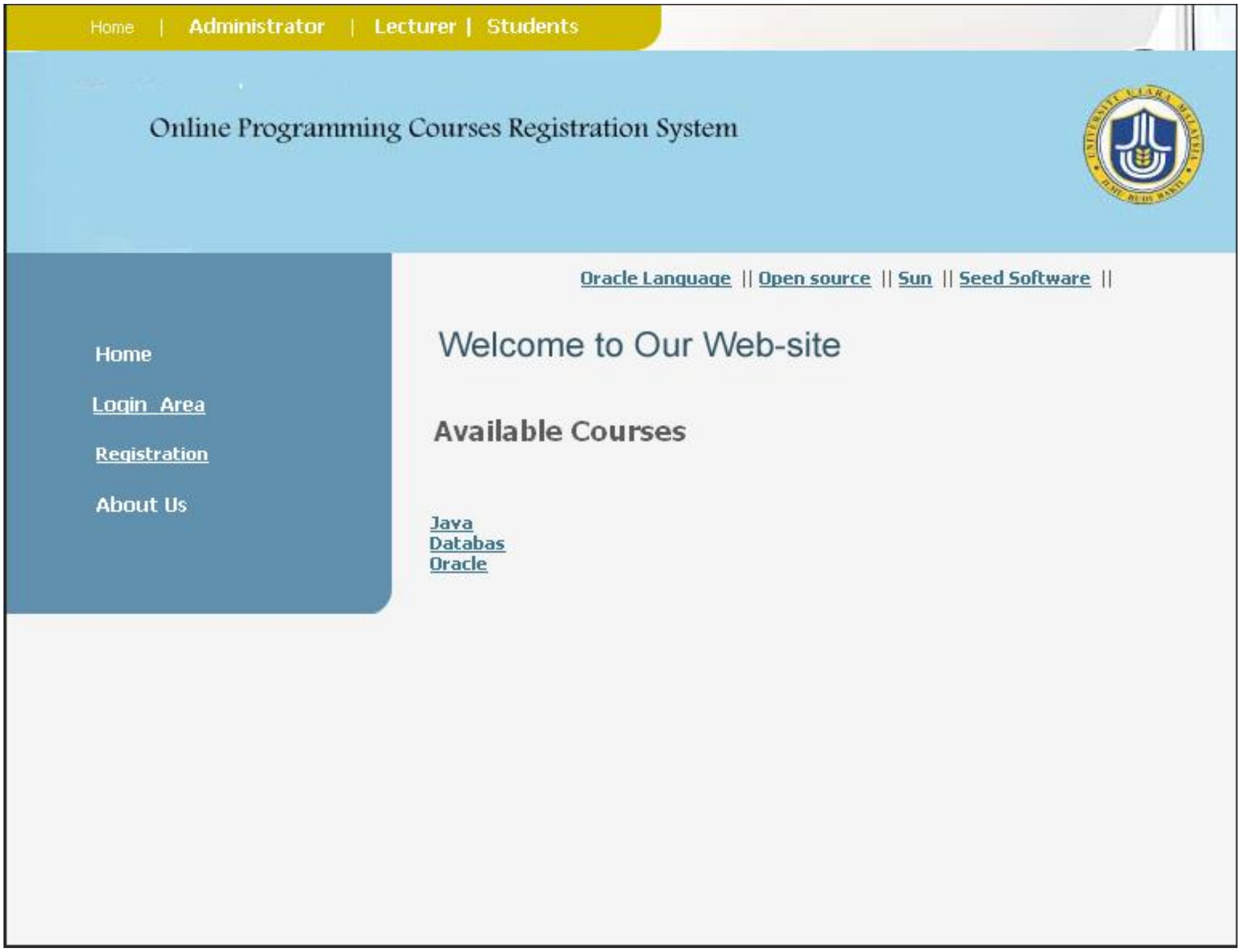
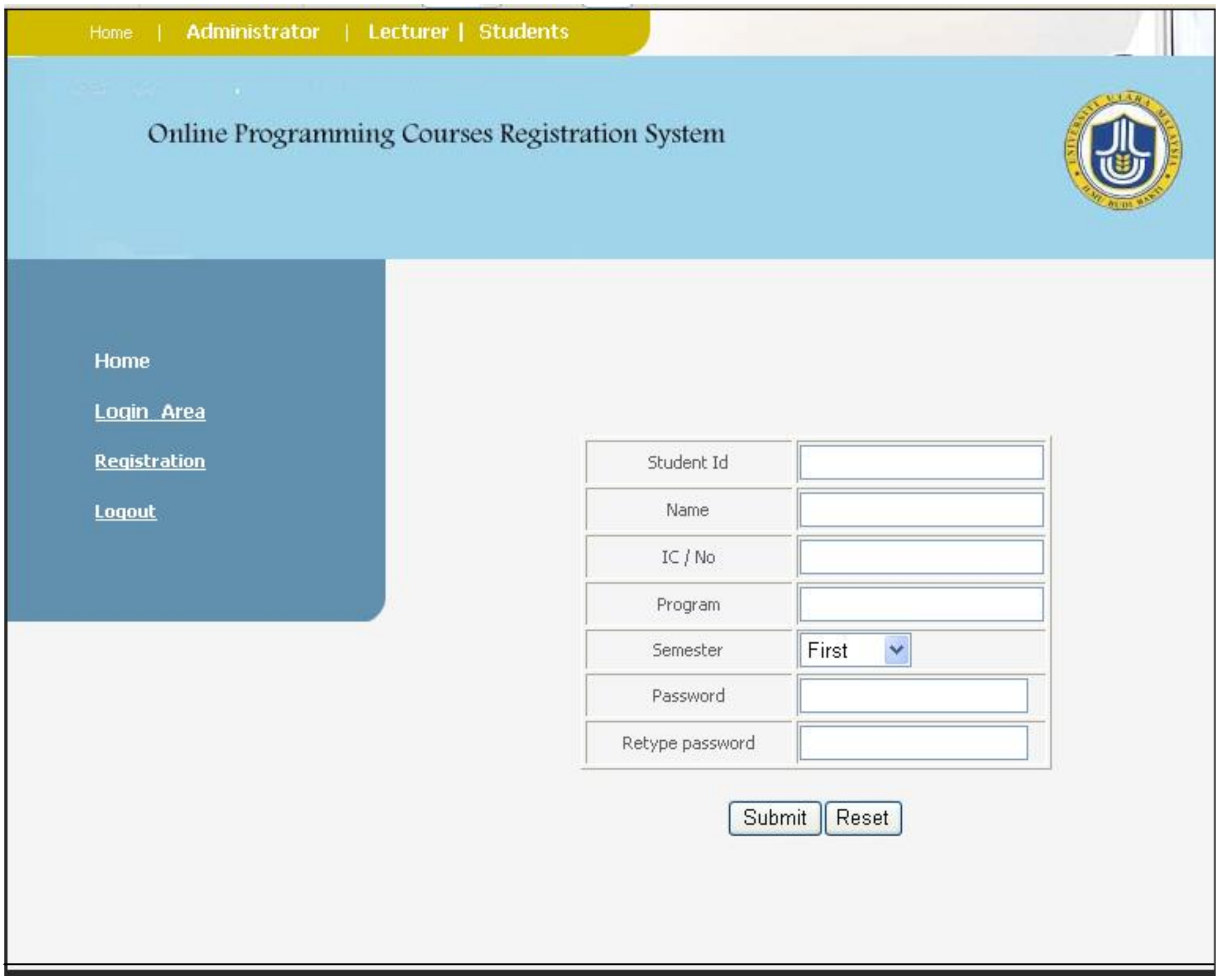

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Login Area
Registration
About Us
Welcome to Our Web-site
Available Courses
Java
Databas
Oracle

Figure 4.21: Main Prototype Page

Main page displays the title of the prototype and announcements for new programming courses such as new open source course, sun and seed software. In additional available programming courses which the students can register it such as Java, Database and Oracle recently. Also Login Area to let user open Login Page. The student can open student application by press registration Link.

4.5.1.1.Student Registration Page


Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Home
Login Area
Registration
Logout
Submit
Reset

Student Id	
Name	
IC / No	
Program	
Semester	First
Password	
Retype password	

Figure 4.22: Student Registration Page

After press "Registration" link in the Main Page, the prototype move to Student Registration Page. The Student can fill his or her application such as Student number, student name, IC/No, program, select Semester, password, and retype password. And then he will press Submit Button. The System will create new record and store this information into Student database. He can Press Back link to go back to Main page or Reset button to empty all fields.

4.5.1.2.Login Page

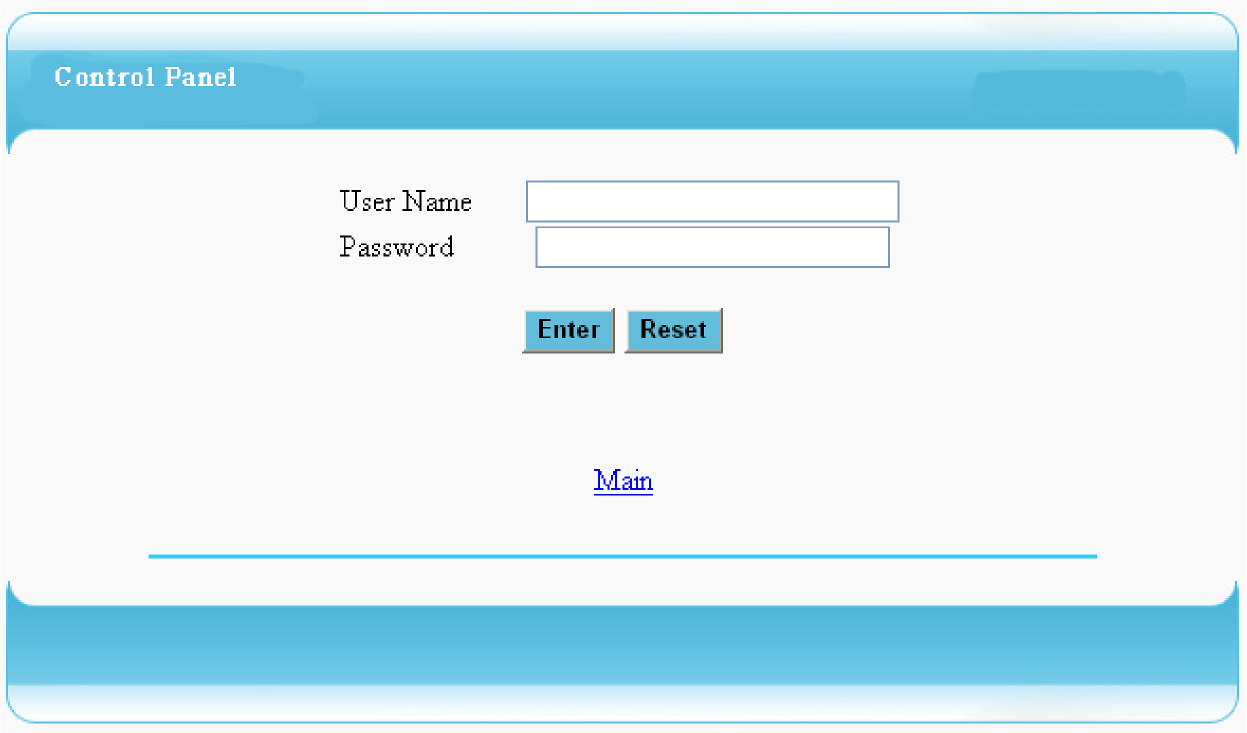
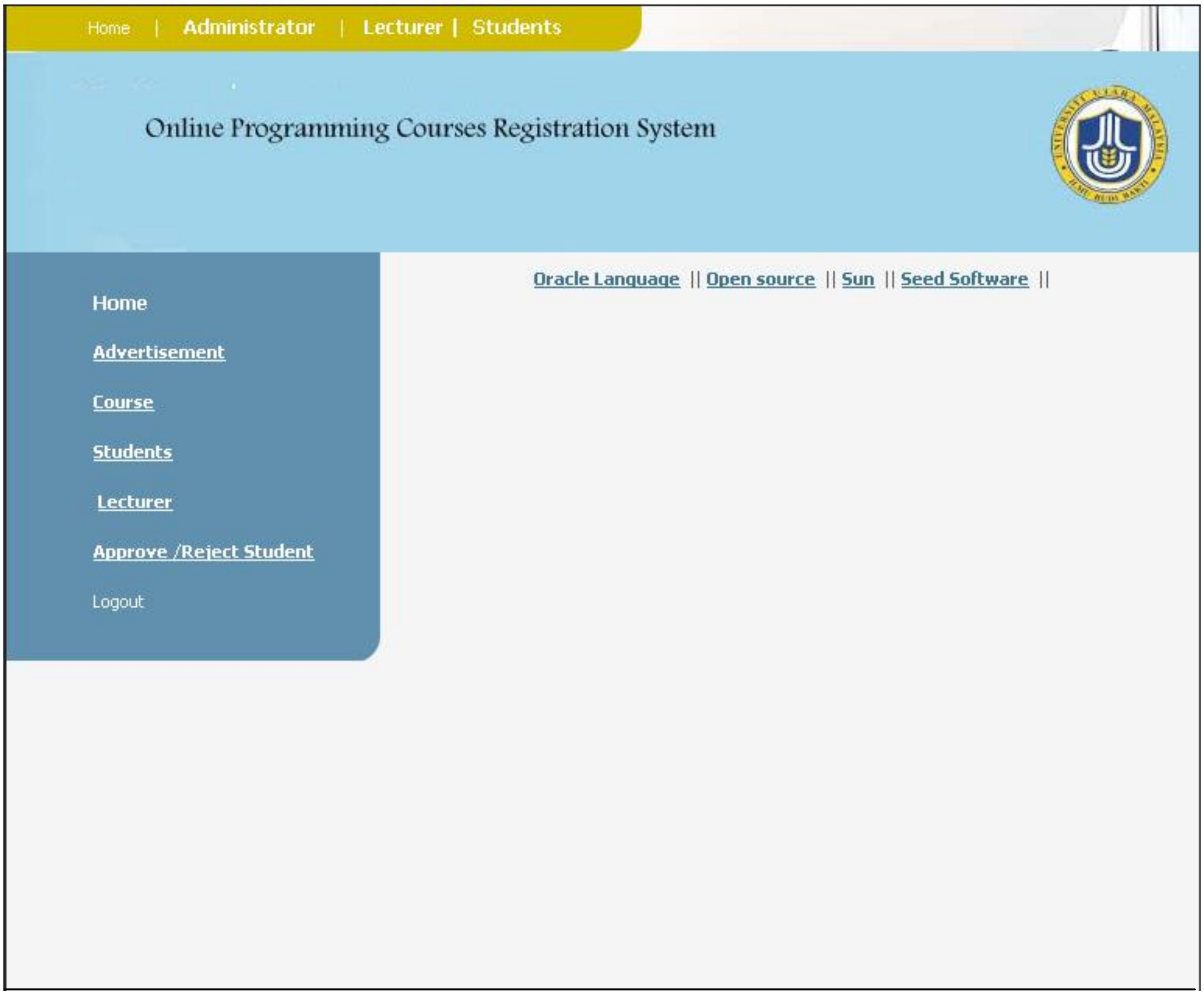


Figure 4.23: Login Page

After press the Login Area Link in the Home page the prototype move to Login page to enter user ID, password. Through this Page the user can enter to his or her main page. And the user can Press main to go back to Home page.

4.5.2. Main Administrator Page

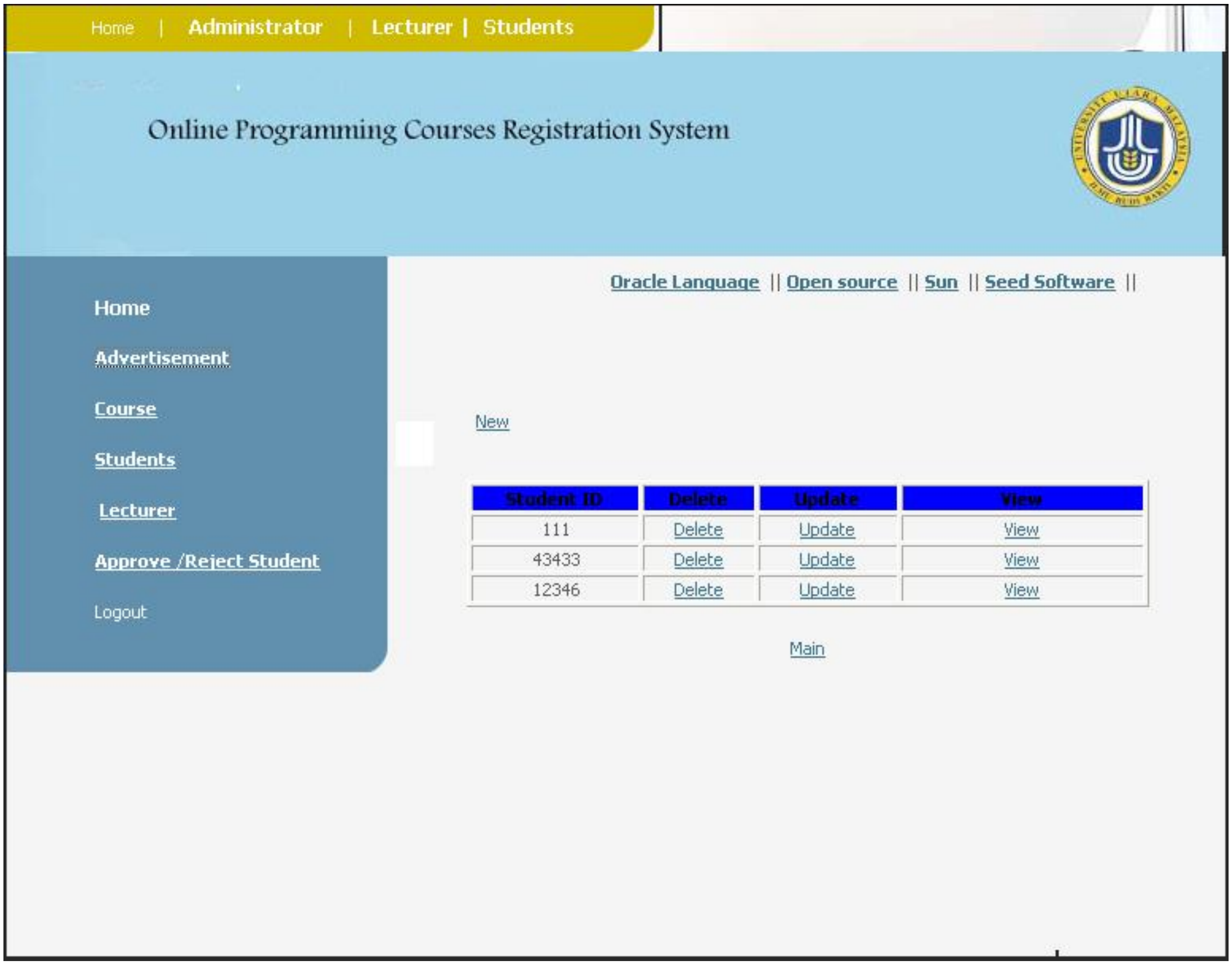


Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Advertisement
Course
Students
Lecturer
Approve /Reject Student
Logout

Figure 4.24: Main Administrator Page

Main Administrators Page displays functionality requirement (Primary functions) for administrator are Advertisement, courses, Students, Approval/Reject student, Lecturer, Home Page and Log Out. The administrator may press Home to go back to Home Page or he may press Log Out button to exist from his/her main page.

4.5.2.1.Student Page

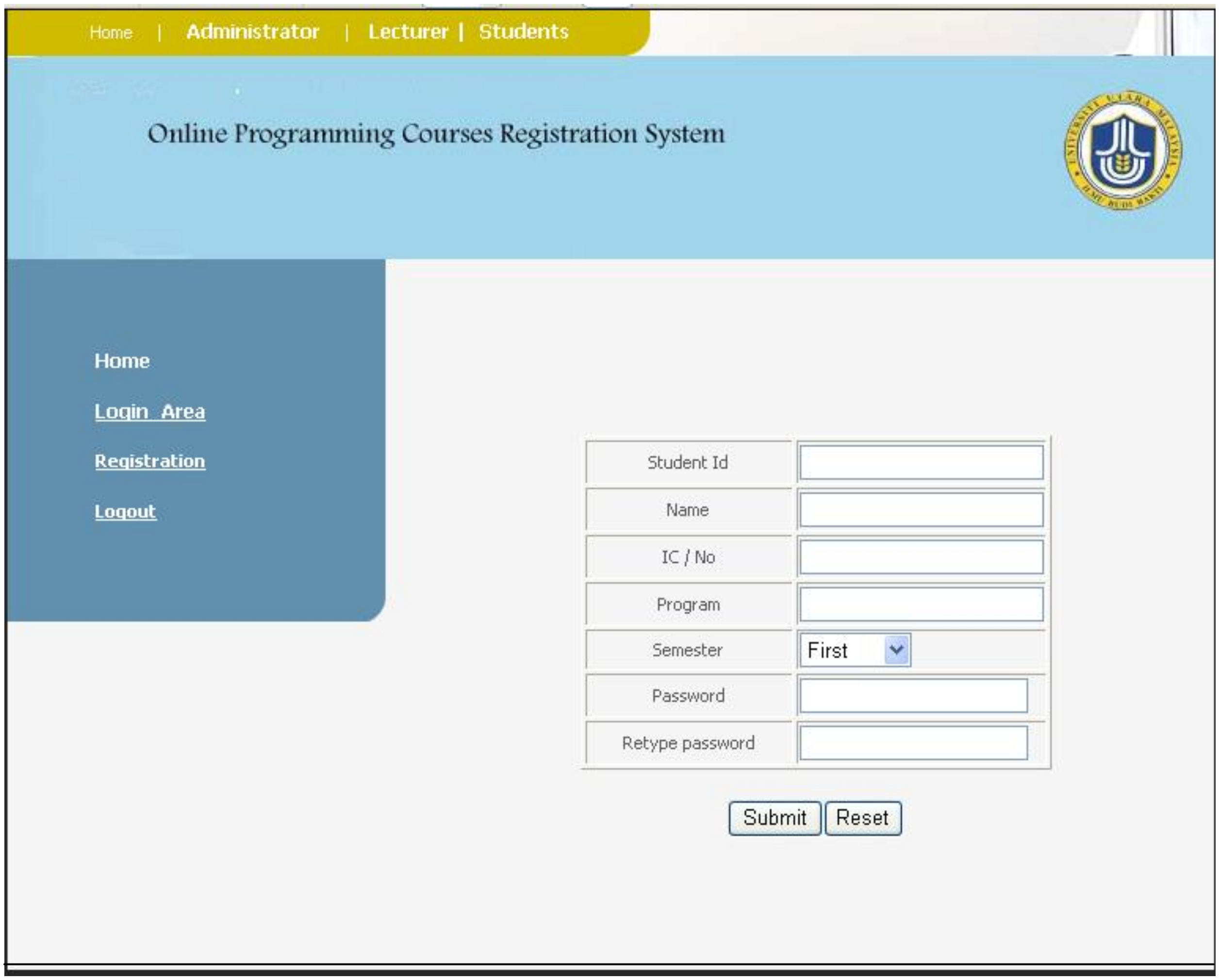

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Advertisement
Course
Students
Lecturer
Approve /Reject Student
Logout
New
Main

Student ID	Delete	Update	View
111	Delete	Update	View
43433	Delete	Update	View
12346	Delete	Update	View

Figure 4.25: Student Page

After press the Student Link in the Main Administrators Page, the prototype move to Student page. He can press "New" link to add new student or "Update" link to update student Information or "Delete" link to delete student or "View" link to view student information. And the administrator can Press main to go back to Main Administrators page or logout to exist from this page and go to Home page.

4.5.2.1.1.Add New Student Page

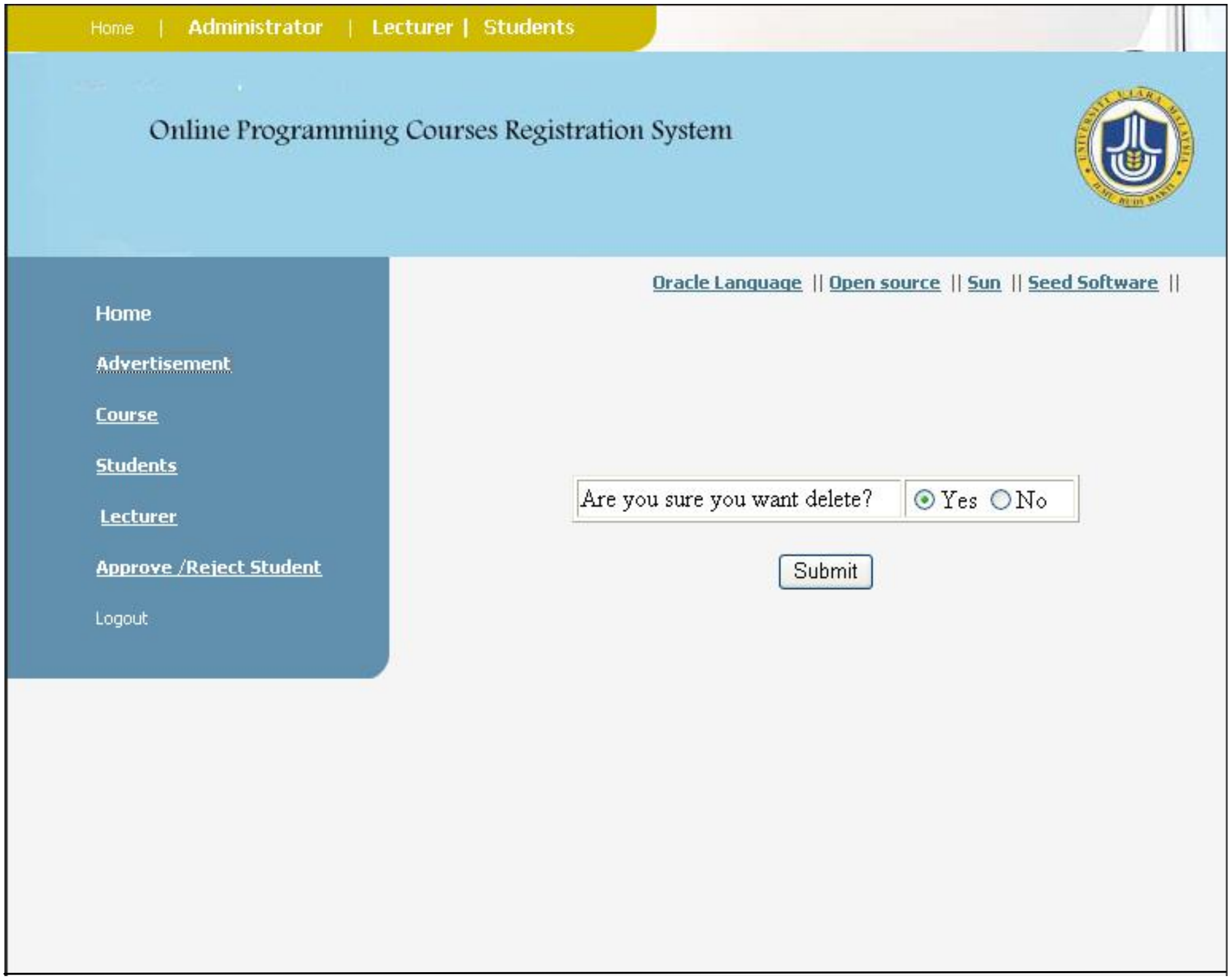

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Home
Login Area
Registration
Logout
Submit
Reset

Student Id	
Name	
IC / No	
Program	
Semester	First
Password	
Retype password	

Figure 4.26: Add New Student Page

After press “New” link in the Student Page, the prototype move to Add Student Page. The administrator will insert information for new Student such as Student number, student name, IC/No, program, select Semester password, and retype password. And then he will press Submit Button. The System will create new record and store this information into Student database. He can Press Back link to go back to Student page or Reset button to empty all fields.

4.5.2.1.2. Delete Student Page

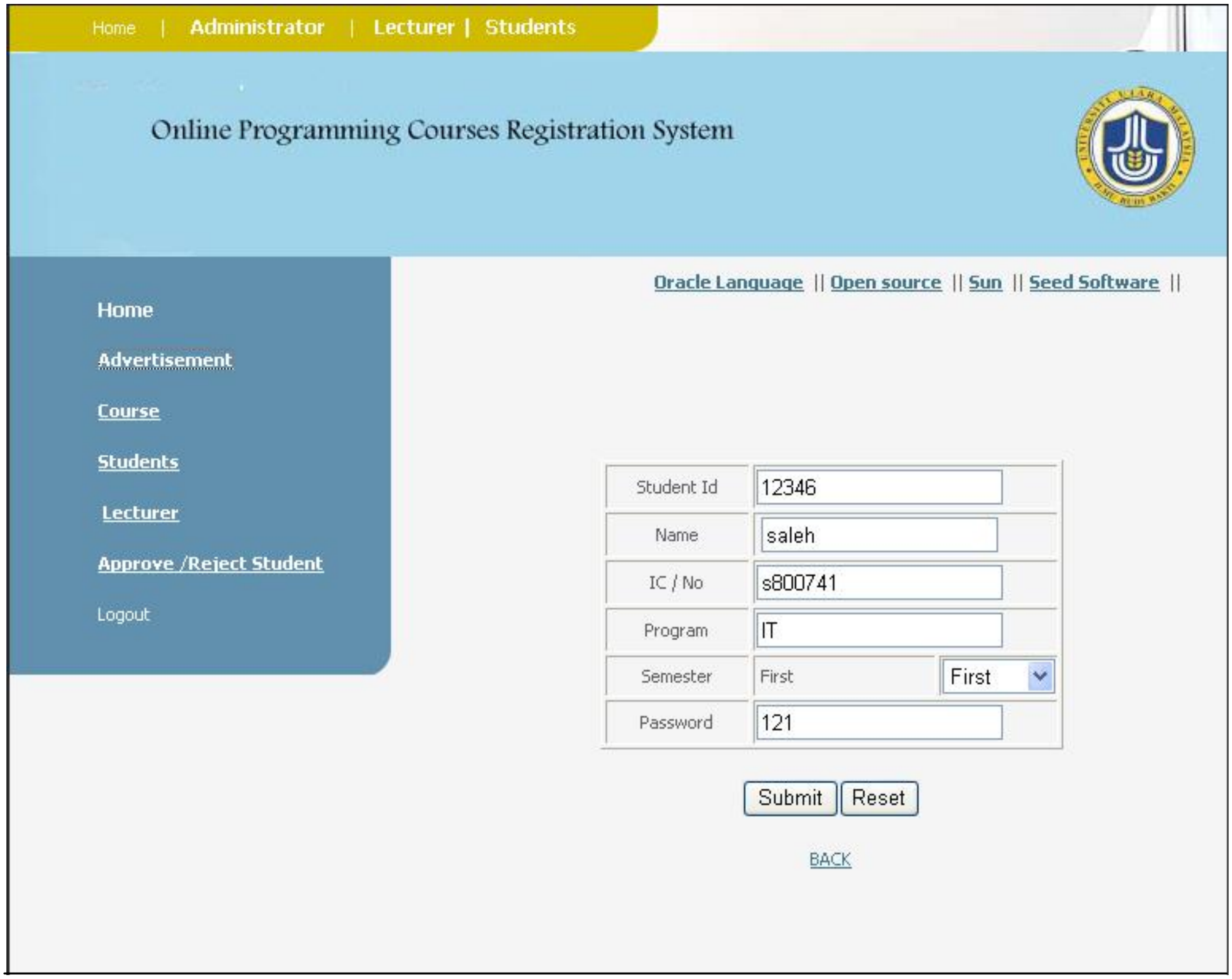

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Advertisement
Course
Students
Lecturer
Approve /Reject Student
Logout
Submit

Are you sure you want delete? ● Yes ○ No

Figure 4.27: Delete Student Page

After press “Delete” Link in the Student Page, the prototype move to Delete Student Page. The System will ask the administrator to confirm Delete process. The administrator will press “Yes” option if he wants delete student information. And then he will press submit button. The system will delete student record from Student database. Or He can Press “No” option to cancel Delete process and then he will press submit button to go back to Student page.

4.5.2.1.3. Update Student Information Page

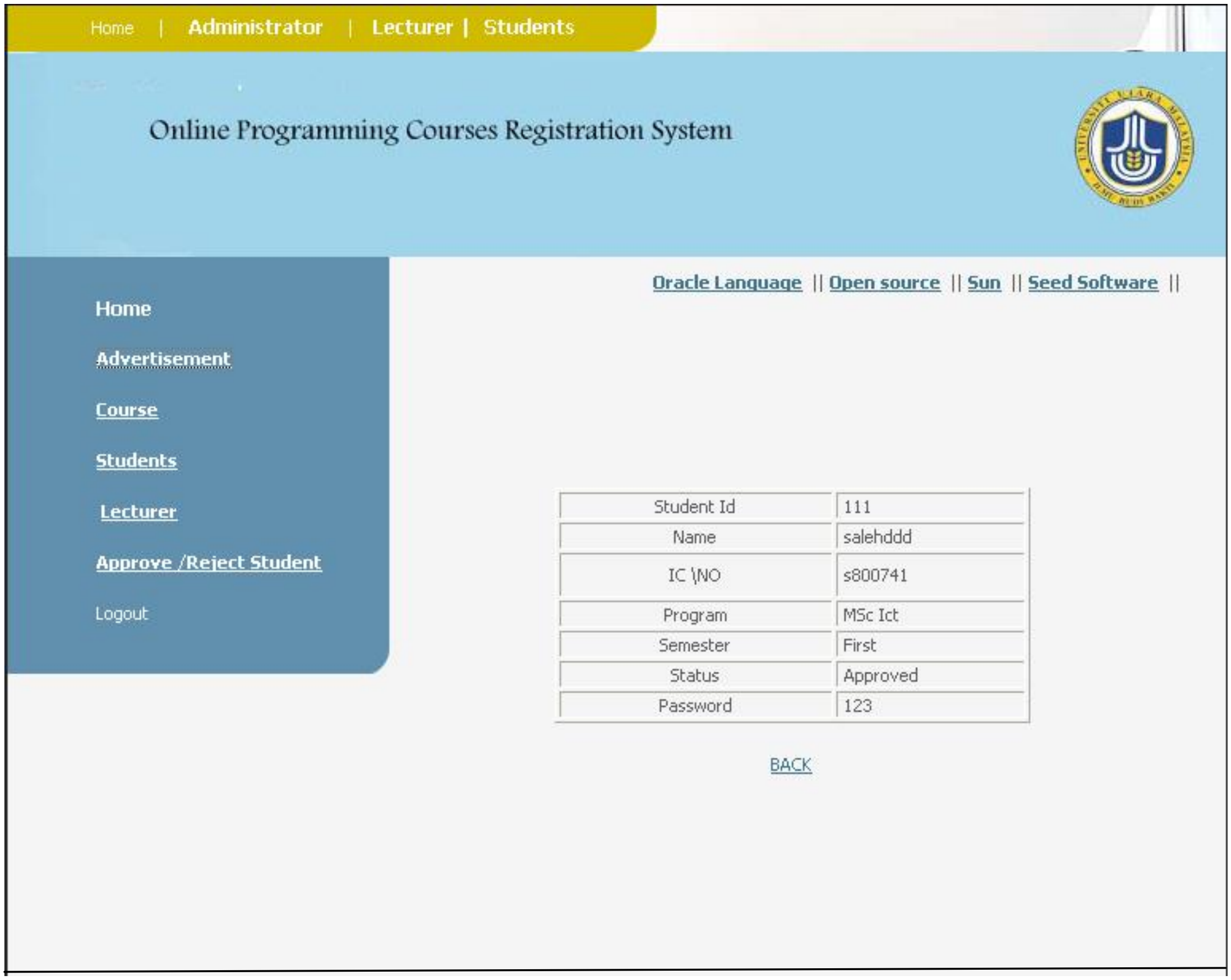

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Advertisement
Course
Students
Lecturer
Approve /Reject Student
Logout
Submit
Reset
BACK

Student Id	12346	
Name	saleh	
IC / No	s800741	
Program	IT	
Semester	First	First
Password	121	

Figure 4.28: Update Student Information Page

After press “Update” Link in the Student Page, the prototype moves to Update Student Information Page. After the administrator finishes update his/her student information, then he will press submit button. And the system will update old student information to new student information. Or He can Press Back link to go back to Student page or Reset button to empty all fields.

4.5.2.1.4. View Student Information Page

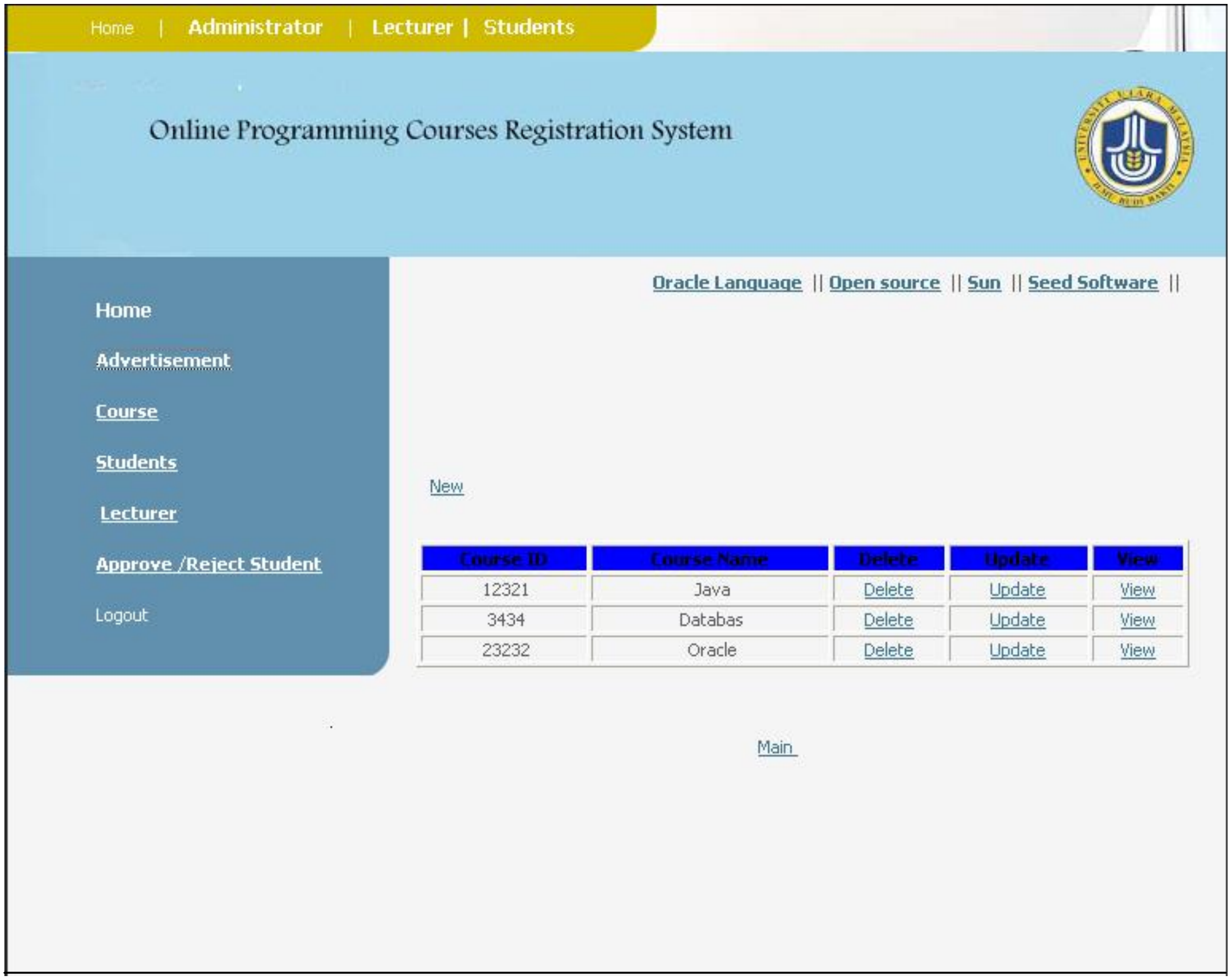

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Advertisement
Course
Students
Lecturer
Approve /Reject Student
Logout
BACK

Student Id	111
Name	salehddd
IC \NO	s800741
Program	MSc Ict
Semester	First
Status	Approved
Password	123

Figure 4.29: View Student Information Page

After press “View” link in the Student Page, the prototype moves to View Student Information Page to view student information. And the system will fetch student information from student database. After that he can Press Back link to go back to Student page.

4.5.2.2.Course Page


Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Advertisement
Course
Students
Lecturer
Approve /Reject Student
Logout
New
Main

Course ID	Course Name	Delete	Update	View
12321	Java	Delete	Update	View
3434	Databas	Delete	Update	View
23232	Oracle	Delete	Update	View

Figure 4.30: Course Page

After press “Course” link in the Main Administrators Page, the prototype moves to Course page. He can press “New” link to add new programming course or “Update” link to update programming course Information or “Delete” link to delete programming course or “View” link to view course information. And the administrator can Press main to go back to Main Administrators page or logout to exist from this page and go to Home page.

4.5.2.2.1. Add New Programming Course Page

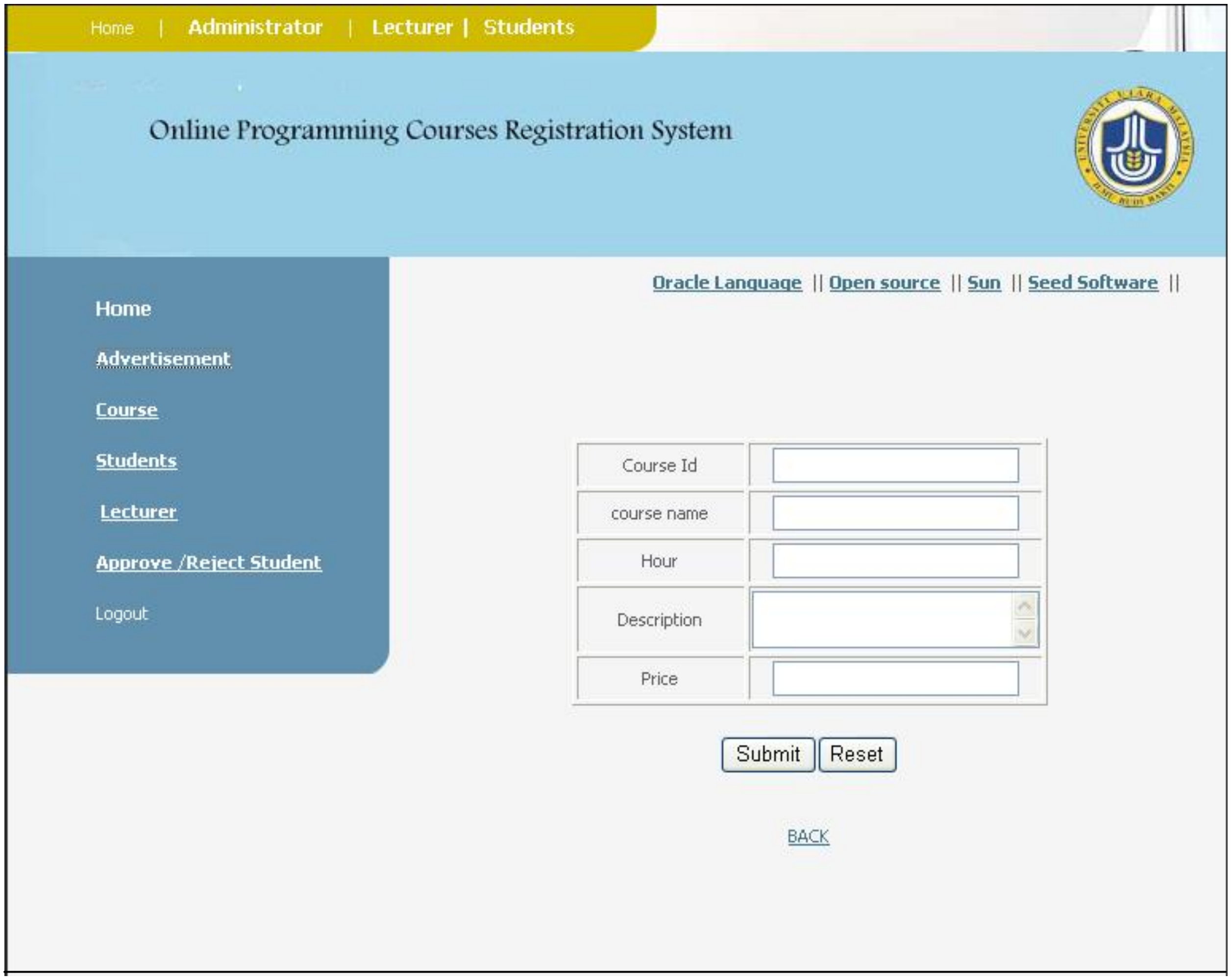
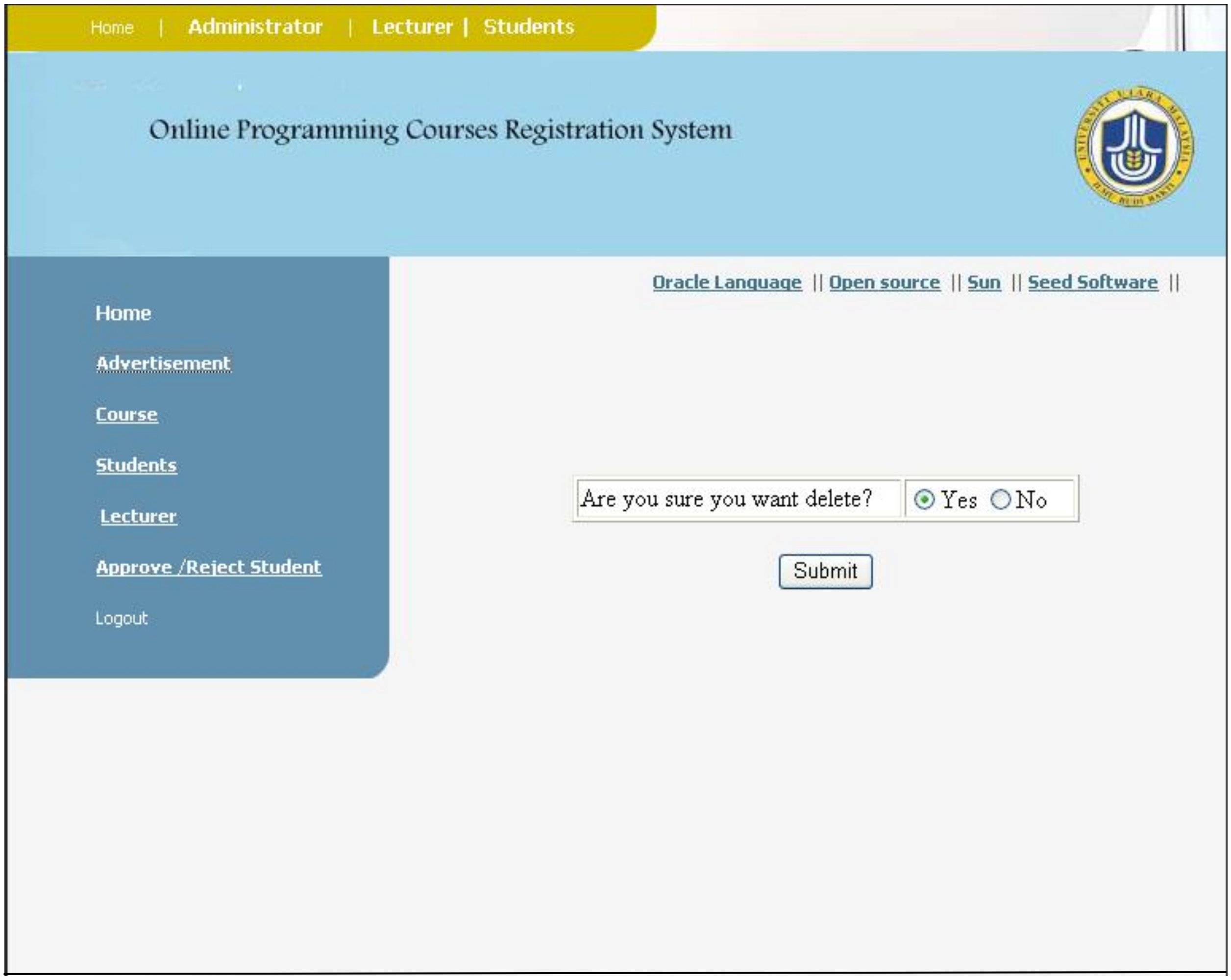

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Advertisement
Course
Students
Lecturer
Approve /Reject Student
Logout
Course Id
course name
Hour
Description
Price
Submit
Reset
BACK

Figure 4.31: Add New Programming Course Page

After press "New" link in the Course Page, the prototype move to Add Programming Course Page. The administrator will insert information for new programming course such as course number, course name, Hour, Description and price. And then he will press Submit Button. The System will create new record and store this information into Course database. He can Press Back link to go back to Course page or Reset button to empty all fields.

4.5.2.2.2.Delete programming Course Page

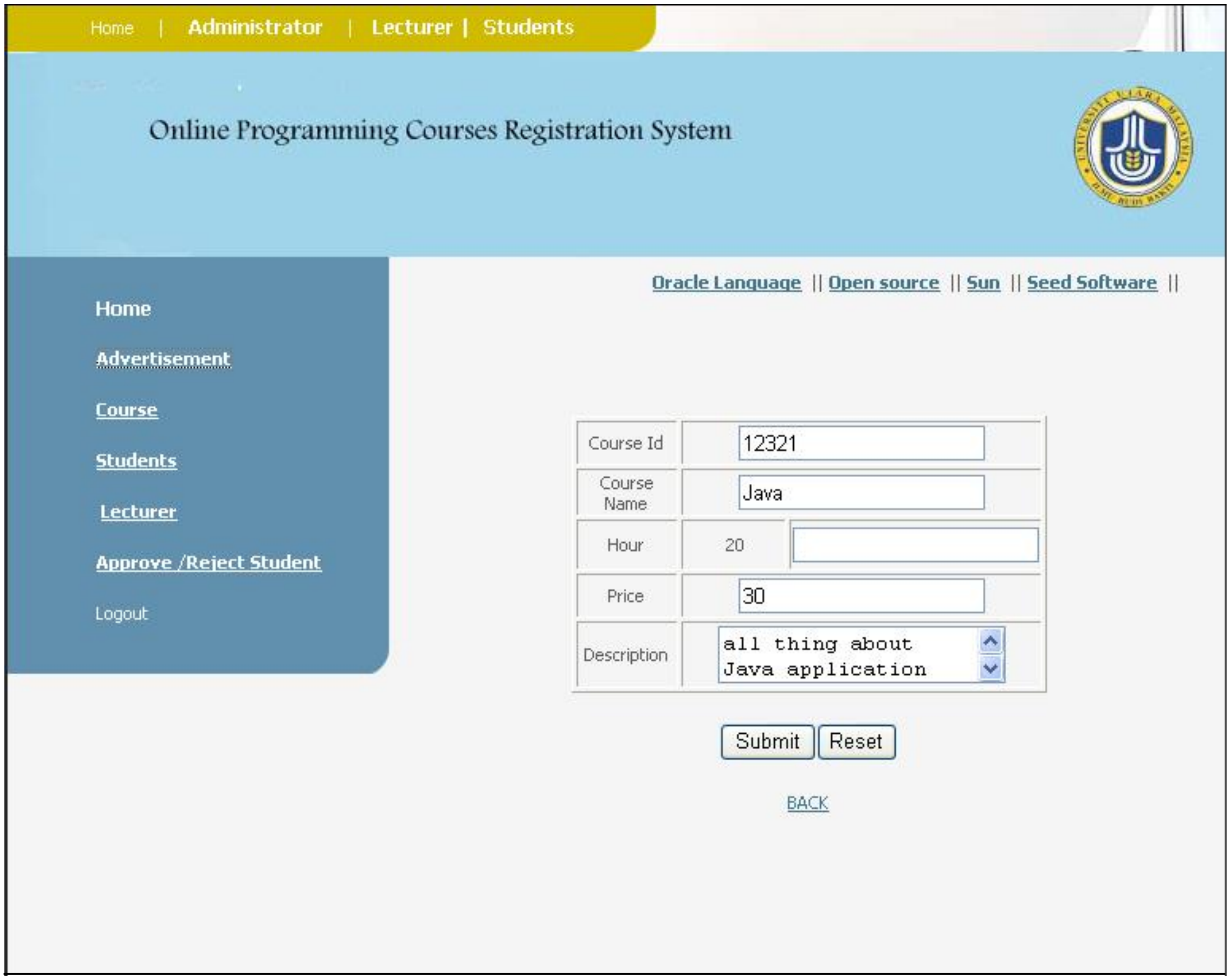

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Advertisement
Course
Students
Lecturer
Approve /Reject Student
Logout
Submit

Are you sure you want delete? ● Yes ○ No

Figure 4.32: Delete Programming course Page

After press “Delete” Link in the Course Page, the prototype move to Delete Programming Course Page. The System will ask the administrator to confirm Delete process. The administrator will press “Yes” option if he wants delete Programming Course information. And then he will press submit button. The system will delete Programming Course record from Course database. Or He can Press “No” option to cancel Delete process and then he will press submit button to go back to Course page.

4.5.2.2.3. Update Programming Course Information Page

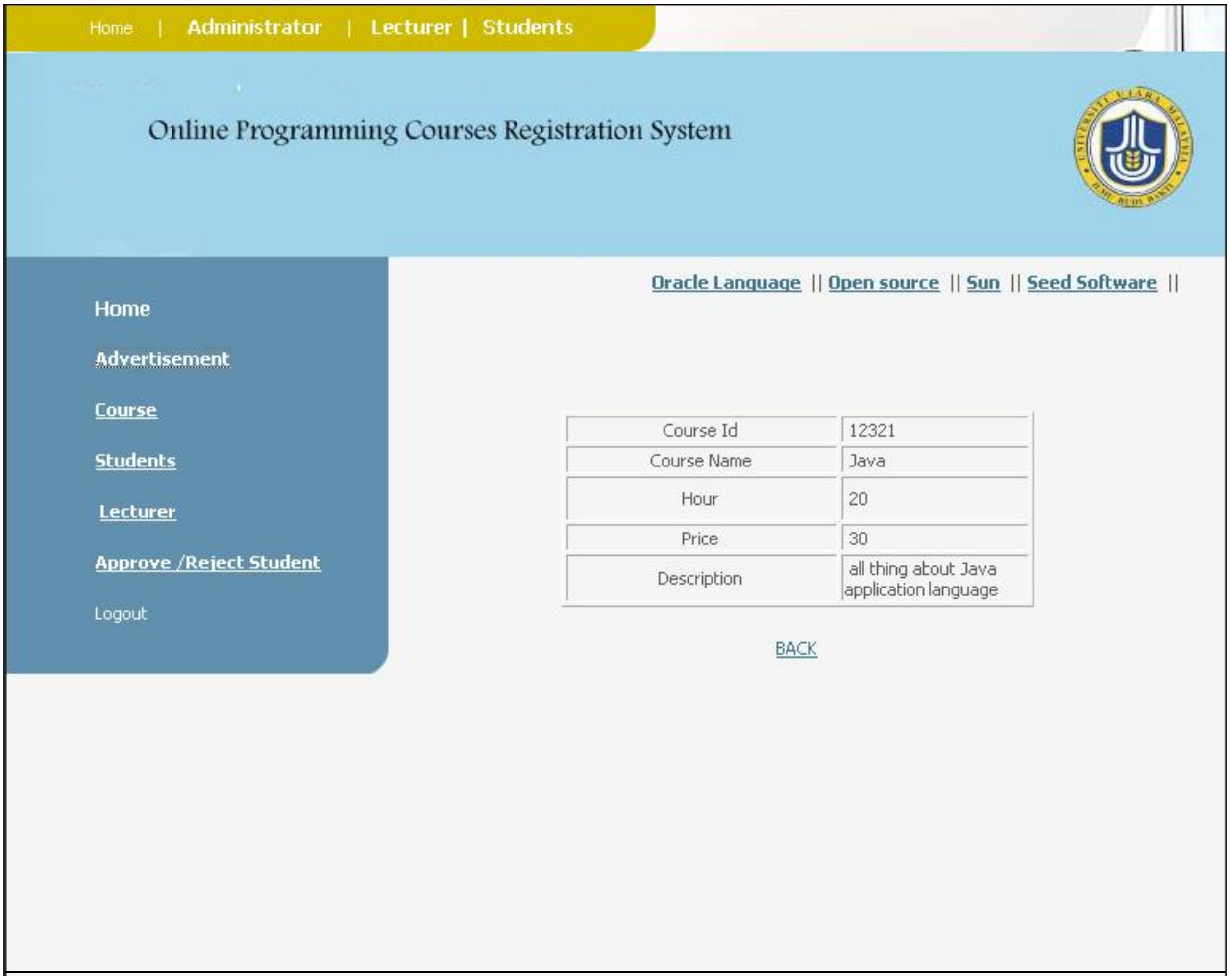

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Advertisement
Course
Students
Lecturer
Approve /Reject Student
Logout
Submit
Reset
BACK

Course Id	12321
Course Name	Java
Hour	20
Price	30
Description	all thing about Java application

Figure 4.33: Update Programming Course Information Page

After press “Update” Link in the Course Page, the prototype moves to Update Programming Course Information Page. After administrator finishes update his/her course information, then he will press submit button. And the system will update old programming course information to new programming course information. Or He can Press Back link to go back to Course page or Reset button to empty all fields.

4.5.2.2.4. View Programming Course Information Page

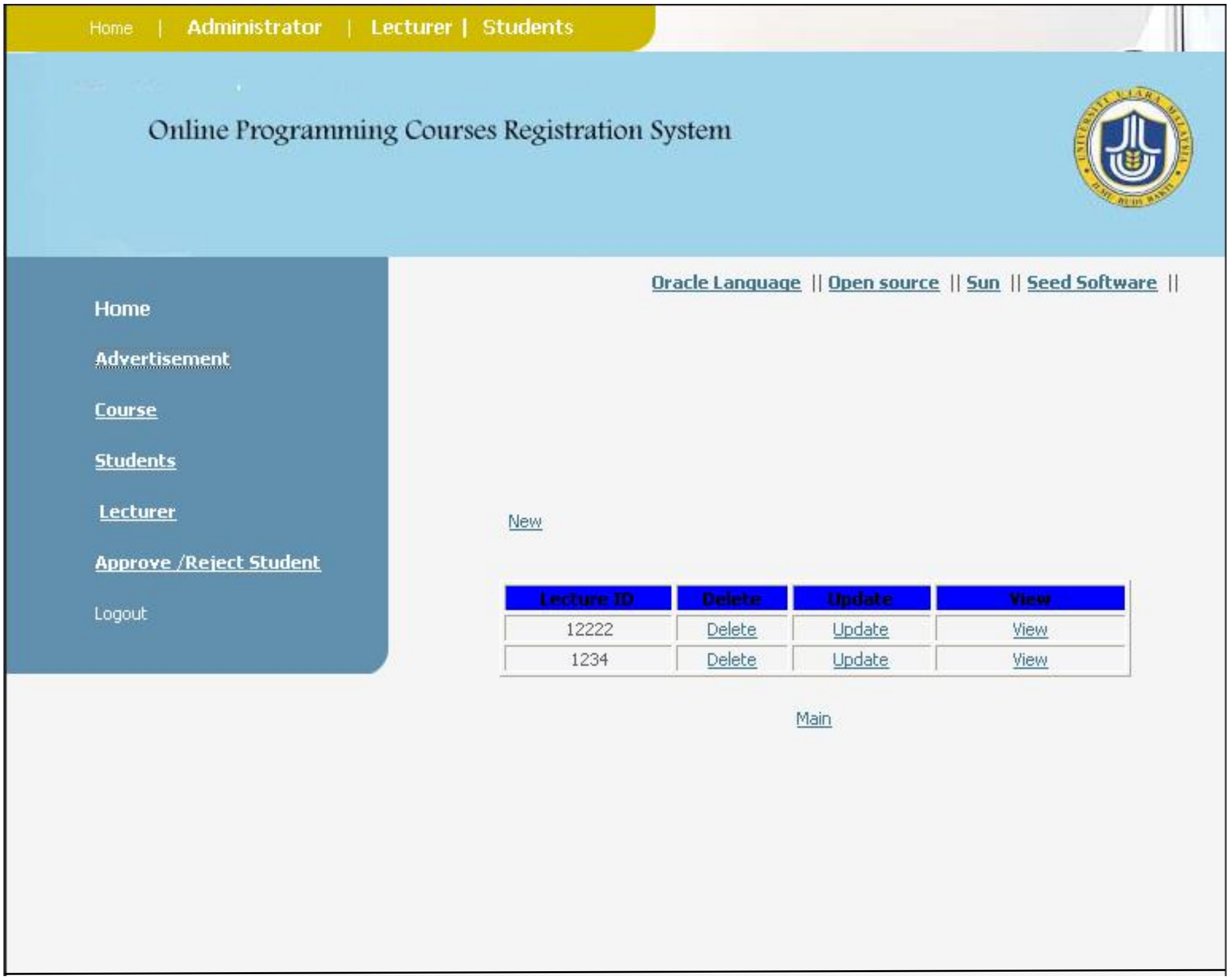

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Advertisement
Course
Students
Lecturer
Approve /Reject Student
Logout
BACK

Course Id	12321
Course Name	Java
Hour	20
Price	30
Description	all thing about Java application language

Figure 4.34: View Programming Course Information Page

After press "View" link in the Course Page, the prototype moves to View Programming Course Information Page. And the system will fetch programming course information from course database. After that he can Press Back link to go back to Course page.

4.5.2.3.Lecturer Page

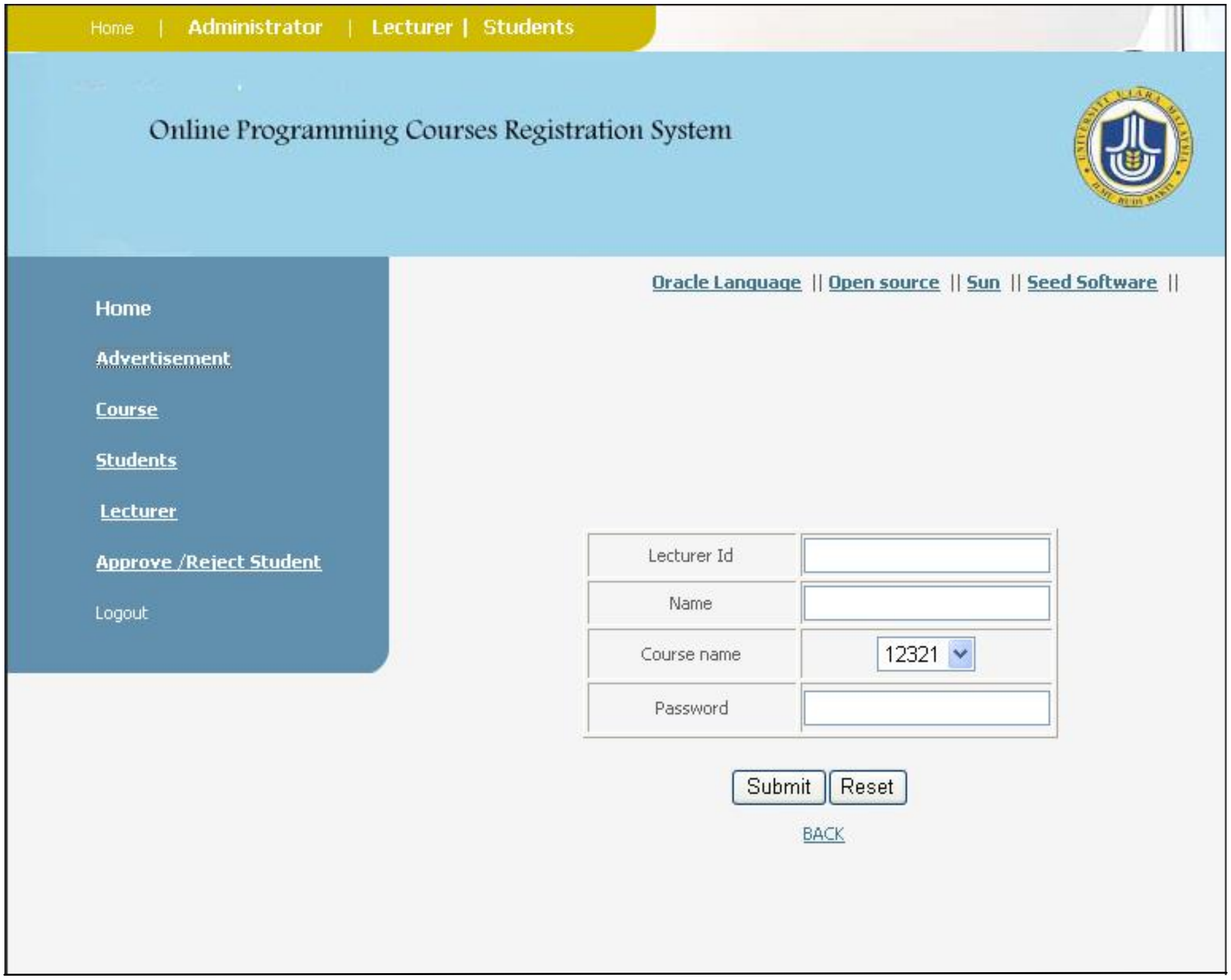

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Advertisement
Course
Students
Lecturer
Approve /Reject Student
Logout
New
Main

Lecture ID	Delete	Update	View
12222	Delete	Update	View
1234	Delete	Update	View

Figure 4.35: Lecturer Page

After press “Lecturer” link in the Main Administrators Page, the prototype moves to Lecturer page. He can press “New” link to add new lecturer or “Update” link to update lecturer Information or “Delete” link to delete lecturer or “View” link to view lecturer information. And the administrator can Press main to go back to Main Administrators page or logout to exist from this page and go to Home page.

4.5.2.3.1. Add New Lecturer Page

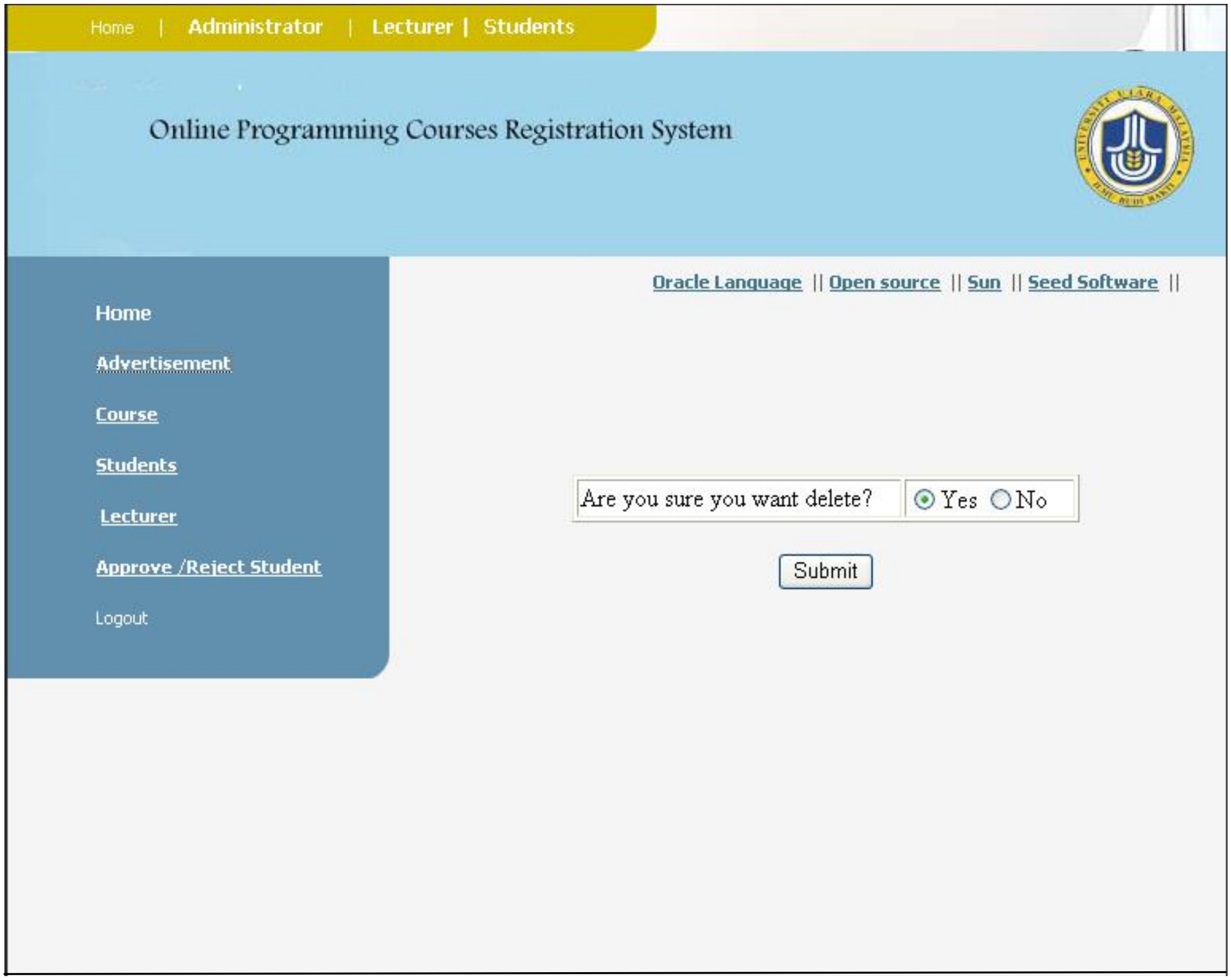

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Advertisement
Course
Students
Lecturer
Approve /Reject Student
Logout
Submit
Reset
BACK

Lecturer Id	
Name	
Course name	12321
Password	

Figure 4.36: Add New Lecturer Page

After press "New" link in the Lecturer Page, the prototype move to Add Lecturer Page. The administrator will insert information for new Lecturer such as lecturer number, lecturer name, course number and password. And then he will press Submit Button. The System will create new record and store this information into Lecturer database. He can Press Back link to go back to Course page or Reset button to empty all fields.

4.5.2.3.2. Delete Lecturer Page

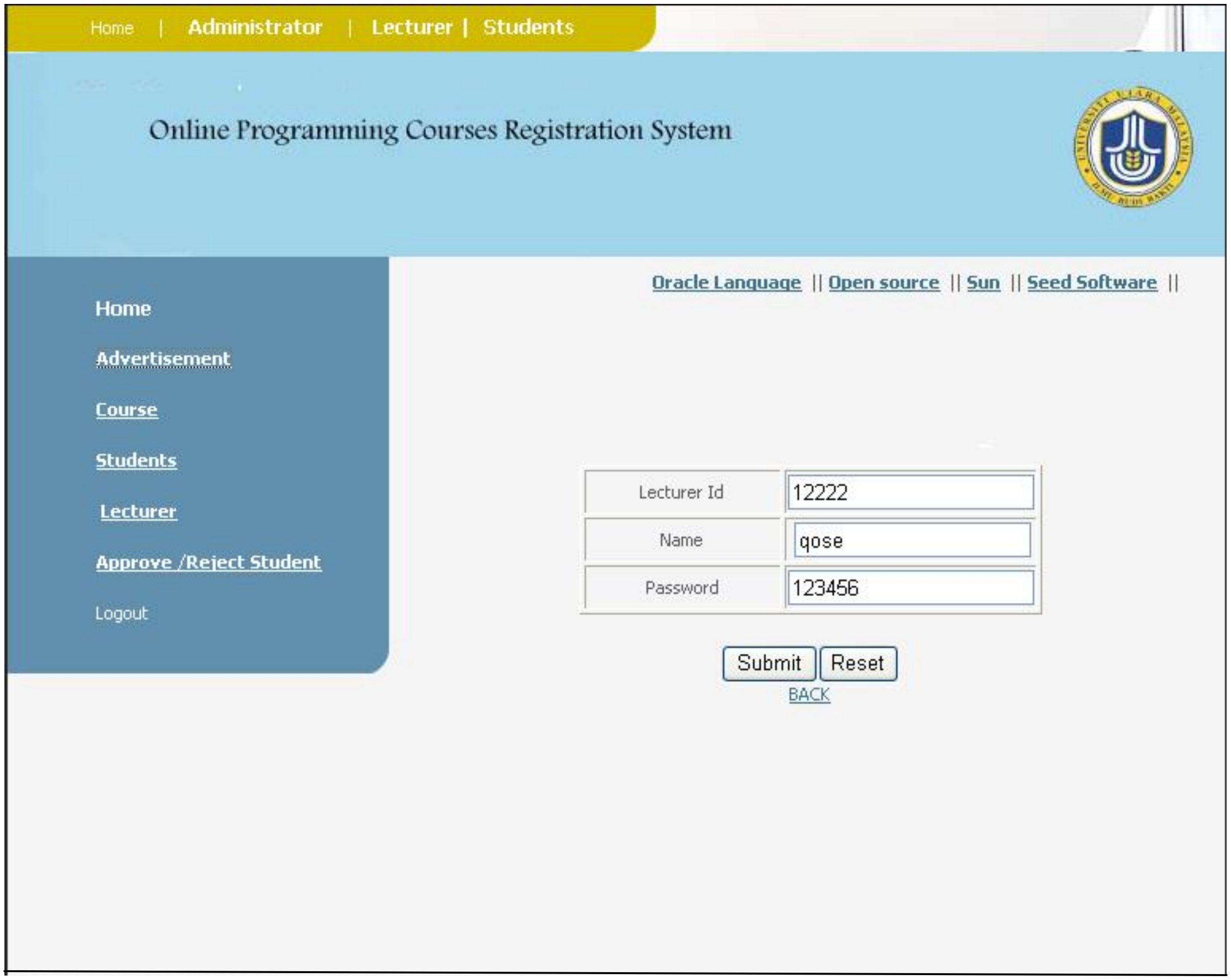

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Advertisement
Course
Students
Lecturer
Approve /Reject Student
Logout
Submit

Are you sure you want delete? ● Yes ○ No

Figure 4.37: Delete Lecturer Page

After press "Delete" Link in the Course Page, the prototype move to Delete Programming Course Page. The System will ask the administrator to confirm Delete process. The administrator will press "Yes" option if he wants delete Programming Course information. And then he will press submit button. The system will delete Programming Course record from Course database. Or He can Press "No" option to cancel Delete process and then he will press submit button to go back to Lecturer page.

4.5.2.3.3.Update Lecturer Information Page

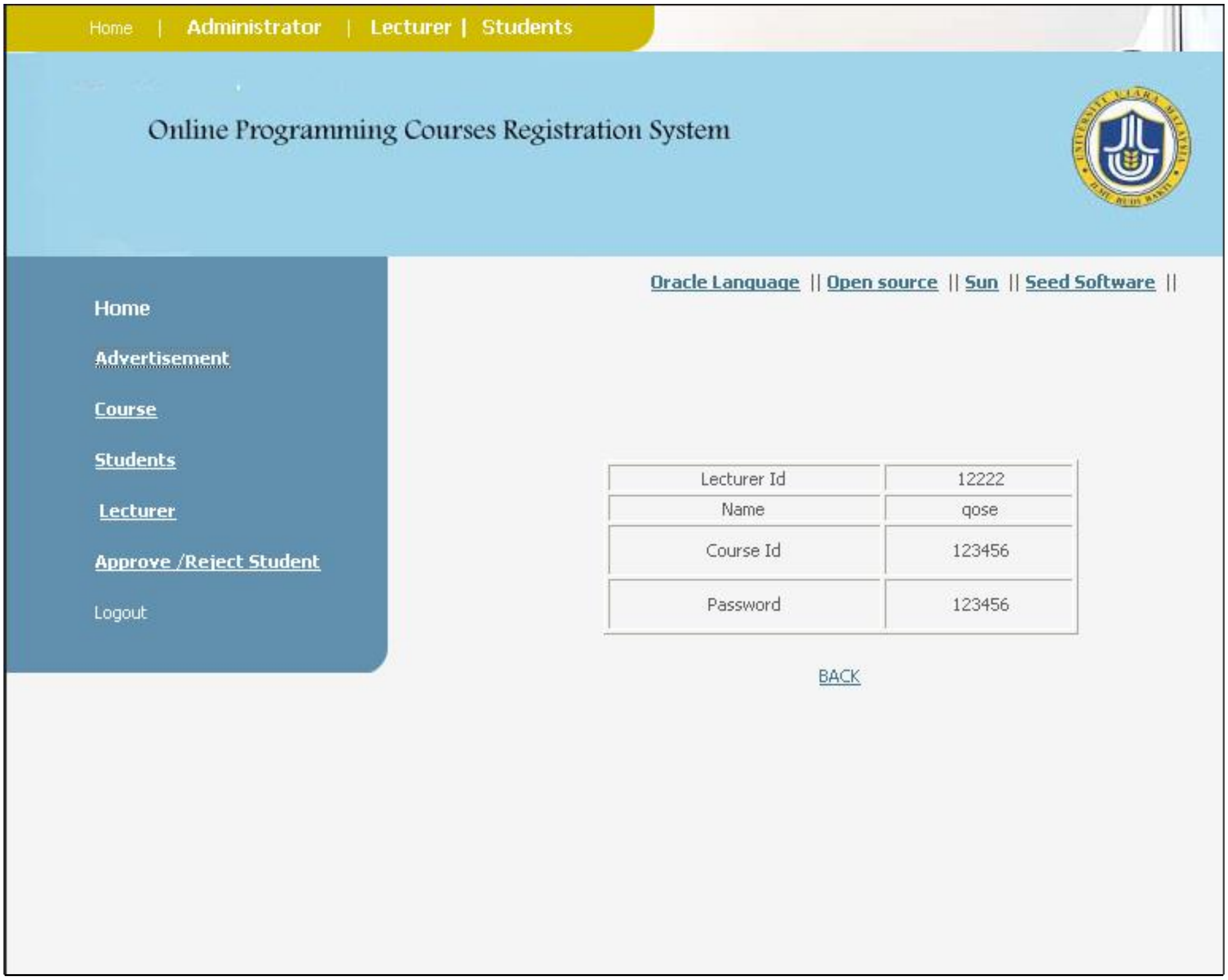

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Advertisement
Course
Students
Lecturer
Approve /Reject Student
Logout
Submit
Reset
BACK

Lecturer Id	12222
Name	qose
Password	123456

Figure 4.38: Update Lecturer Information Page

After press "Update" Link in the Lecturer Page, the prototype moves to Update Lecturer Information Page. After administrator finishes update his/her lecturer information, then he will press submit button. And the system will update old lecturer information to new lecturer information. Or He can Press Back link to go back to lecturer page or Reset button to empty all fields.

4.5.2.3.4. View Lecturer Information Page

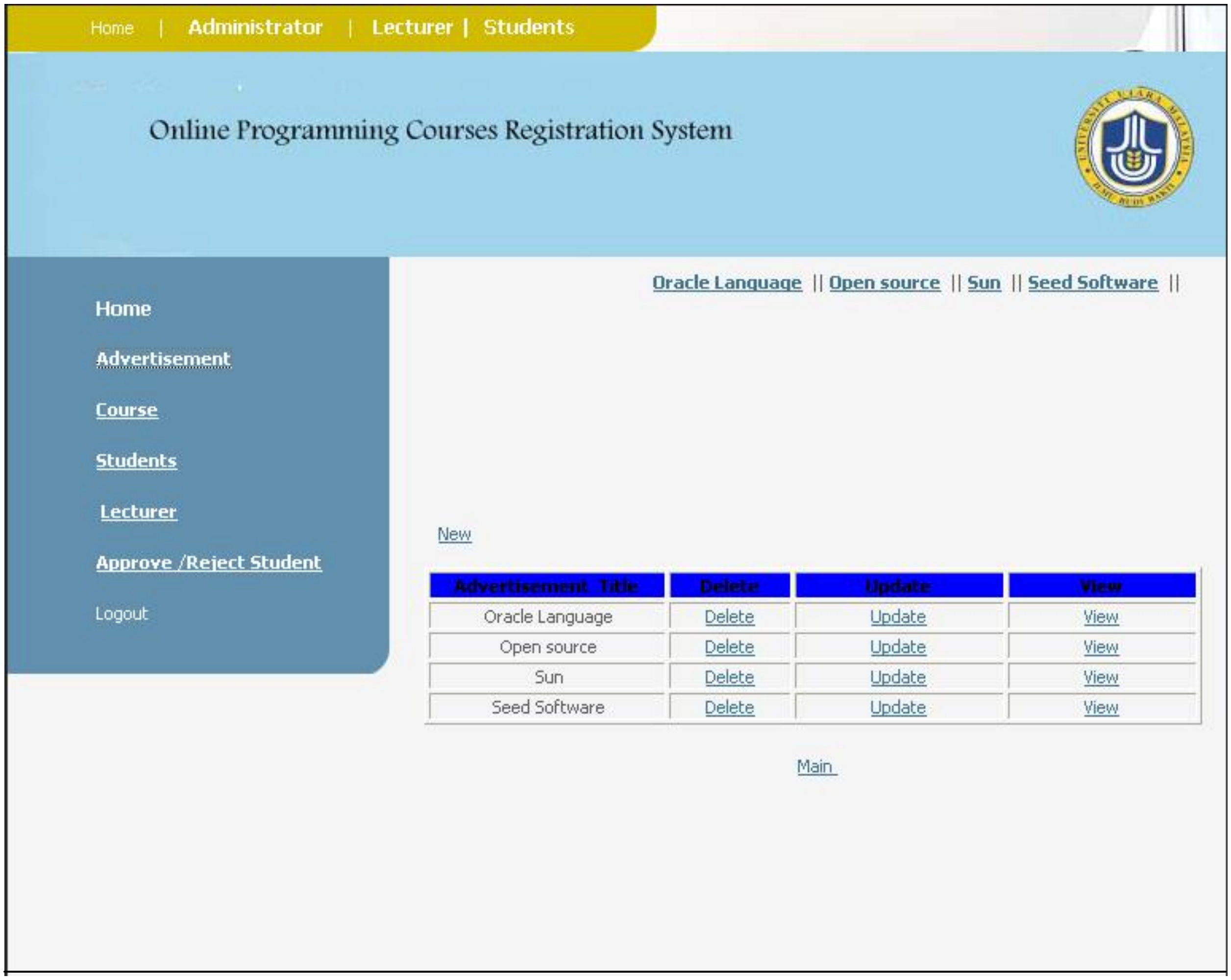

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Advertisement
Course
Students
Lecturer
Approve /Reject Student
Logout
BACK

Lecturer Id	12222
Name	qose
Course Id	123456
Password	123456

Figure 4.39: View Lecturer Information Page

After press “View” link in the Lecturer Page, the prototype moves to View Lecturer Information Page. And the system will fetch lecturer information from Lecturer database. After that he can Press Back link to go back to Lecturer page.

4.5.2.4.Advertisement Page


Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Advertisement
Course
Students
Lecturer
Approve /Reject Student
Logout
New
Main

Advertisement Title	Delete	Update	View
Oracle Language	Delete	Update	View
Open source	Delete	Update	View
Sun	Delete	Update	View
Seed Software	Delete	Update	View

Figure 4.40: Advertisement Page

After press "Advertisement" link in the Main Administrators Page, the prototype moves to Advertisement page. He can press "New" link to add new Advertisement or "Update" link to update Advertisement Information or "Delete" link to delete Advertisement or "View" link to view Advertisement information. And the administrator can Press main to go back to Main Administrators page or logout to exist from this page and go to Home page.

4.5.2.4.1.Add New Advertisement Page

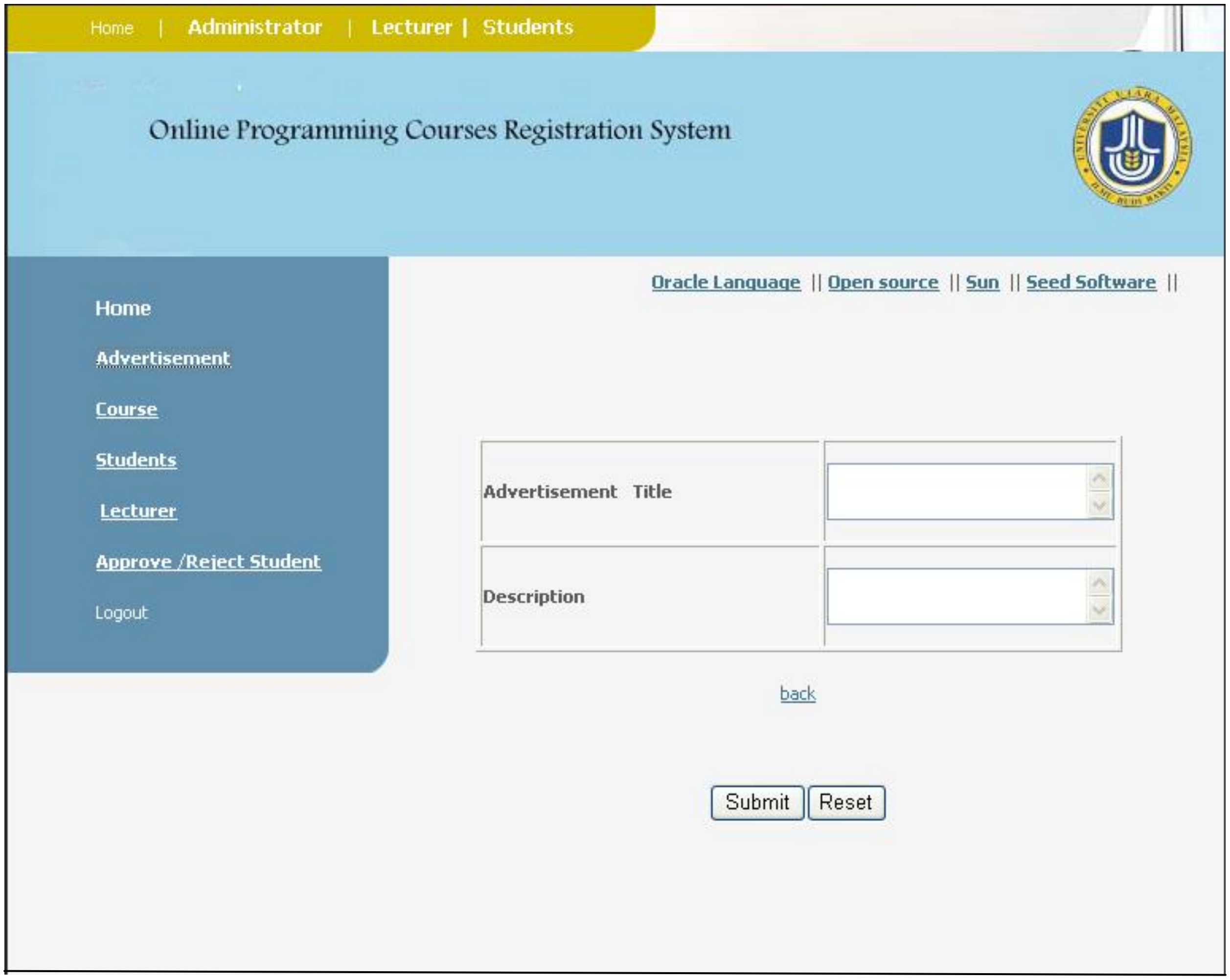
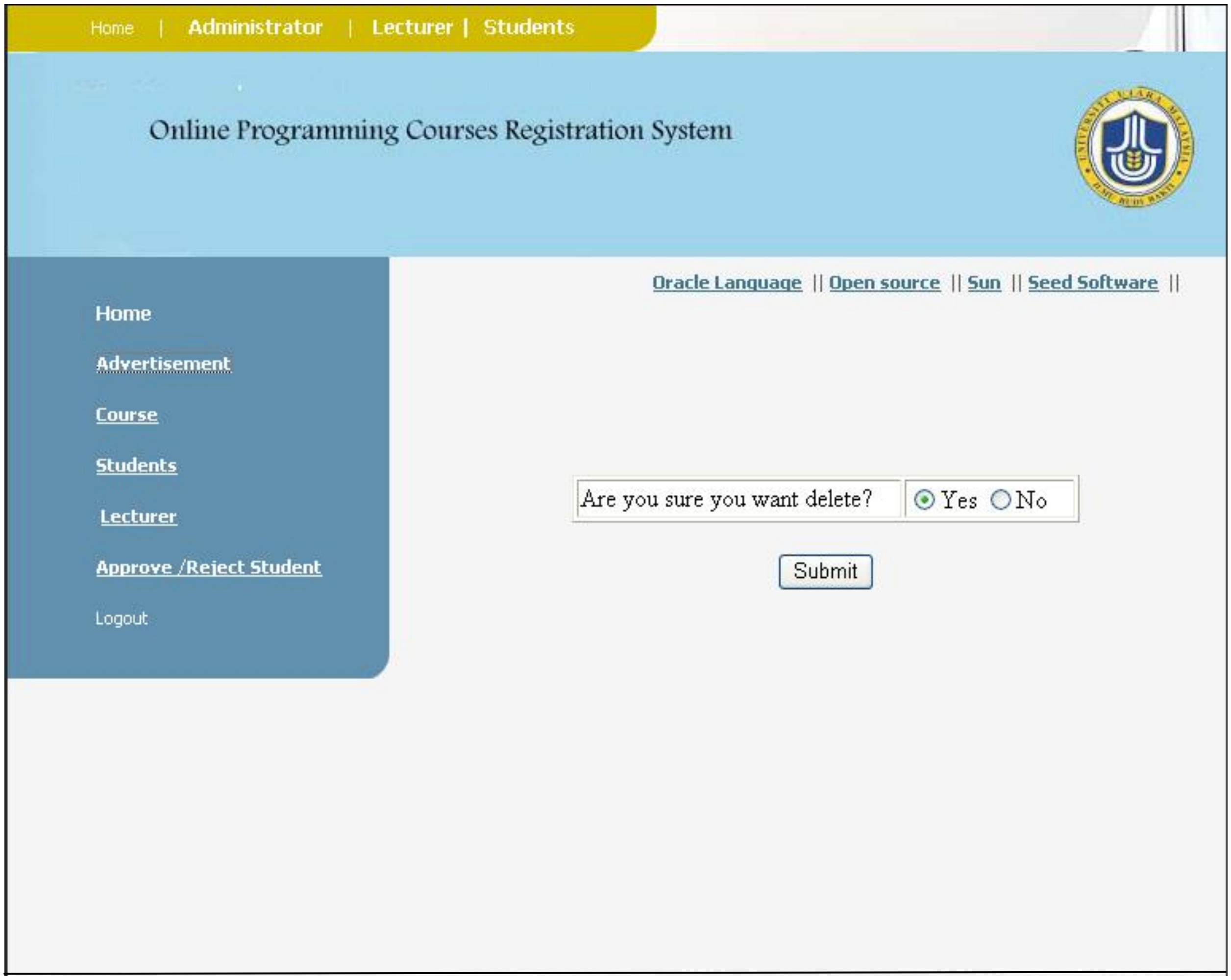

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Advertisement
Course
Students
Lecturer
Approve /Reject Student
Logout
Advertisement Title
Description
back
Submit
Reset

Figure 4.41: Add New Advertisement Page

After press “New” link in the Advertisement Page, the prototype move to Add Advertisement Page. The administrator will insert information for new Advertisement such as Advertisement Title and Description. And then he will press Submit Button. The System will create new record and store this information into Advertisement database. He can Press Back link to go back to Advertisement page or Reset button to empty all fields.

4.5.2.4.2.Delete Advertisement Page

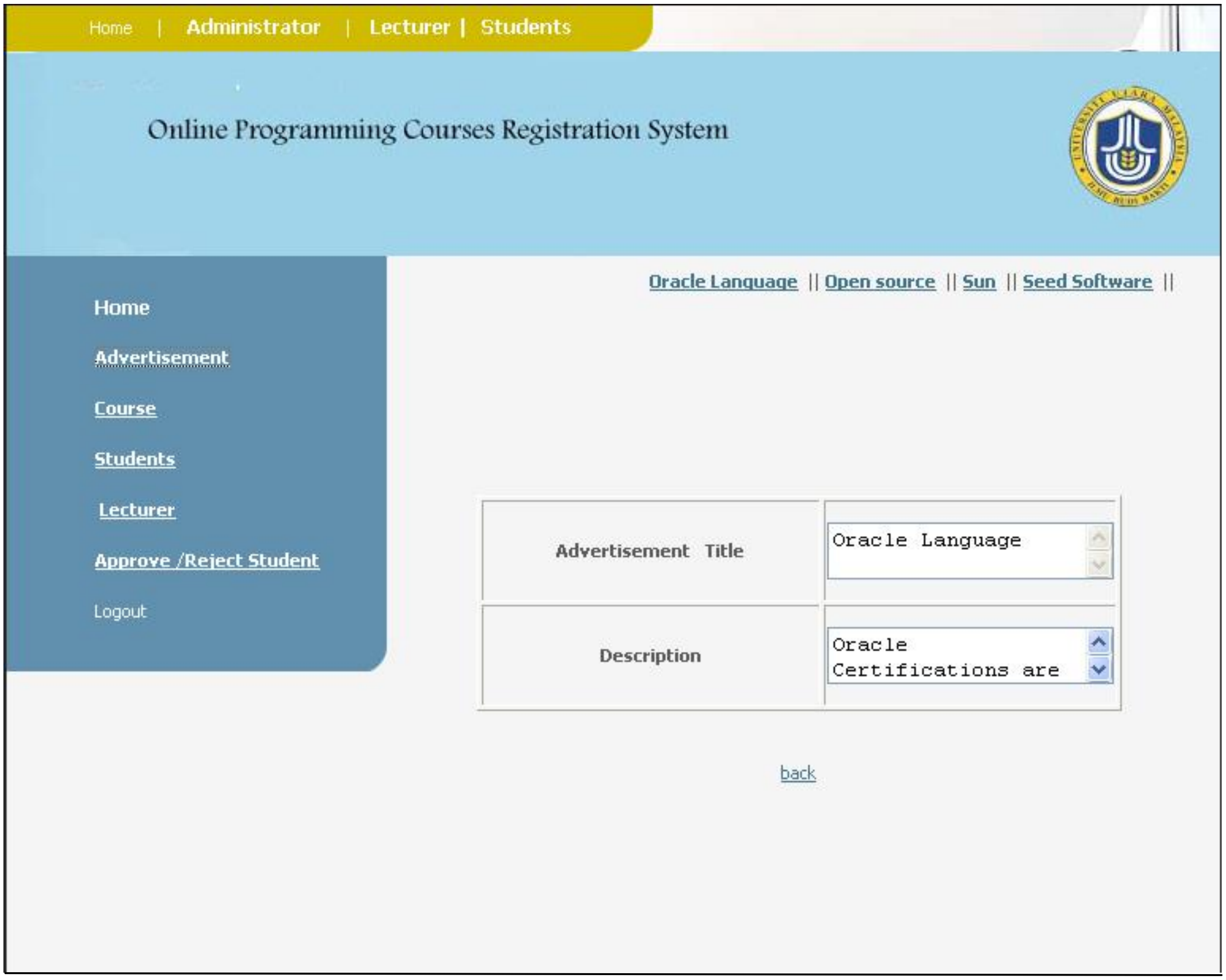

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Advertisement
Course
Students
Lecturer
Approve /Reject Student
Logout
Submit

Are you sure you want delete? ● Yes ○ No

Figure 4.42: Delete Advertisement Page

After press “Delete” Link in the Advertisement Page, the prototype move to Delete Advertisement Page. The System will ask the administrator to confirm Delete process. The administrator will press “Yes” option if he wants delete Advertisement information. And then he will press submit button. The system will delete Advertisement record from Advertisement database. Or He can Press “No” option to cancel Delete process and then he will press submit button to go back to Advertisement page.

4.5.2.4.3. Update Advertisement information Page

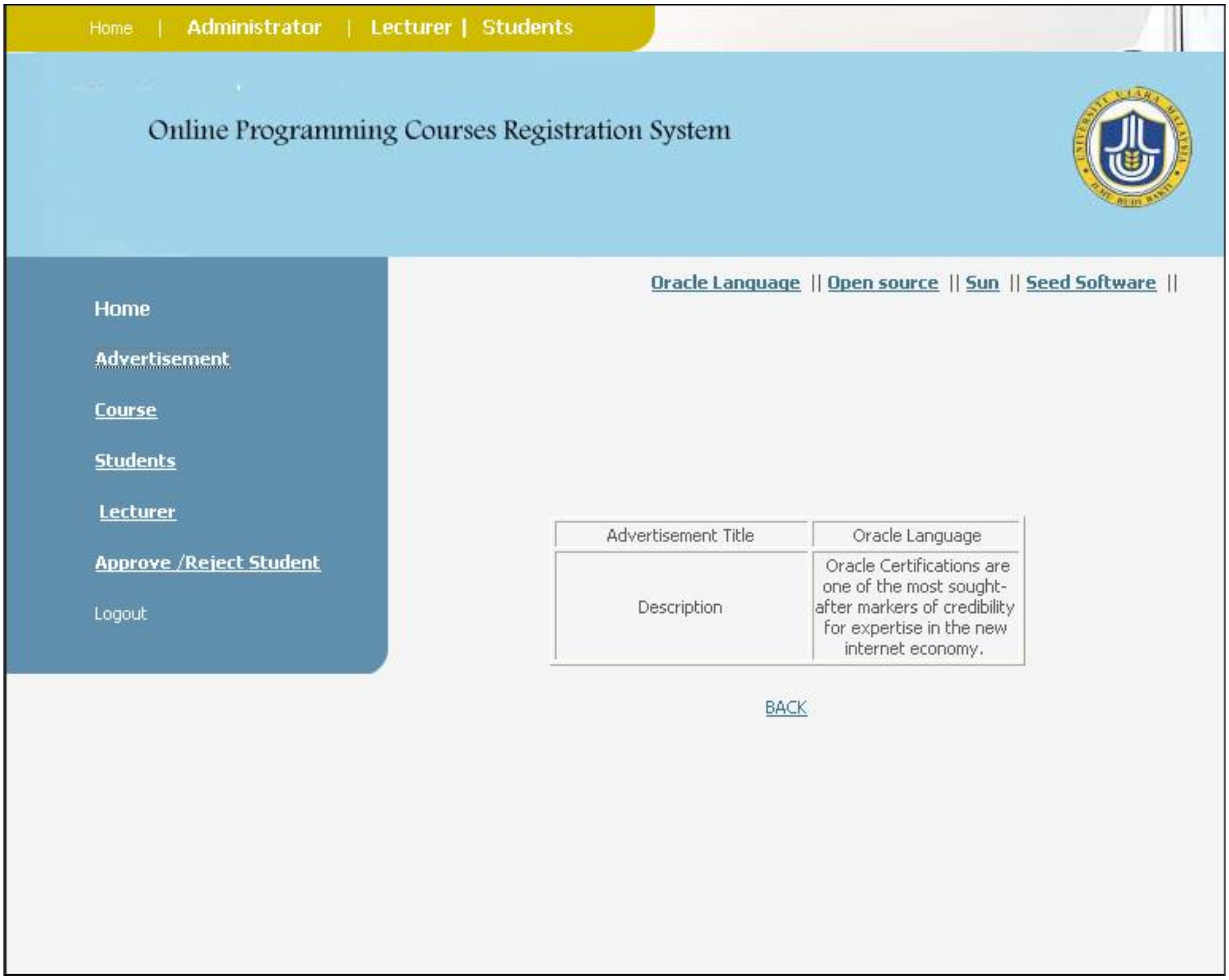

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Advertisement
Course
Students
Lecturer
Approve /Reject Student
Logout
back

Advertisement Title	Oracle Language
Description	Oracle Certifications are

Figure 4.43: Update Advertisement Information Page

After press “Update” Link in the Advertisement Page, the prototype moves to Update Advertisement Information Page. After administrator finishes update his/her Advertisement information, then he will press submit button. And the system will update old Advertisement information to new Advertisement information. Or He can Press Back link to go back to Advertisement page or Reset button to empty all fields.

4.5.2.4.4. View Advertisement Information Page

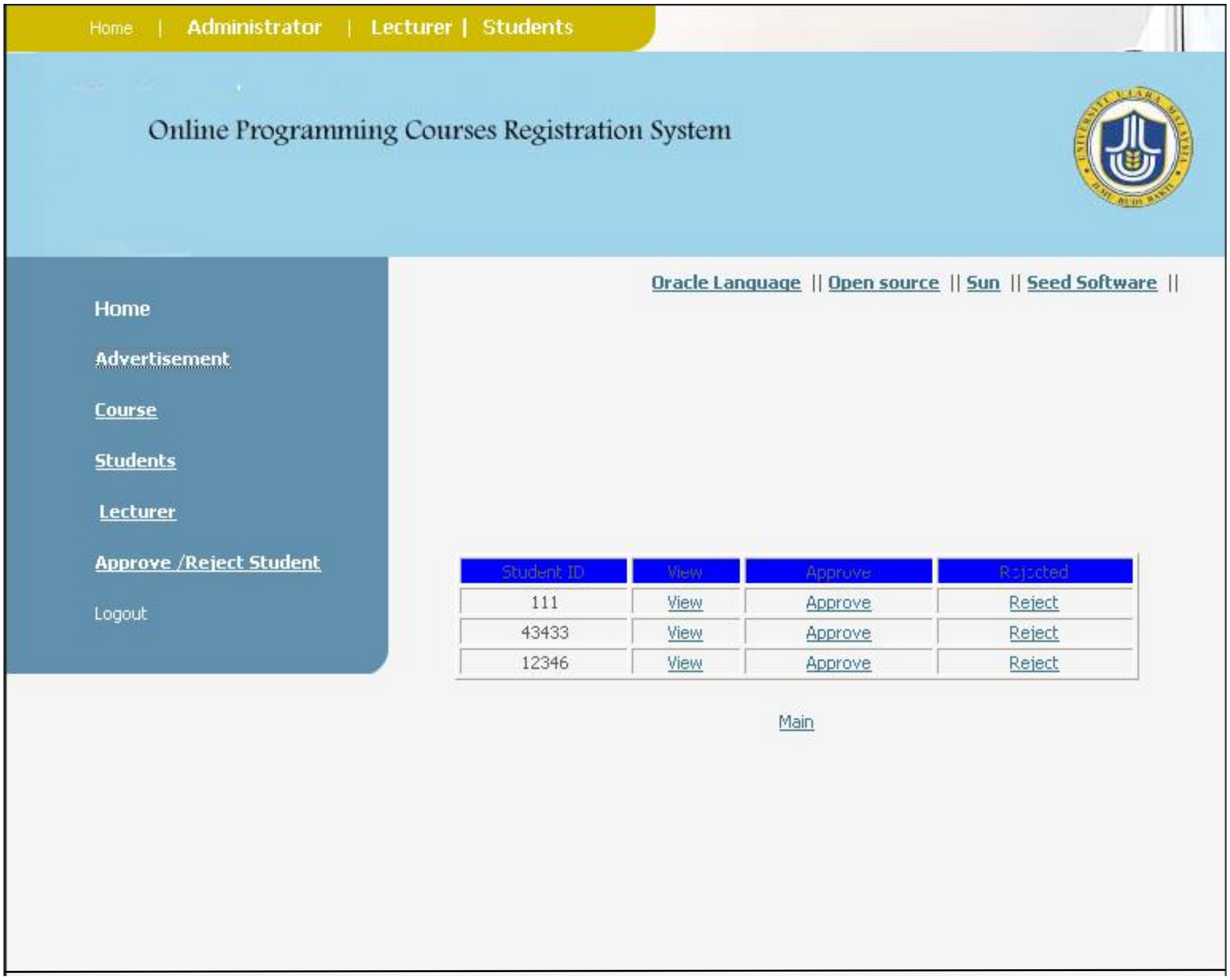

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Advertisement
Course
Students
Lecturer
Approve /Reject Student
Logout
BACK

Advertisement Title	Oracle Language
Description	Oracle Certifications are one of the most sought-after markers of credibility for expertise in the new internet economy.

Figure 4.44: View Advertisement Information Page

After press “View” link in the Advertisement Page, the prototype moves to View Advertisement Information Page. And the system will fetch Advertisement information from Advertisement database. After that he can Press Back link to go back to Advertisement page.

4.5.2.5.Approval/Reject Student Page

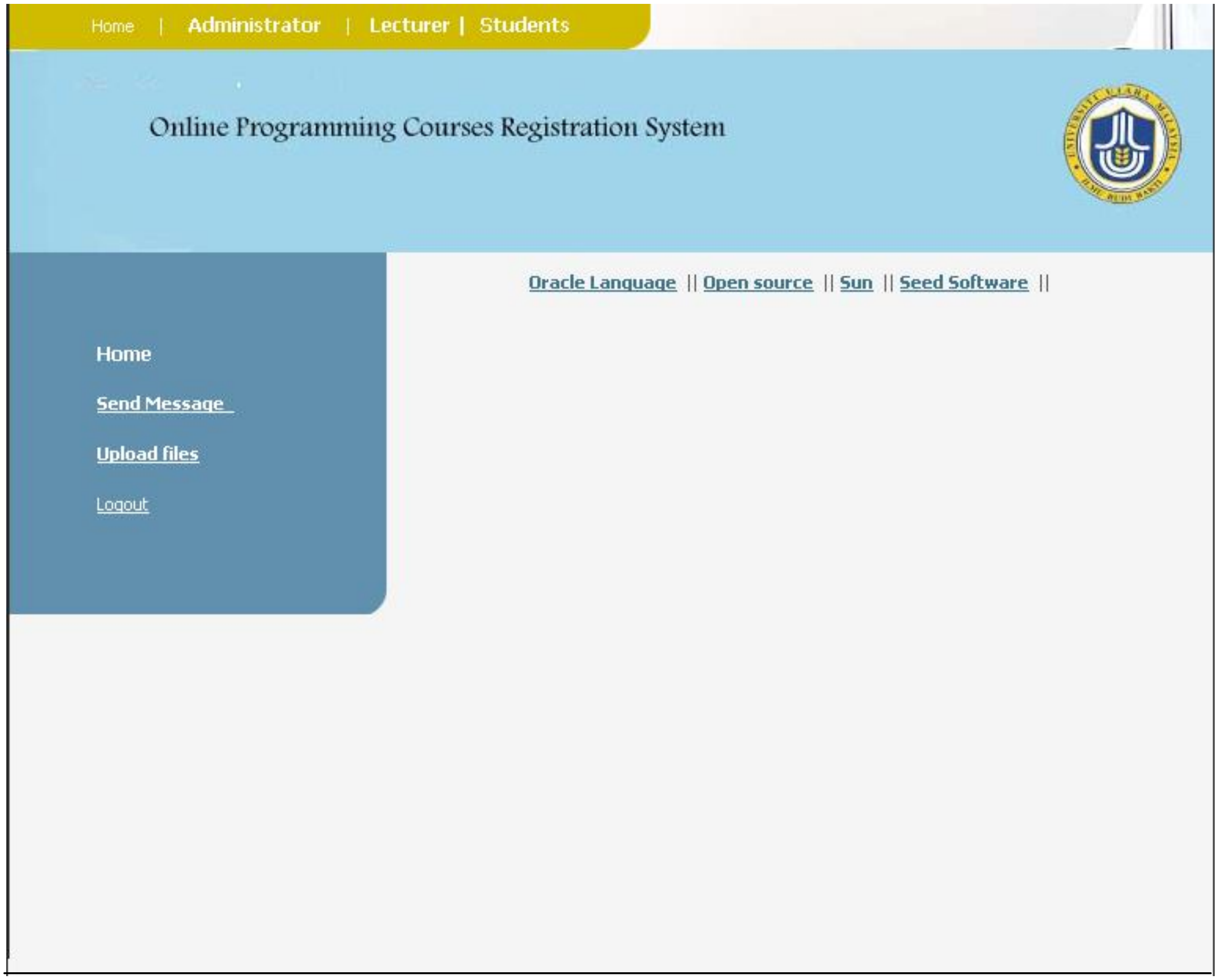

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Advertisement
Course
Students
Lecturer
Approve /Reject Student
Logout
Main

Student ID	View	Approve	Rejected
111	View	Approve	Reject
43433	View	Approve	Reject
12346	View	Approve	Reject

Figure 4.45: Approval/Reject Student Page

After press the Approval/Reject Students Link in the Main Administrators Page, the prototype move to Approval/Reject Student page to grant Approve or Reject for students applications. The system will change student status to approve if the administrator press “Approve” and the student status will reject if he press “Reject”. He can Press main to go back to Main Administrators page or logout to exist from this page and go to Home page.

4.5.3. Main Lecturer Page



Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Send Message
Upload files
Logout

Figure 4.46: Main Lecturer Page

Main Lecturer Page displays functionality requirement (Primary functions) for Lecturer are send message to his/her students, Upload Files. The Lecturer may press Home Link to go back to Home Page or the Lecturer may press Log Out button to exist from his/her main page.

4.5.3.1. Send Message Page

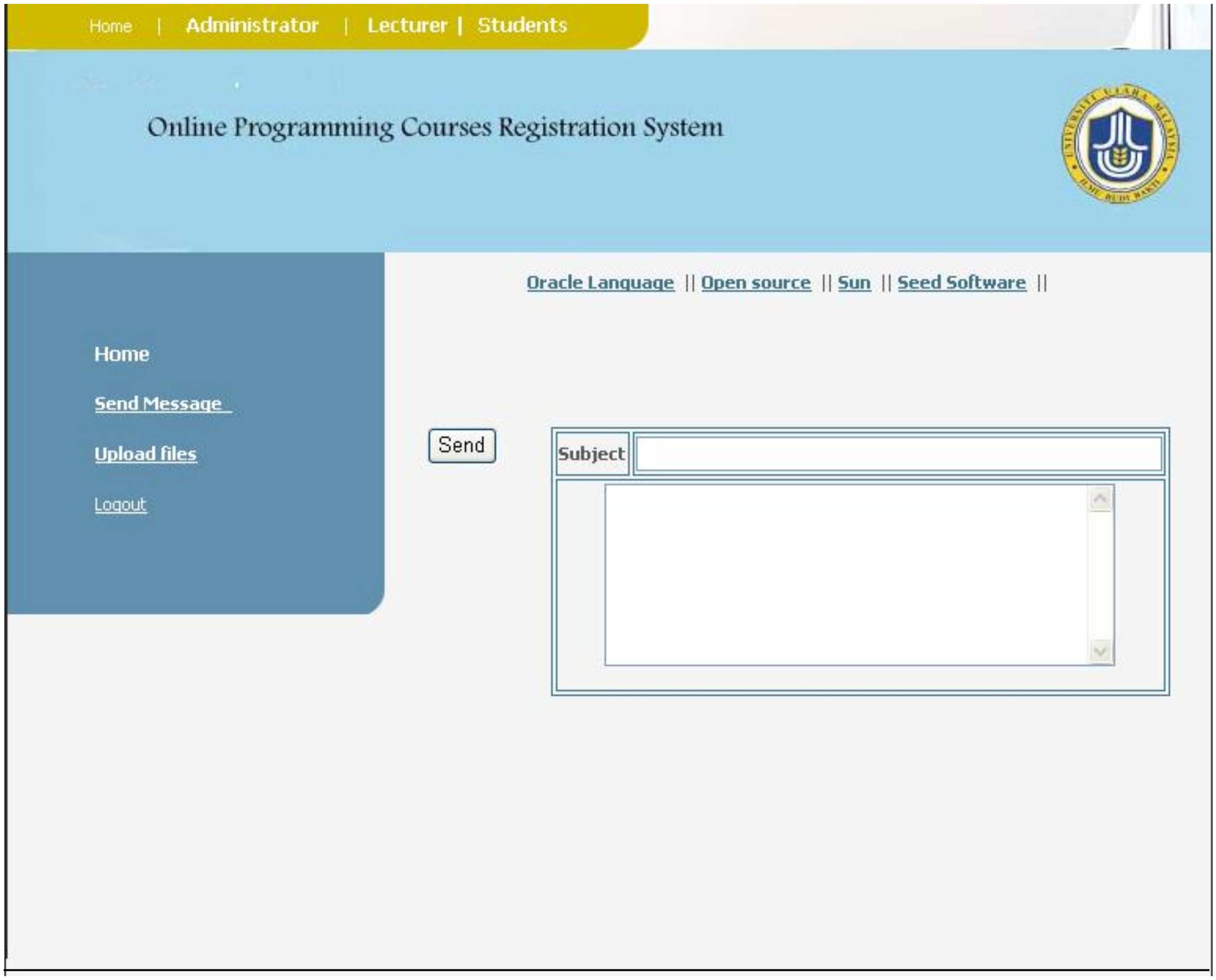

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Send Message
Upload files
Logout
Send
Subject

Figure 4.47: Send Message Page

After press “Send Message” link in the Lecturer Page, the prototype moves to Send Message Page. The Lecturer can write the message subject and description it which he will send to all his/her Students, and then he will press Send button. And the system will create record and store Message information into Email database. After that he can Press Back link to go back to Main Lecturer Page.

4.5.3.2.Upload Files Page

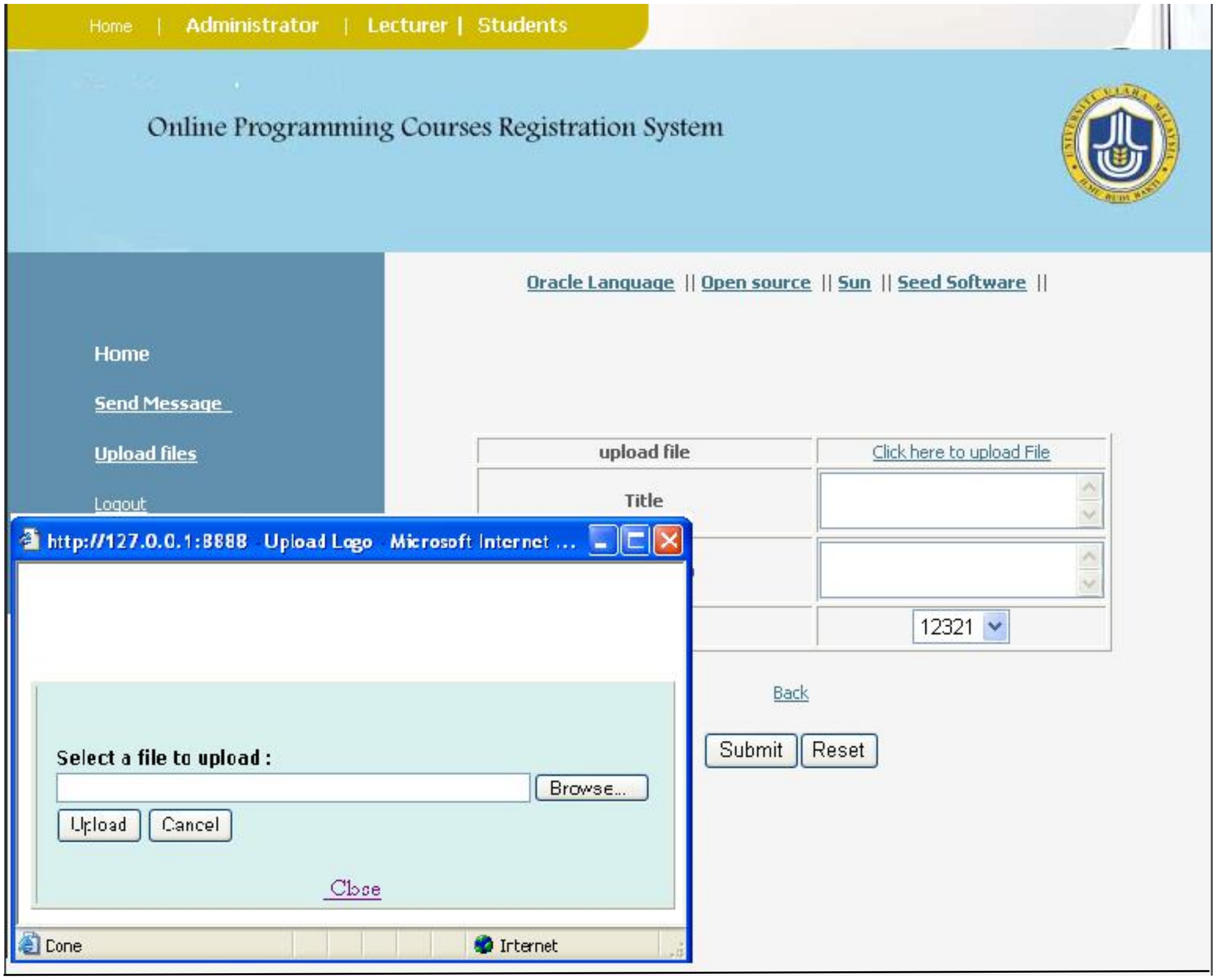

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Send Message
Upload files
Logout
upload file
Click here to upload File
Title
12321
Back
Submit
Reset
http://127.0.0.1:8888 Upload Logo Microsoft Internet ...
Select a file to upload :
Browse...
Upload
Cancel
Close
Done
Internet

Figure 4.48: Upload Files Page

After press Upload Files button in the Lecturer Page, the prototype moves to Upload Files page. The Lecturer will press "click here to upload file" Link to upload files such as PFD or Doc to his/her students. He also can write the file title and description it and select course Id which he wants upload file for it. And then he will press Send button. And the system will create record and store File information into Upload database. After that he can Press Back link to go back to Lecturer page.

4.5.4. Main Student Page

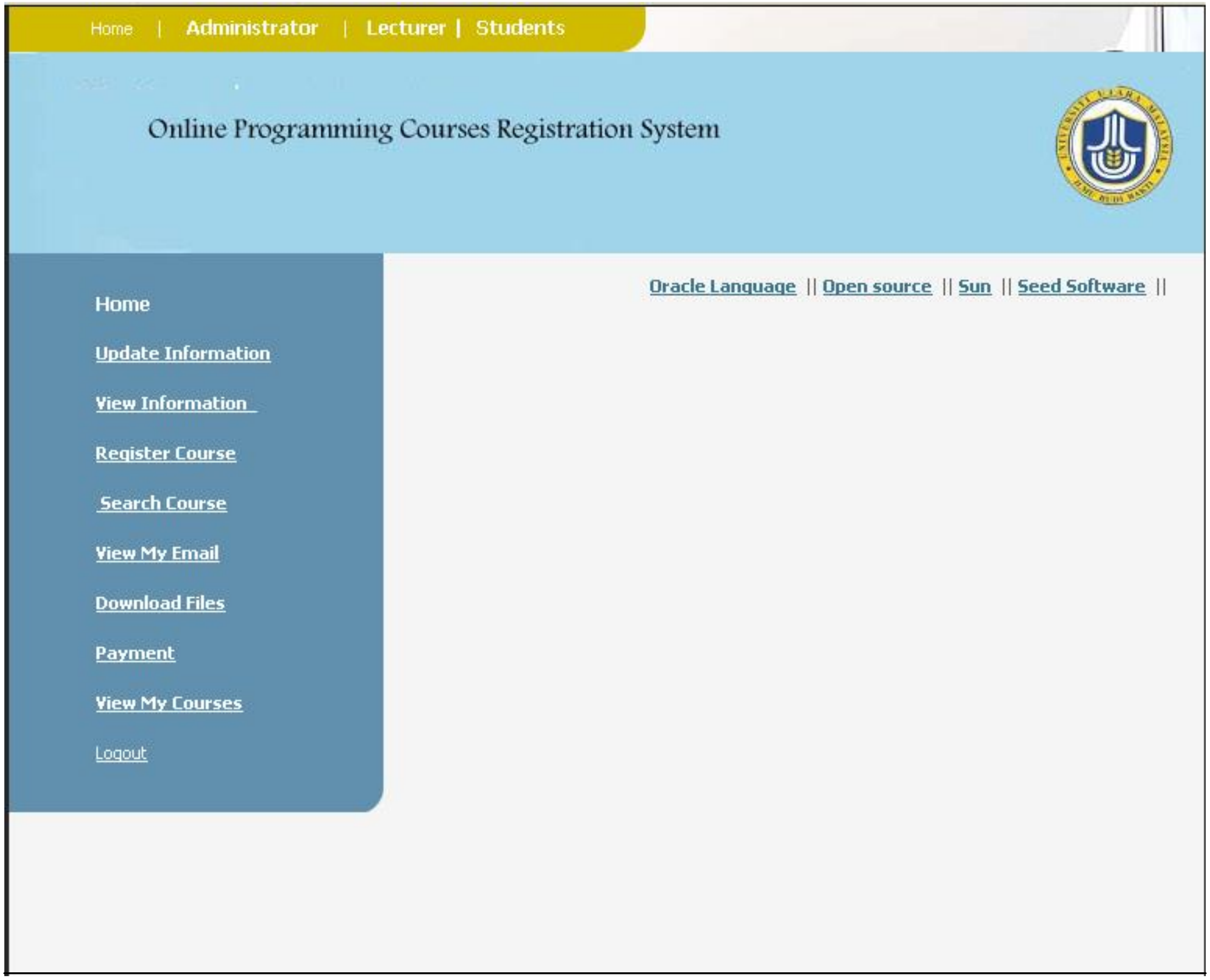
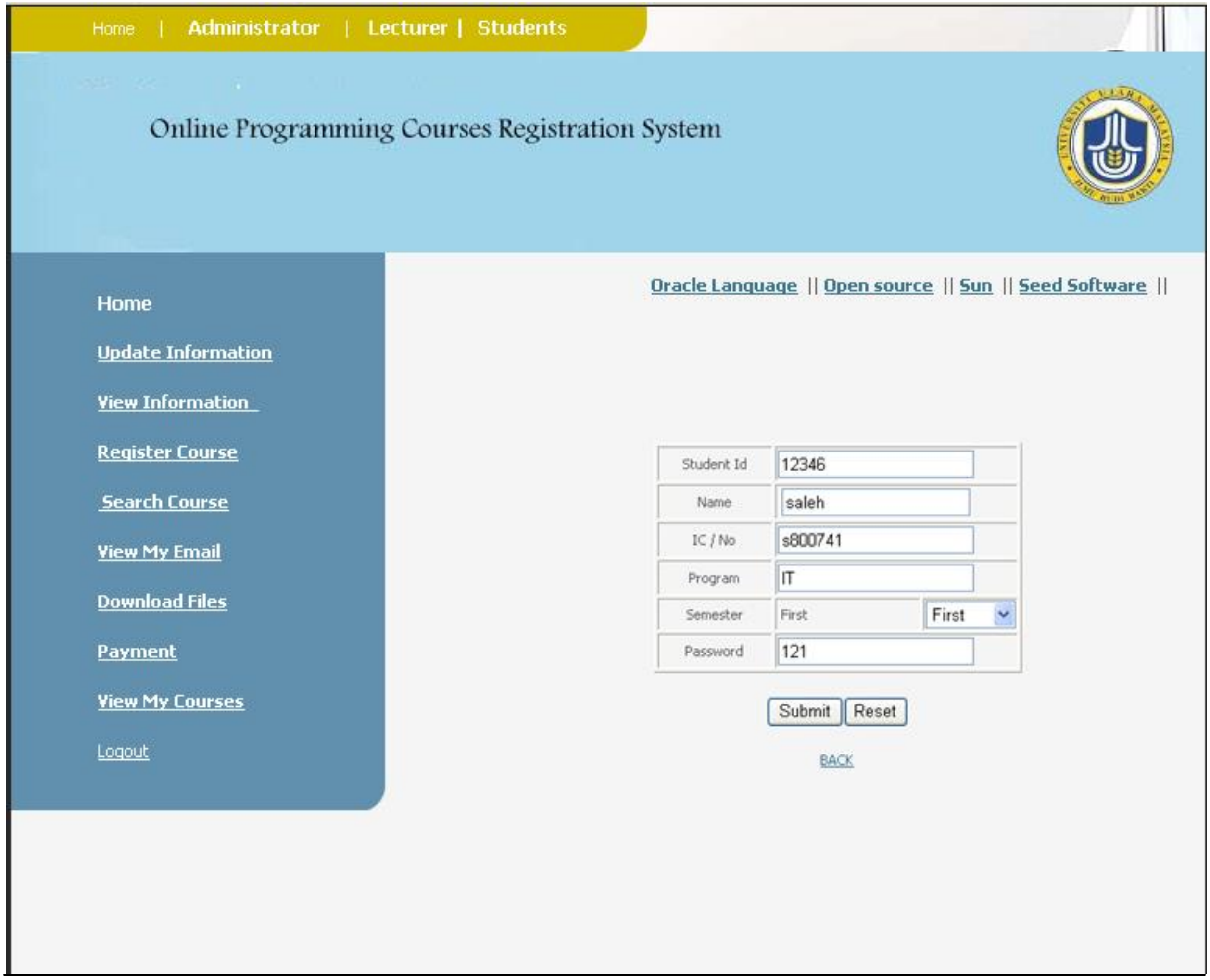

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Update Information
View Information
Register Course
Search Course
View My Email
Download Files
Payment
View My Courses
Logout

Figure 4.49: Main Student Page

Main Student Page displays functionality requirement (Primary functions) for Student are Update Information, view Information, Register Course, Search Course, View My Email, Download Files, View My course, Payment. The Student may press Home Page button to go back to Home Page or the Student may press Log Out button to exist from his/her account.

4.5.4.1.Update Student Information Page

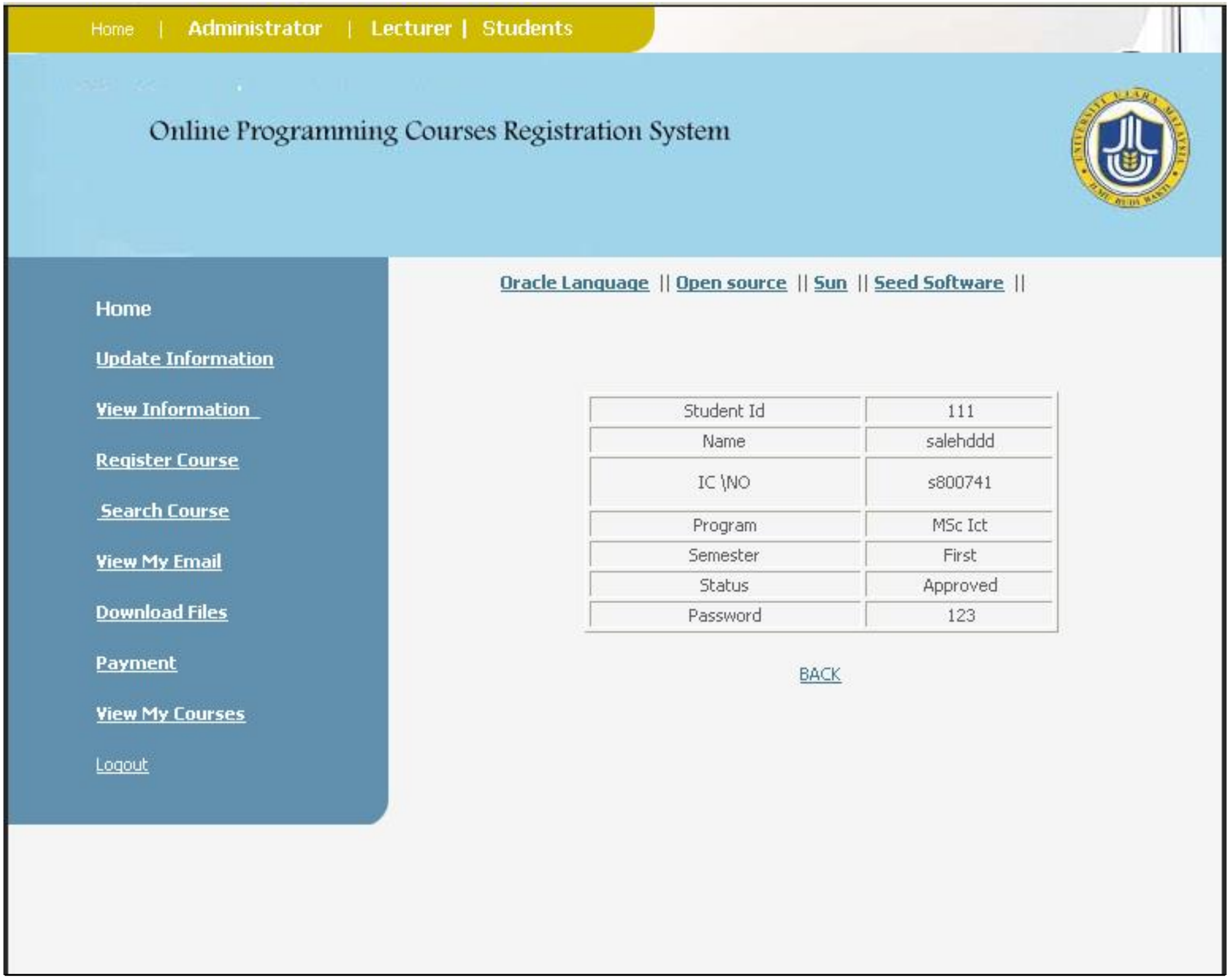

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Home
Update Information
View Information
Register Course
Search Course
View My Email
Download Files
Payment
View My Courses
Logout
Oracle Language || Open source || Sun || Seed Software ||
Submit
Reset
BACK

Student Id	12346	
Name	saleh	
IC / No	s800741	
Program	IT	
Semester	First	First
Password	121	

Figure 4.50: Update Student Information Page

After press “Update Information” Link in the Main Student Page, the prototype moves to Update Student Information Page. After the student finishes update his/her student information, the student will press submit button. And the system will update old student information to new student information. Or He can Press Back link to go back to Main Student page or Reset button to empty all fields.

4.5.4.2.View Student Information Page


Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Update Information
View Information
Register Course
Search Course
View My Email
Download Files
Payment
View My Courses
Logout
BACK

Student Id	111
Name	salehddd
IC \NO	s800741
Program	MSc Ict
Semester	First
Status	Approved
Password	123

Figure 4.51: View Student Information Page

After press “View Information” link in the Main Student Page, the prototype moves to View Student Information Page. And the system will fetch student information from student database. After that he can Press Back link to go back to Main Student page.

4.5.4.3.Register Programming course Page

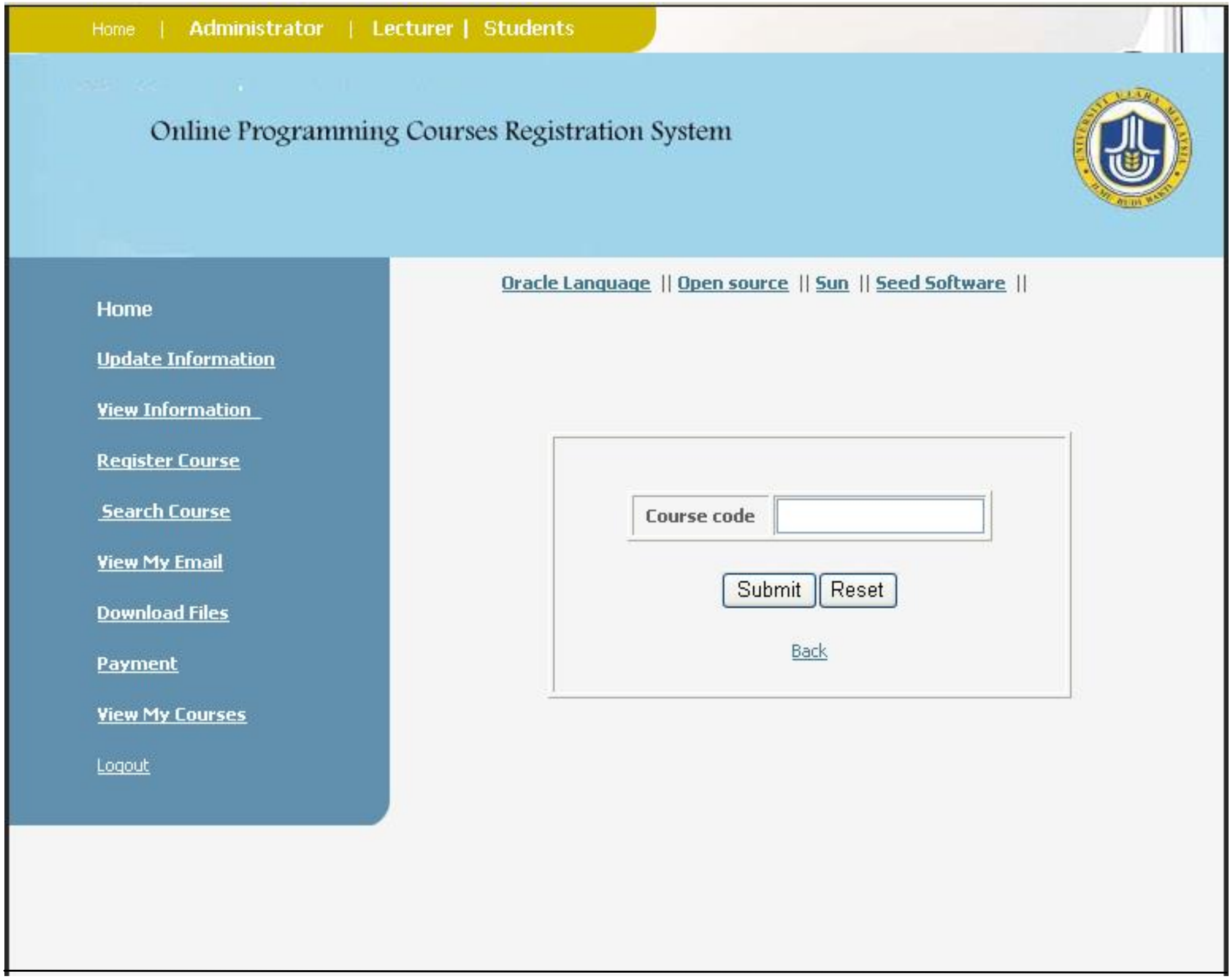
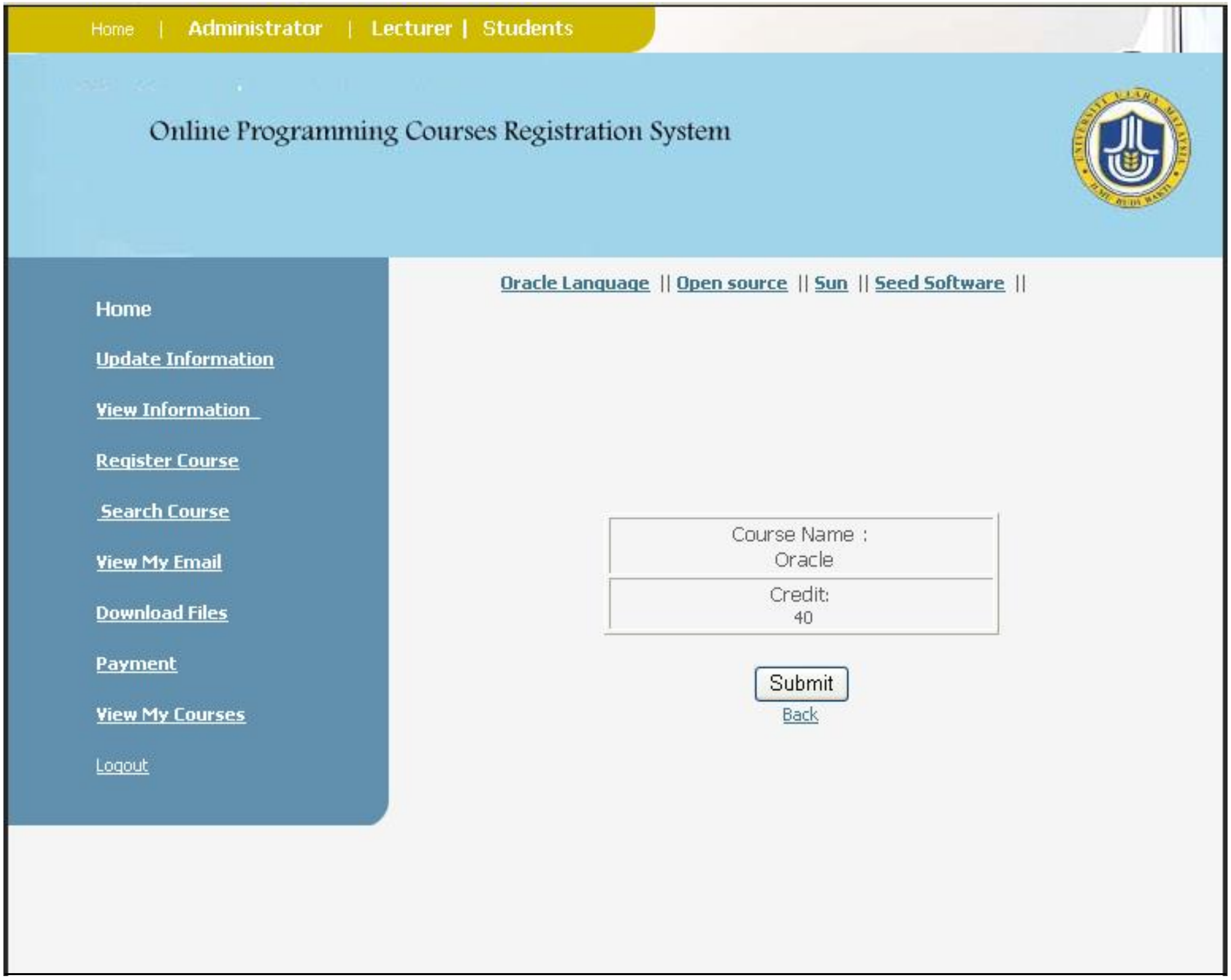


Figure 4.52: Register Programming Course Page

After press “Register Course “link in the Main Student Page, the prototype move to Register Programming Courses Page. Through this page the student will insert course code to course that he want register it. The student will press submit button. And the system will check course code if course code is valid then the prototype will move to next page (View Register Programming Course Page). The Student can Press Back link to go back to Main Student page.

4.5.4.3.1. View Register Programming Course Information Page


Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Update Information
View Information
Register Course
Search Course
View My Email
Download Files
Payment
View My Courses
Logout
Submit
Back

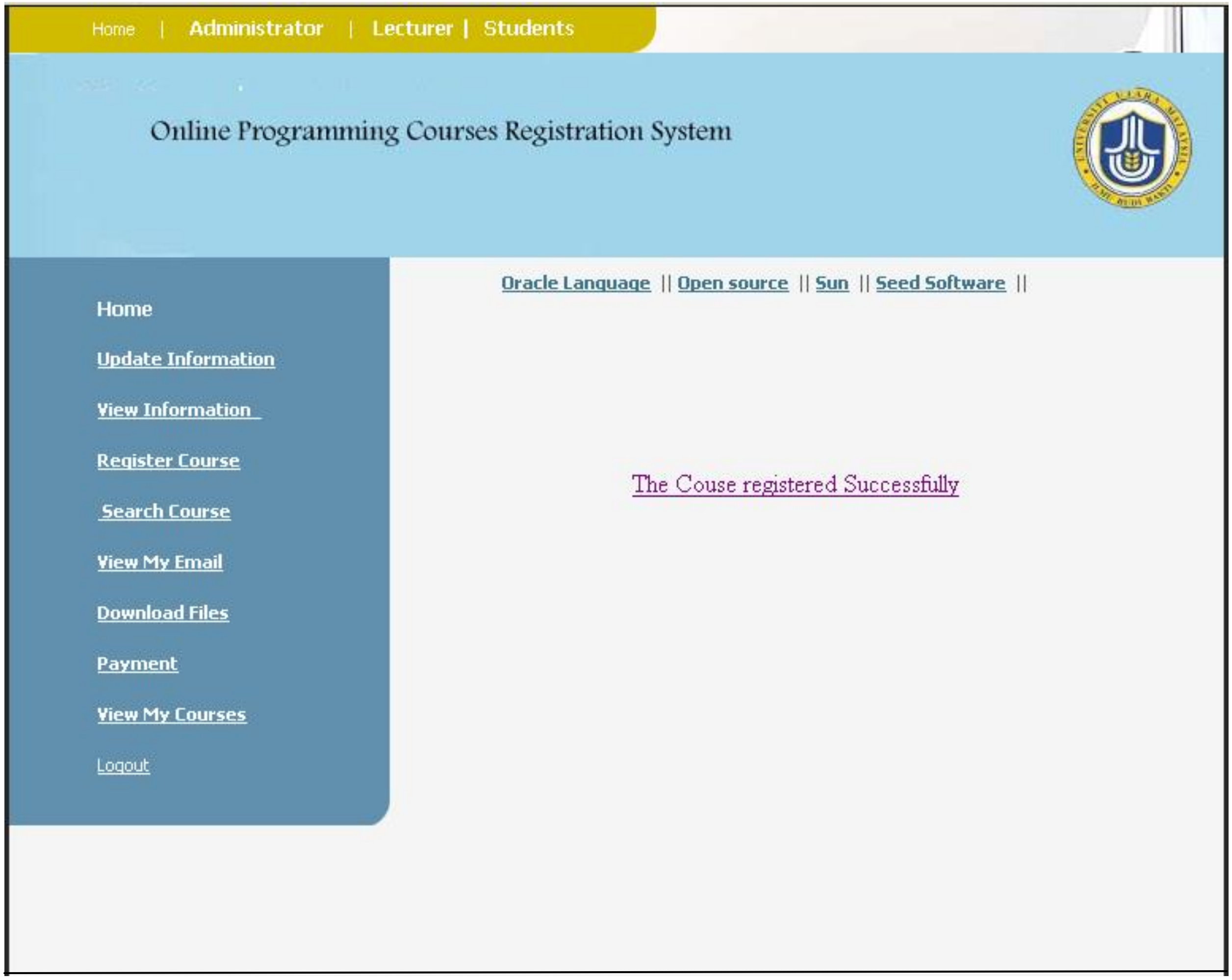
Course Name :
Oracle

Credit:
40

Figure 4.53: View Register Programming Course Information Page

After press “Submit” button in the Register Course Page, the prototype moves to View Register Programming Course Information page to view course information for course which he wants register it such as Course name and Credit. And then the student press submit button to continuous Register course process. After that Student can Press Back link to go back to Main student page.

4.5.4.3.1.1.Message Register Page


Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Update Information
View Information
Register Course
Search Course
View My Email
Download Files
Payment
View My Courses
Logout

The Couse registered Successfully

Figure 4.54: Message Register Page

After press submit button in the view register programming course information Page, the prototype move to massage Register page to tell student register course has been successfully. After that student can press the register course successfully link to go back to Main student Page.

4.5.4.4.Search Programming Course Page

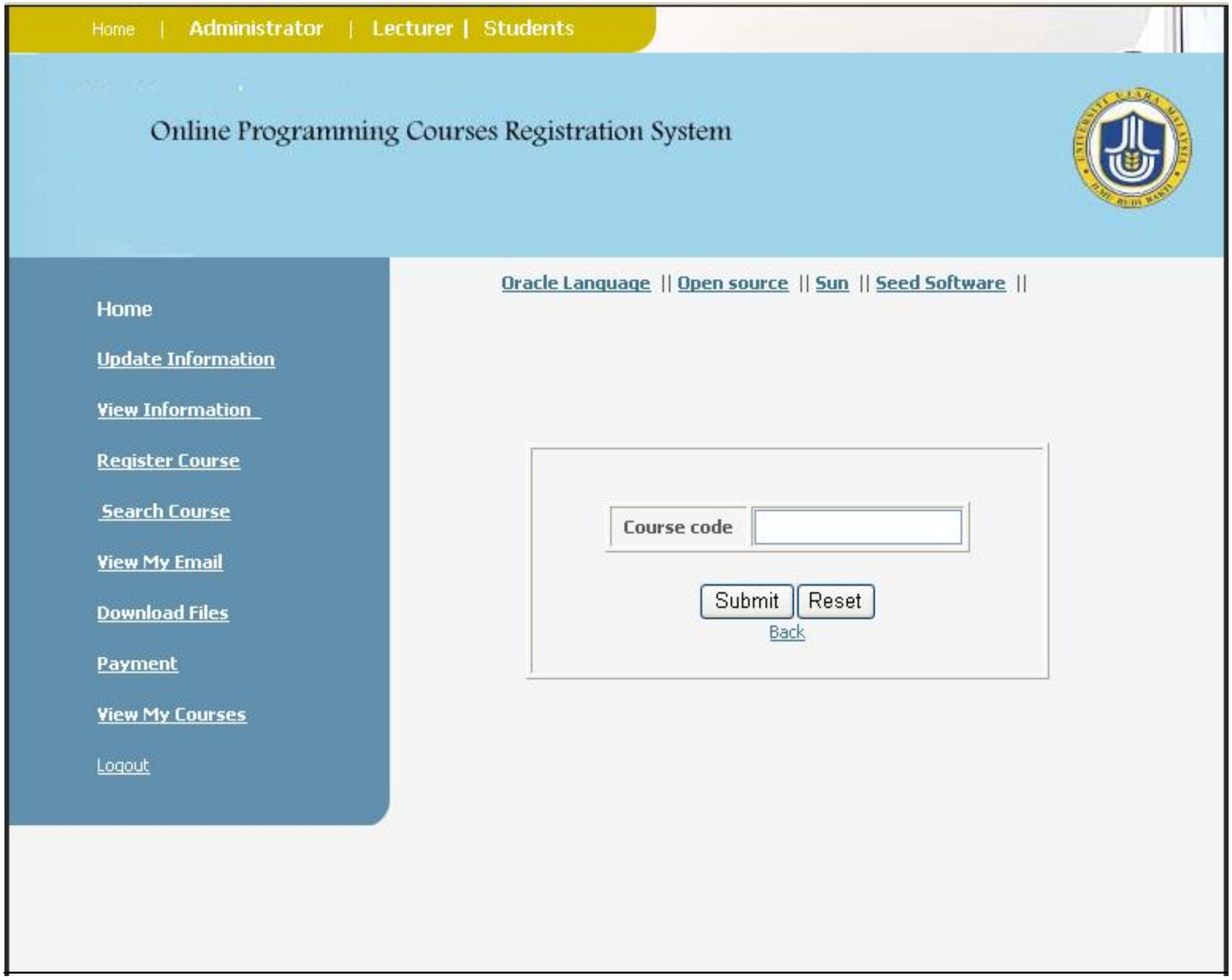
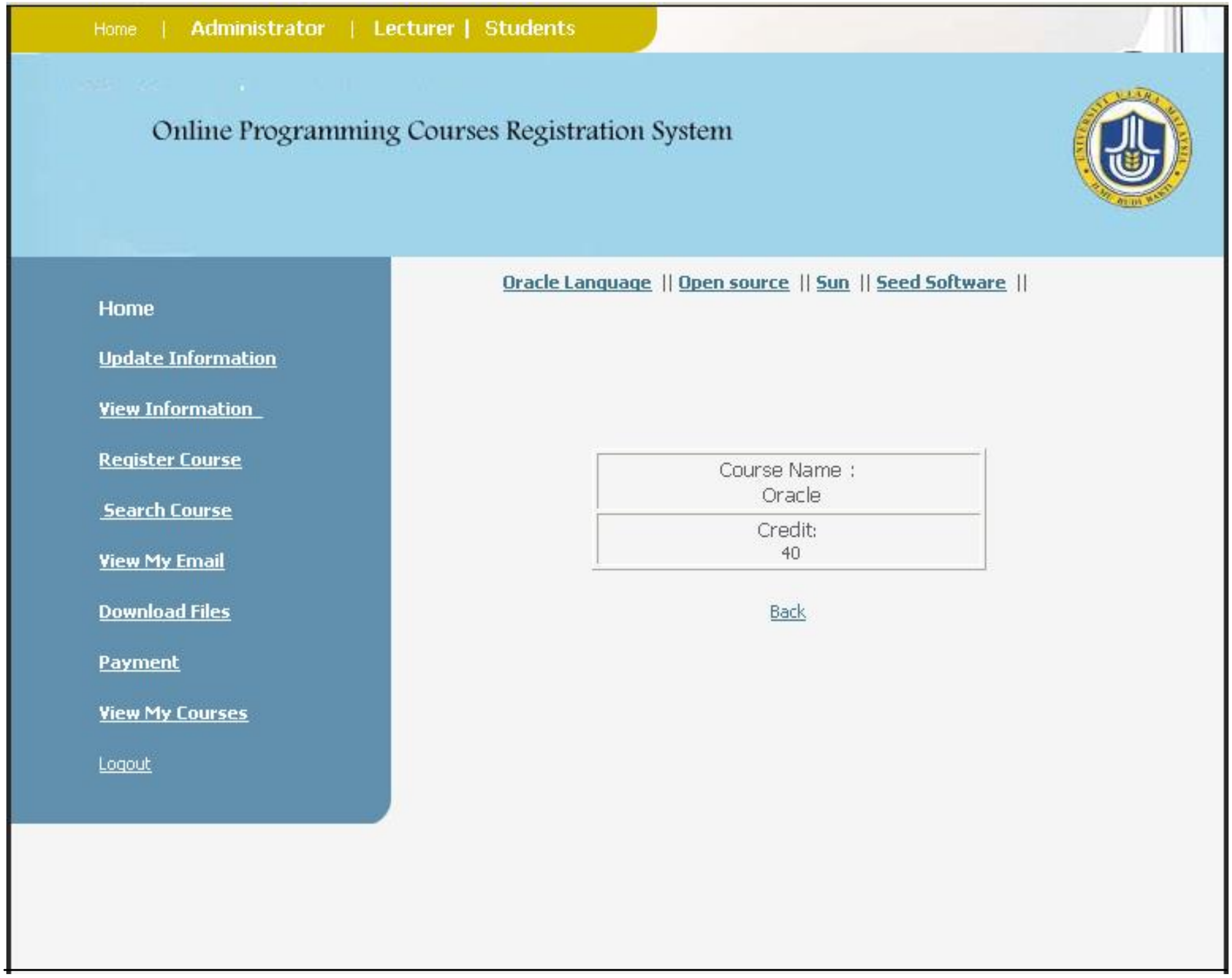


Figure 4.55: Search Programming Course Page

After press “Search Course” link in the Main Student Page, the prototype move to Search Courses page. Through this page the student can search about course that he wants register it to confirm if the course available or not. The student will press submit button to show course information on next page. And the system will check course code if course code is valid. The Student can Press Back link to go back to Main Student page.

4.5.4.4.1. View Programming Course Information Page



Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Update Information
View Information
Register Course
Search Course
View My Email
Download Files
Payment
View My Courses
Logout
Back

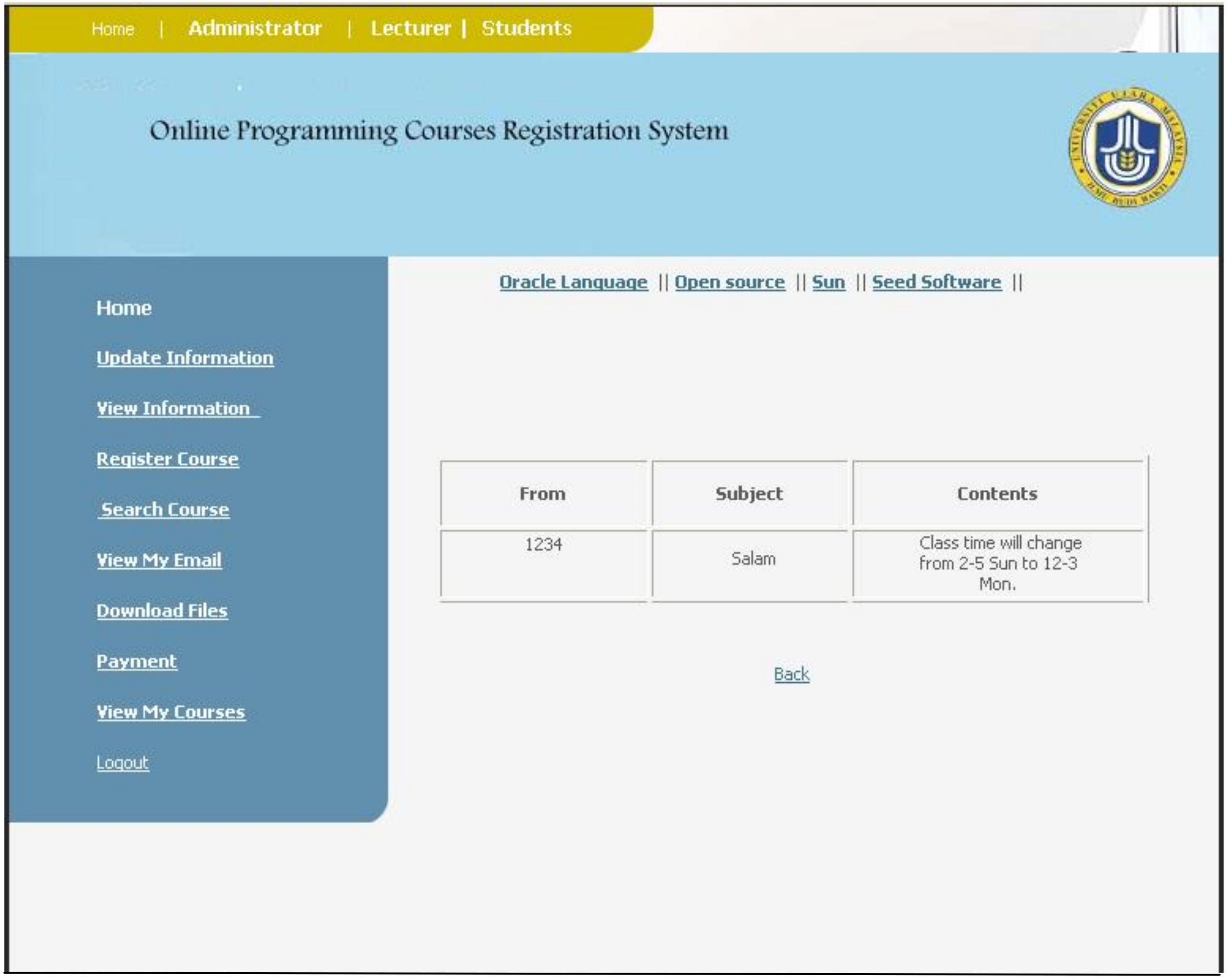
Course Name : Oracle

Credit: 40

Figure 4.56: View Programming Course Information Page

After press Submit button in the search Programming Course Page, the prototype move to View programming course information page to view course information which he did search about it. And the system will fetch course information such as course name and credit from course database. After that student can Press Back link to go back to Main Student page.

4.5.4.5.View My Email Information Page

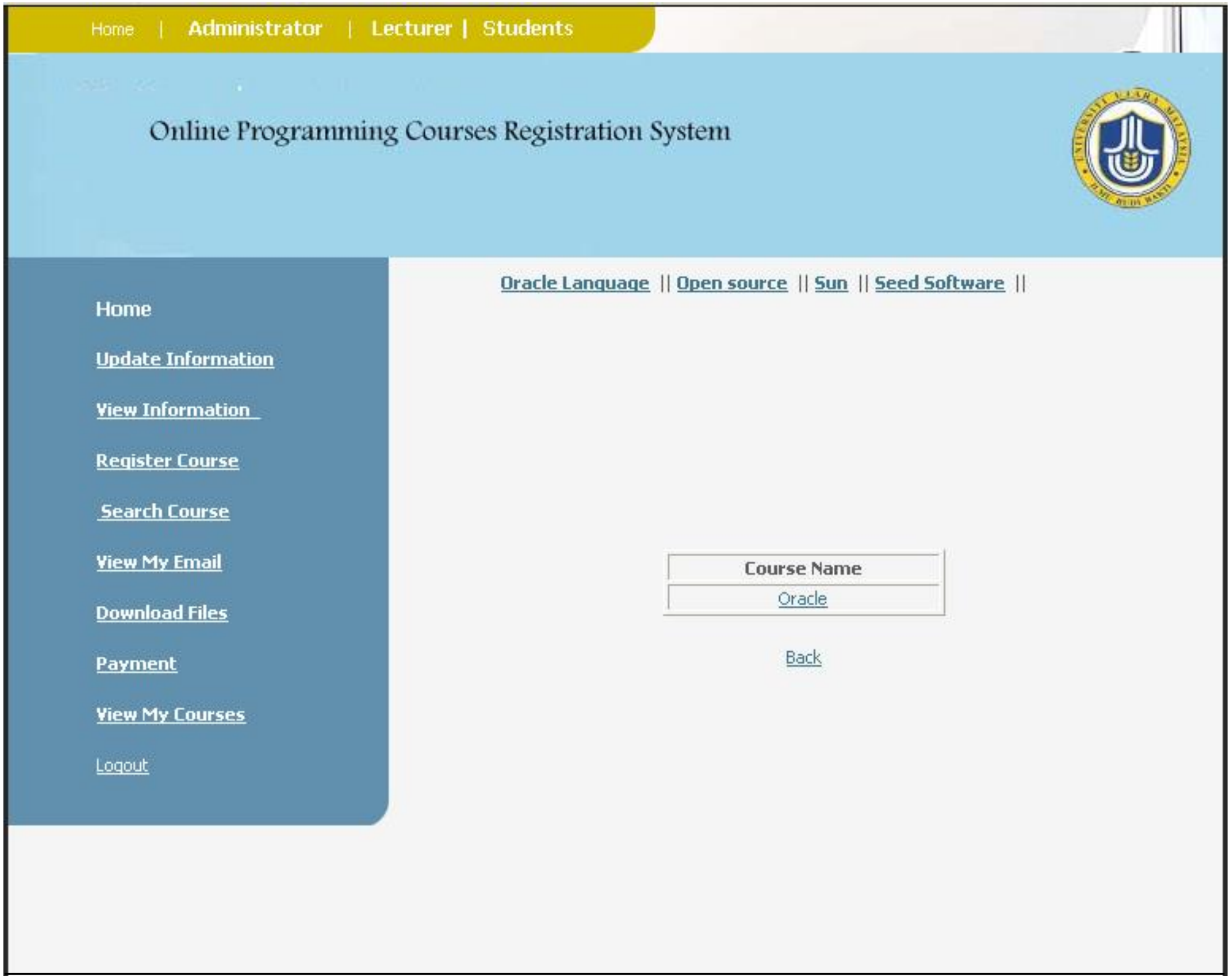

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Update Information
View Information
Register Course
Search Course
View My Email
Download Files
Payment
View My Courses
Logout
Back

From	Subject	Contents
1234	Salam	Class time will change from 2-5 Sun to 12-3 Mon.

Figure 4.57: View My Email Information Page

After press “View My Email” link in the Main Student Page, the prototype move to View Email information page. The student can view his or her email information. And the system will fetch email information such as lecturer Id, Subject and contents from Email database. After that student can Press Back link to go back to Main Student page.

4.5.4.6. Download Page

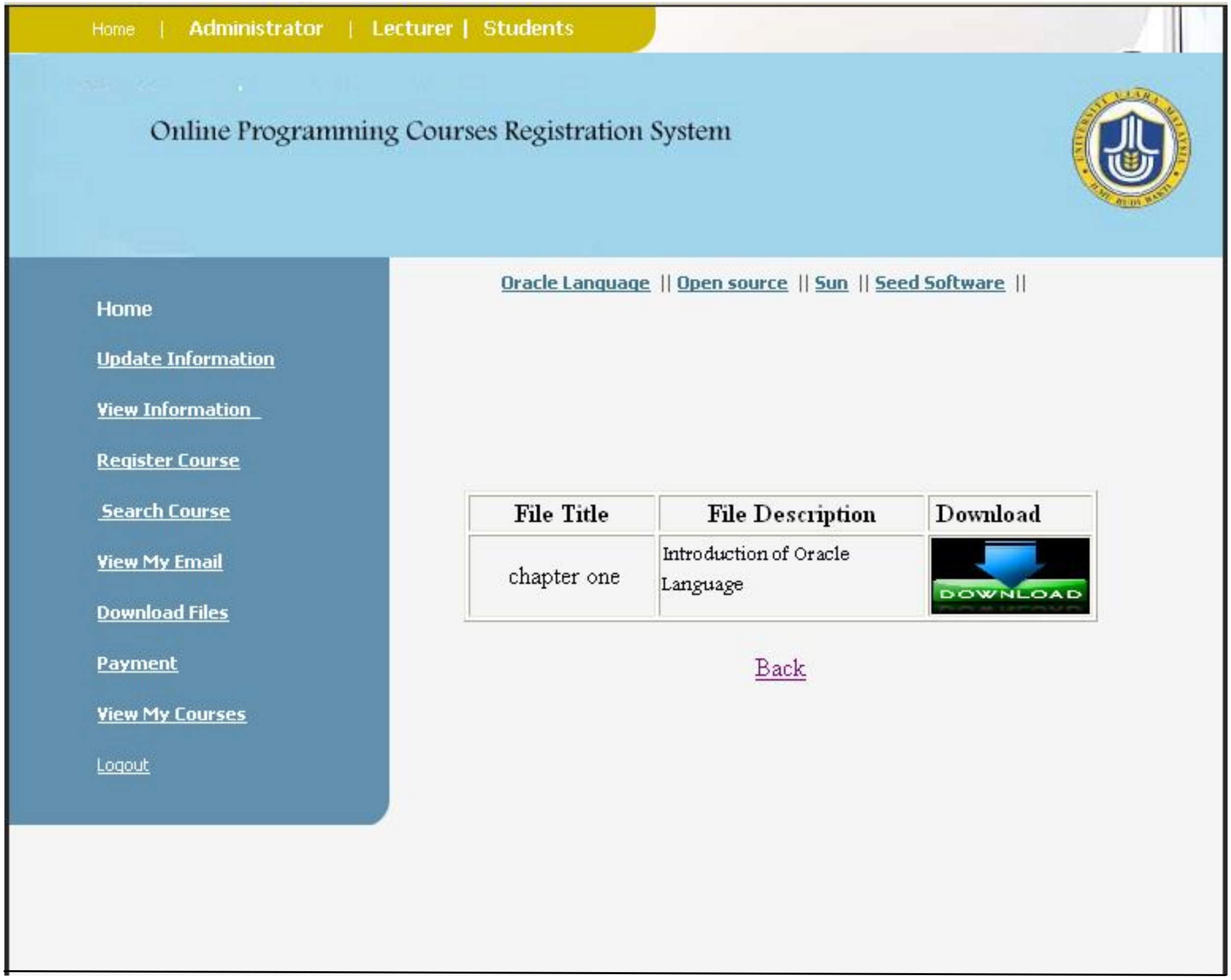

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Update Information
View Information
Register Course
Search Course
View My Email
Download Files
Payment
View My Courses
Logout
Back

Course Name
Oracle

Figure 4.58: Download Page

After press “Download Files” link in the Student Page, the prototype move to download page to view programming course name. This programming course which a student already registered it. The student can press course name which he want download it. And the Prototype will move to next page (View Download File information Page). After that student can Press Back link to go back to main student page.

4.5.4.6.1. View Download File Information Page

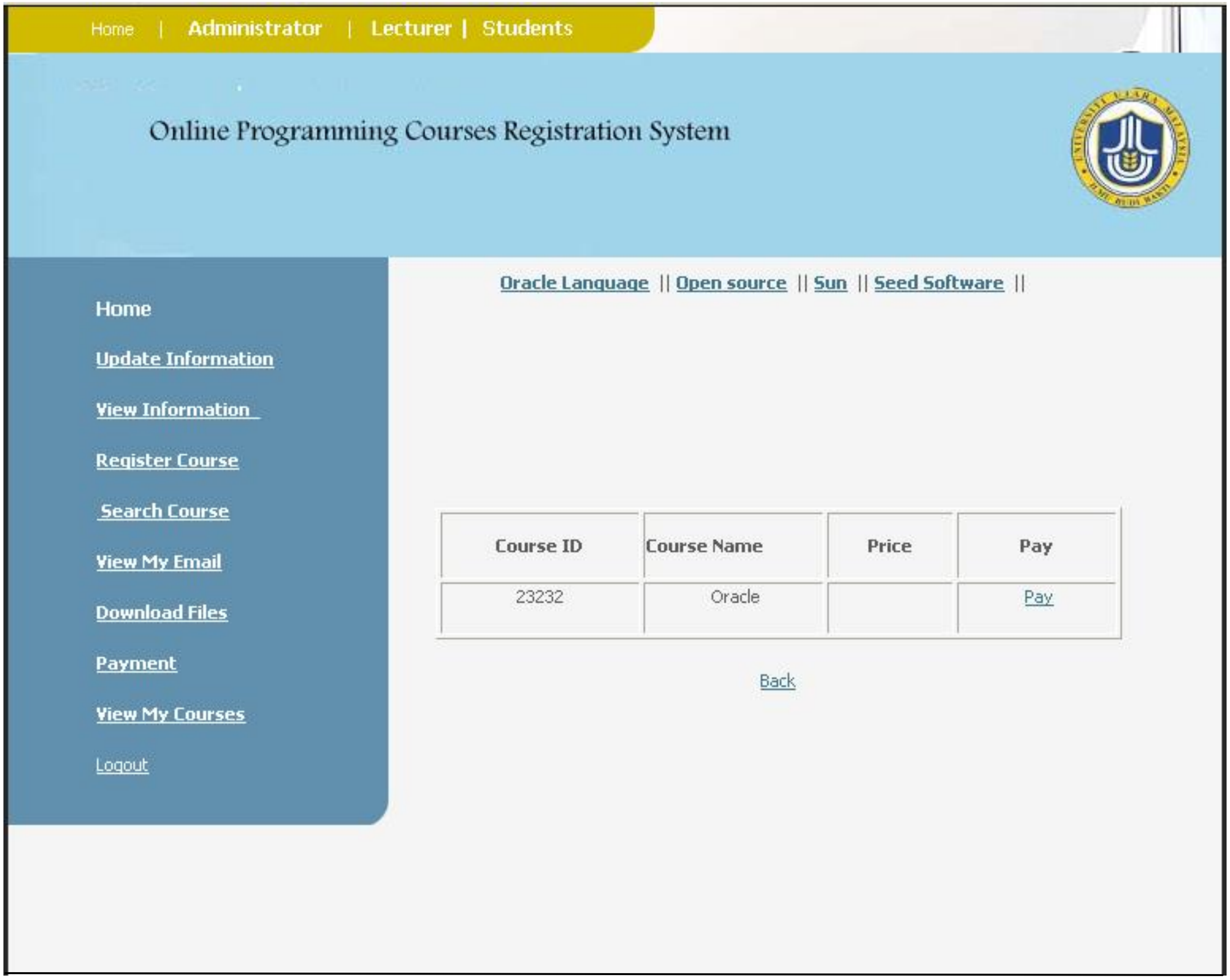

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Update Information
View Information
Register Course
Search Course
View My Email
Download Files
Payment
View My Courses
Logout
DOWNLOAD
Back

File Title	File Description	Download
chapter one	Introduction of Oracle Language	DOWNLOAD

Figure 4.59: View Download File Information Page

After press "Course name" Link in the Download Page, the prototype move to view download Files Information Page. This page includes information about the file such as file Title, file description and Download button. The system will fetch file information from Upload database. Through this page the student can download file by press Download button. Also the student can Press Back link to go back to student page.

4.5.4.7.Payment Page



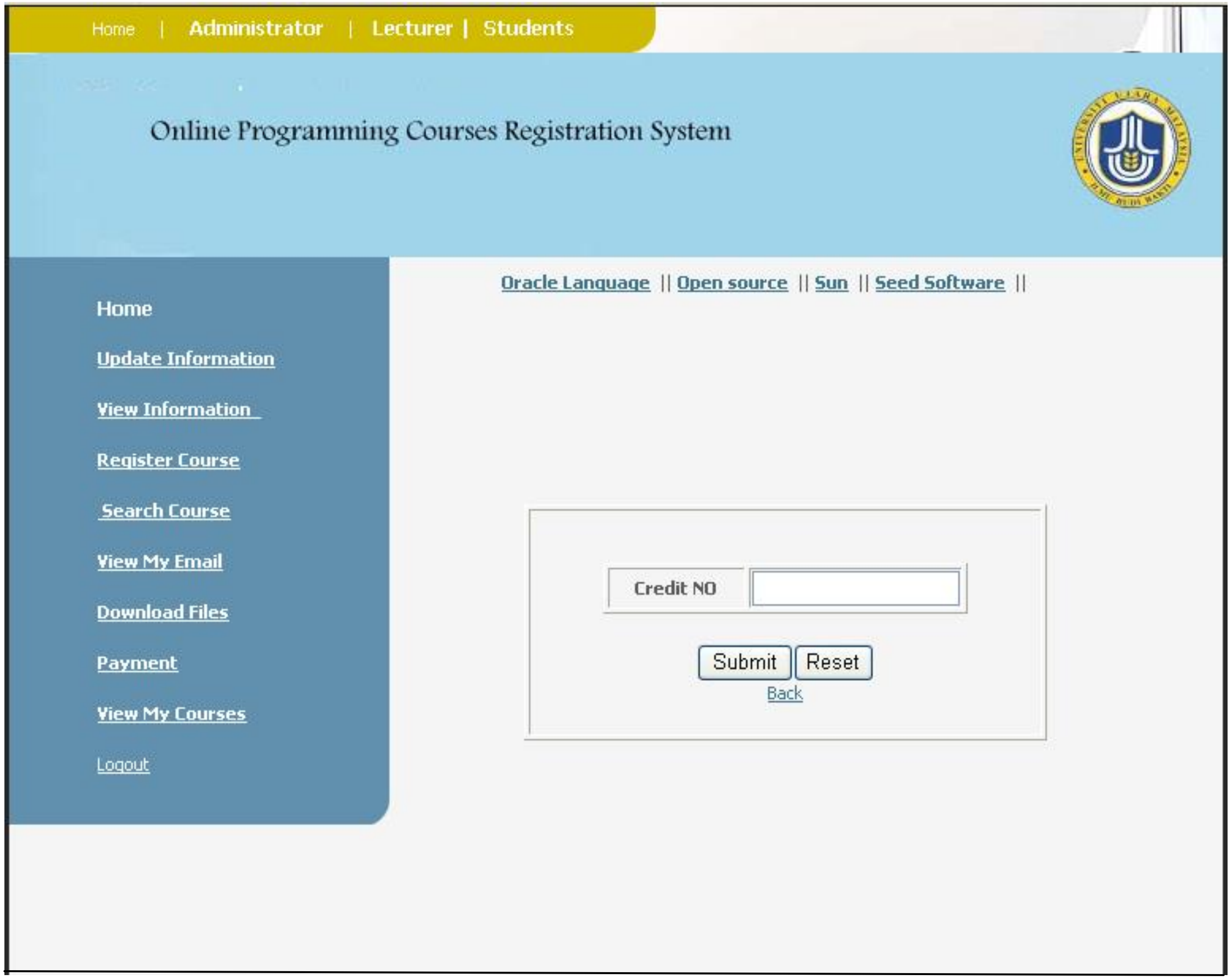
Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Update Information
View Information
Register Course
Search Course
View My Email
Download Files
Payment
View My Courses
Logout
Back

Course ID	Course Name	Price	Pay
23232	Oracle		Pay

Figure 4.60: Payment Page

After press “Payment” link in the student Page, the prototype moves to Payment page. This page includes course information such as course Id, course name and Price button. The student will press pay Link to pay fees for this programming course (Oracle course), then the prototype will move to next page (Account Page). The student can Press Back link to go back to student page.

4.5.4.7.1. Account Page

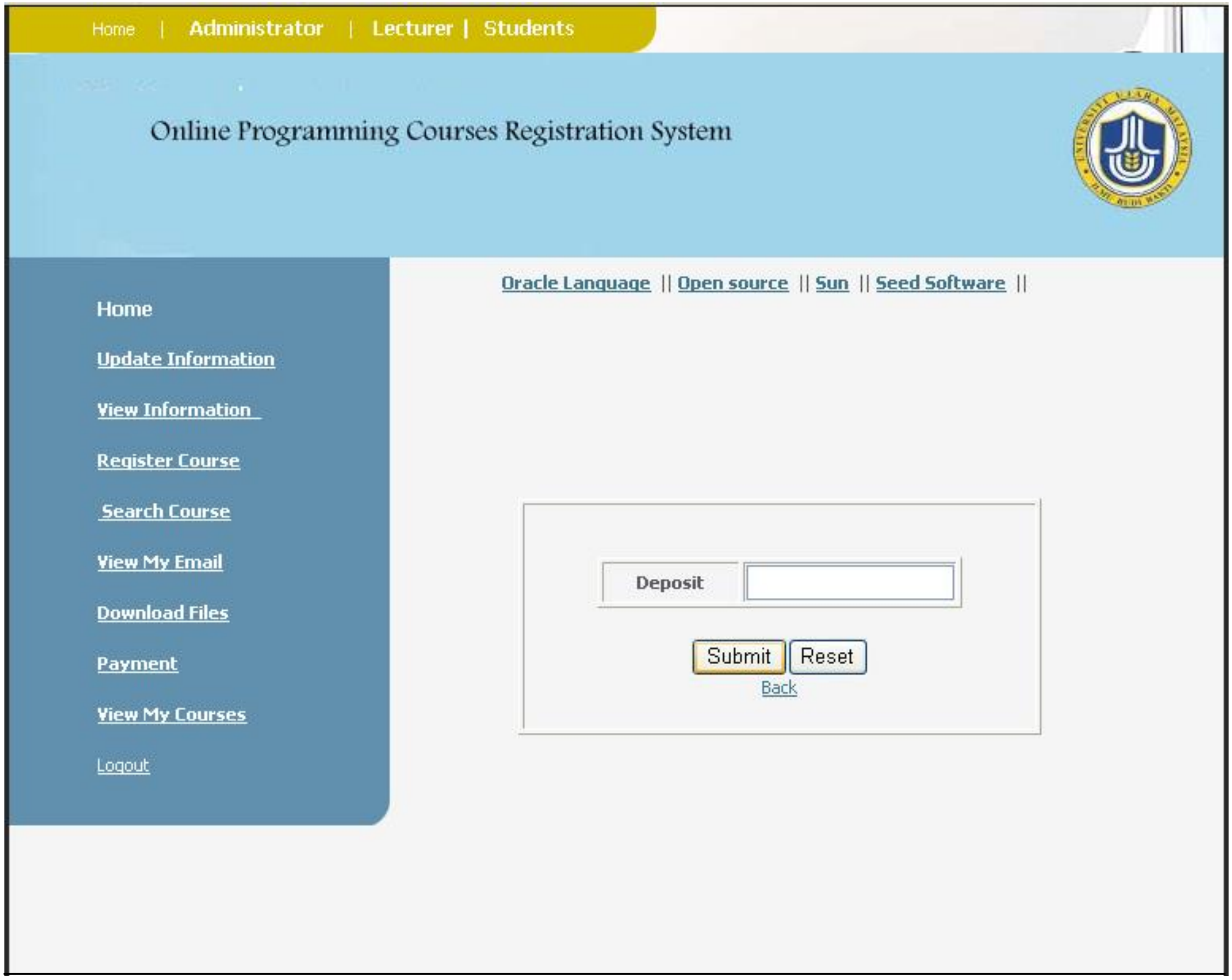


Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Update Information
View Information
Register Course
Search Course
View My Email
Download Files
Payment
View My Courses
Logout
Credit NO
Submit
Reset
Back

Figure 4.61: Account Page

After press "Pay" Link in the Payment Page, the prototype moves to Account page. Through this page the student will insert his/her credit no. And then he will press submit button. The system will verify his/her credit no if credit no valid or not. Or He can Press Back link to go back to student page or Reset button to empty all fields.

4.5.4.7.1.1.Deposit Page

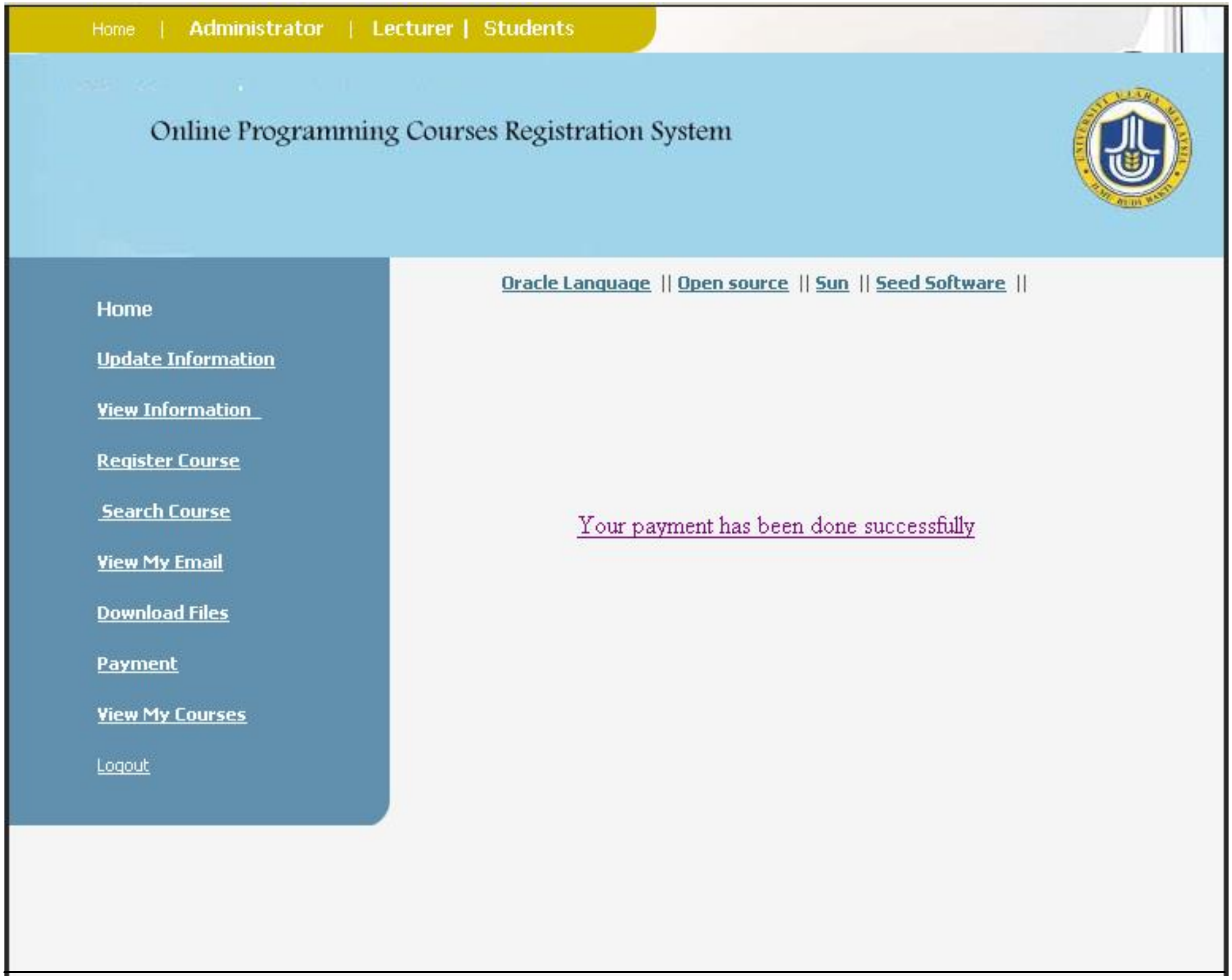


Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Update Information
View Information
Register Course
Search Course
View My Email
Download Files
Payment
View My Courses
Logout
Deposit
Submit
Reset
Back

Figure 4.62: Deposit Page

After press submit button in the Account Page, the prototype moves to Deposit page. Through this page the student will insert deposit value which he wants to pay for his/her course. And then he will press submit button. The system will verify his/her balance no if his/her balance >= deposit value, then next page will appear. Or He can Press Back link to go back to student page or Reset button to empty all fields.

4.5.4.7.1.1.1. Message Payment Page

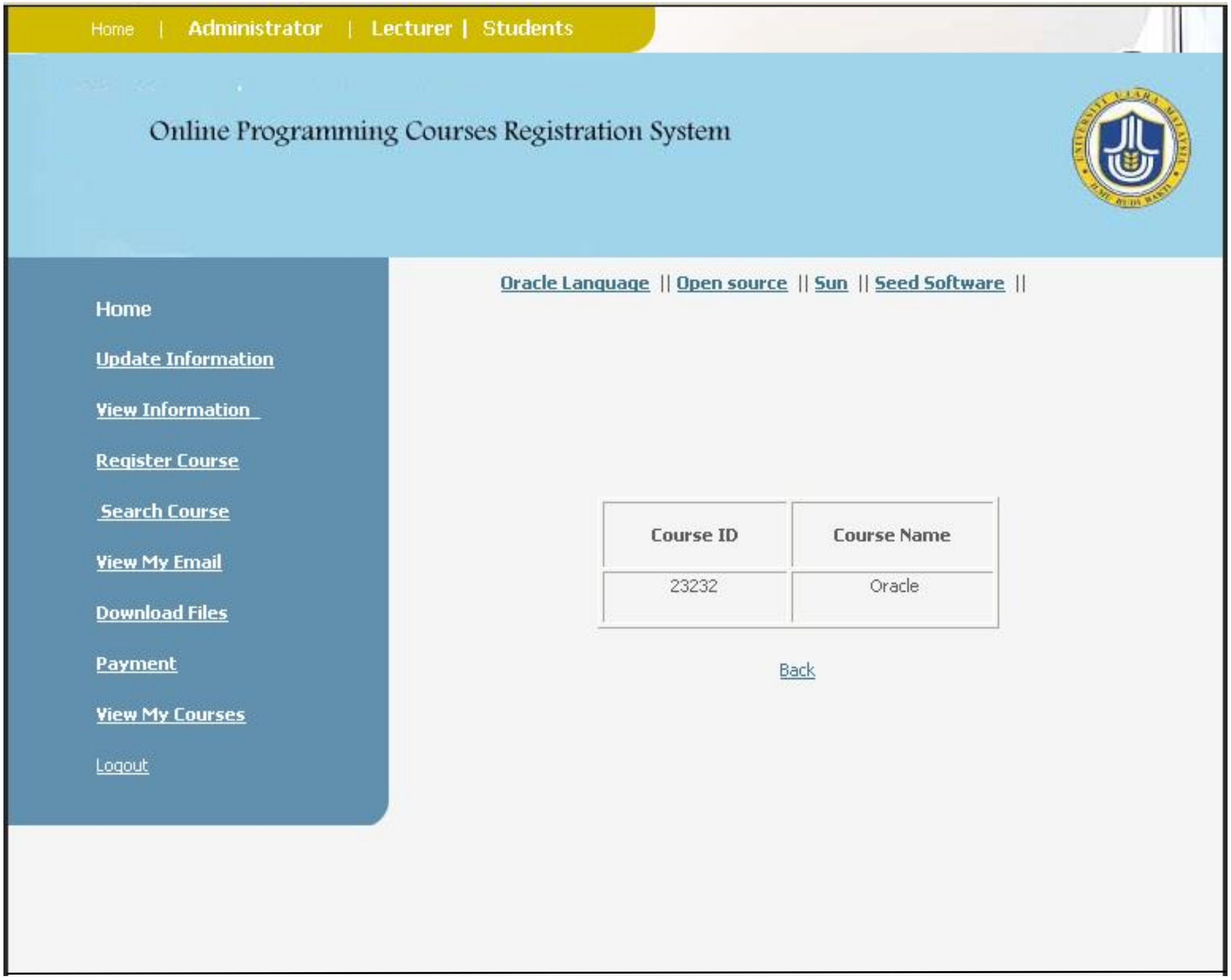

Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Update Information
View Information
Register Course
Search Course
View My Email
Download Files
Payment
View My Courses
Logout

Your payment has been done successfully

Figure 4.63: Message Payment Page

After press “submit” button in the view Deposit Page, the prototype move to massage payment page to tell student your payment has been successfully. After that student can press the register course successfully link to go back to student page.

4.5.4.8.View My Course Page


Home | Administrator | Lecturer | Students
Online Programming Courses Registration System
Oracle Language || Open source || Sun || Seed Software ||
Home
Update Information
View Information
Register Course
Search Course
View My Email
Download Files
Payment
View My Courses
Logout
Back

Course ID	Course Name
23232	Oracle

Figure 4.64: View My Course Page

After press “View My Course” link in the StudentPage, the prototype moves to View My Course page which will appear programming courses information that he already paid fees it. After that student can press back link to go back to student page.

4.6.Summary

This chapter introduces the Analyses Online Programming Course Registration and list of functional, non functional and software requirements. Also discuss Use Case Model, Actors and Use-Case. After that, this chapter discusses Programming Course Registration Prototype Design through snapshot of the prototype of the different actor who played in the prototype.

CHAPTER FIVE

DISCUSSION AND EVALUATION

5.0 Introduction

The main aim of this chapter is to discuss the evaluation of the On-line Programming Course Registration System. A usability test is one of the most fundamental methods in usability evaluation, because real test users are asked to use the product. The moderator of the test gives predetermined test tasks one at a time to the test user, who in turn performs the tasks with the user interface (Nielson, 1993). The users are usually asked to think while doing the test tasks. Interviews are also often used in order to gain more insight into the user's actions with the interface.

The Evaluation Questionnaire was designed according to the Likert Scale. According to Uebersax (2006) Likert scales were first developed by Rensis Likert, a sociologist at the University of Michigan from 1946 to 1970. Likert relates to the measurement of psychological attitudes and hopes to do so in a "scientific" way. The questioners have two part the first one talk about General information and the second one system aspects have some of categories are perceived usefulness, perceived ease of use, Attribute of Usability and User Satisfaction. Also the interview is used with the program manager of Computing Professional Enrichment & Development Division (CoPED) (In appendix A).

5.1. Evaluation Techniques

According to Nielson (2000) the evaluation uses usability testing based on the standard tests followed by interview in a closed environment with video equipment. Testing with potential users can obtain as efficient feedback as possible in a short time frame and with the available resources. It is also irrelevant to ask people in a focus group to predict whether they would like something they have not tried, so the only way to get valid data is to let users experience the technology before opinions are sought (Nielson, 1998).

The system evaluation measures the system usability that achieved the proposed objective which is:

To improve the interaction between the students of FTM College in UUM and the Professional Enrichment & Development Division (CoPED) Center.

5.2 The Questionnaire Question

The questions in the questionnaire are dividing into two parts:

a) Part one: General Information.

- Gender
- Age
- Educational background

b) Part Two: System Aspects.

- Perceived usefulness.
- Perceived ease of use.
- Attribute of usability.
- User satisfaction.

5.3 DATA ANALYSIS

The data collected through the questionnaire has been analyzed using SPSS software, version 12.0. Different statistics were used for data analysis. The following section describes the result obtained through analysis of data.

The following information was gathering from the first section of the questionnaire. All of the 30 participants were IT students and Staff of Professional Enrichment & Development Division (CoPED) Center. The analysis shows that:

5.3.1 General information

Table 5.1: Descriptive Statistics for General Information

	N	Mean	Std. Deviation
Gender	30	1.5667	.50401
Age	30	1.4333	.50401
Educational background	30	1.8000	.66436
Valid N	30		

According to the table above**,** that shown the main schema (N, Mean and Std deviation).The system evaluation measure using the Online programming courses registration system to allow for the students to register programming course and to download material at anytime and anywhere. According o the result above the result Mean of Gender was 1.5667. The result Mean of Age was 1.4333 and the result Mean of Educational background was 1.8000. The following figures have shown the percent for each once.

1) Figure 5.1: The percent of the age.

The following information was gathering from the first section of the questionnaire for participants. The analysis shows that 43.3% of participants are male (M) and 56.7% are female (F).

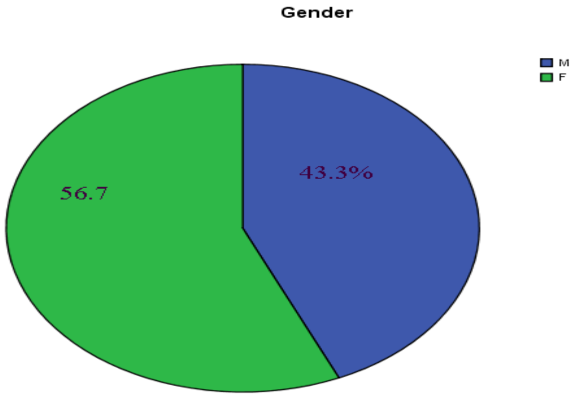


Figure 5.1: Gender of the Respondents

2) Figure 5.2: The percent of the age.

The age distribution of the study sample, were: 43.3% participants who their age 19-23 and 56.7% participants who their age 24-28.

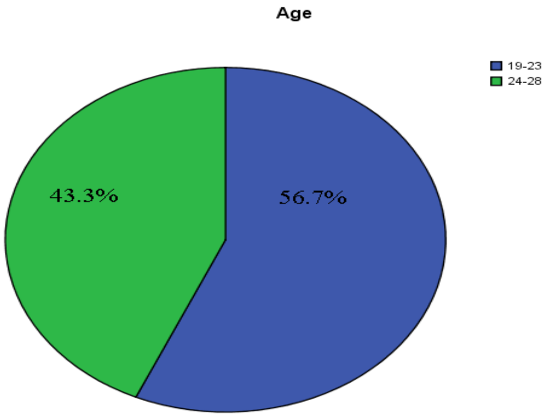


Figure 5.2: The percent of the age.

3) Figure 5.2: The percent of the educational background.

The education background distribution of the study sample, were: Diploma 33.3%, Degree 53.3% and master 13.3%.

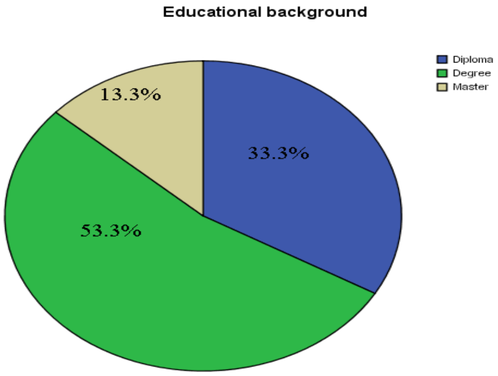


Figure 5.3: the percent of the Educational background

5.3.2 SYSTEM ASPECTS

This part has four sections to evaluate the prototype, here in this section the prototype usability combine the frequency factor. Therefore, in the following sections will discuss around the usability scales.

Reliability is the degree to which measure are free from error and therefore yield consistent results. According to Sekaran (2003), the closer the reliability coefficient gets to 1.0, the better it is, and those values over .80 are considered as good. Those value in the .70 is considered as acceptable and those reliability value less than .60 is considered to be poor (Sekaran, 2003).

5.3.2.1 Perceived Usefulness

This first section to evaluate the perceived usefulness the following table describes the descriptive for these questions.

Table 5.2: descriptive statistics for perceived usefulness

Perceived Usefulness		Question_4	Question_5	Question_6	Question_7	Question_8
N	Valid	30	30	30	30	30
	Missing	0	0	0	0	0
Mean		4.3667	4.0333	3.9000	4.1333	4.1000
Std. Deviation		.76489	1.03335	1.09387	.93710	.84486
Sum		131.00	121.00	117.00	124.00	123.00

According to the table above**,** that shown the main schema (N, Mean and Std deviation).The system evaluation measure the perceived usefulness of using the Online programming courses registration system to allow for the students to register programming course and to download material at anytime and anywhere. In reliability analysis (Appendix A), the Cronbach's Alpha value for independent variables (question 4, question 5, question 6, question 7, and question 8) is 0.704 . That value in the .704 is considered as acceptable (Sekaran, 2003). According o the result above that gave the proposed solution a high usefulness from the Student. However the result of:

1. Question 4 (OPCRS Prototype will enable the student to register programming courses and pay fees) was when the mean= 4.3667 from 5. And the following figure shown demonstrates that 53.3 % of the respondents Strongly Agree, 30.0% are Agree, 16.7 % are Neutral.

Because the scale of the Strongly Agree percentage value (53, 3 %) for the above question is greater than the other scales percentage value. it can be concluded, the participants believe that the system is able to allow them to register and pay fees electronically, rather than the manual way which was registration process and pay fees manually by banking and bring the slip for the center. This mean Registration process and payment will be easier for students.

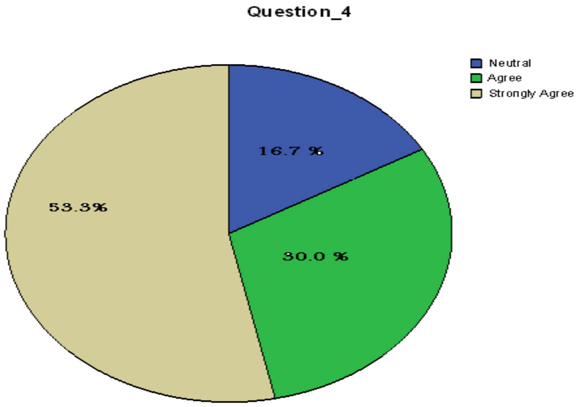


Figure 5.4: Question 4 statistical diagram

2. Question 5 (Using **OPCRS** will reduce students effort) was when the mean= 4.0333 from 5. And the following figure shown demonstrates that 43.3 % of the respondents Strongly Agree, 26.7 % are Agree, 20.0 % are Neutral and 10.0 % disagree.

Because the scale of the Strongly Agree percentage value (26.7 %) for the above question is greater than the other scales percentage value. it can be concluded, the participants believe that the system is reduce effort them. The student can to make registration from anywhere and anytime. Rather than, the manual way this was registration manually and required from the student to come to the center for registration. Therefore, the system will reduce an effort for student.

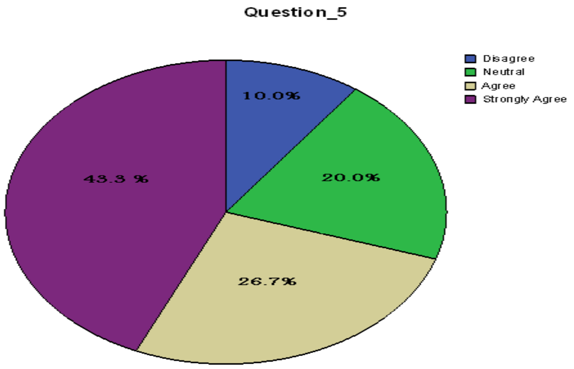


Figure 5.5: Question 5 statistical diagram

3. Question 6 (Using **OPCRS** will reduce students time**)** was when the mean= 3.9000 from 5. And the following figure shown demonstrates that 40.0 % of the respondents Strongly Agree, 23.3 % are Agree, 23.3% are Neutral and 13.3 % disagree.

Because the scale of the Strongly Agree percentage value (23.3 %) for the above question is greater than the other scales percentage value, it can be concluded The participants believe that the system is reduce time them. The student can to make registration from anywhere and anytime. Rather than, the manual way this was registration manually and required from the student to come to the center for registration. That was spent time him and effort him.

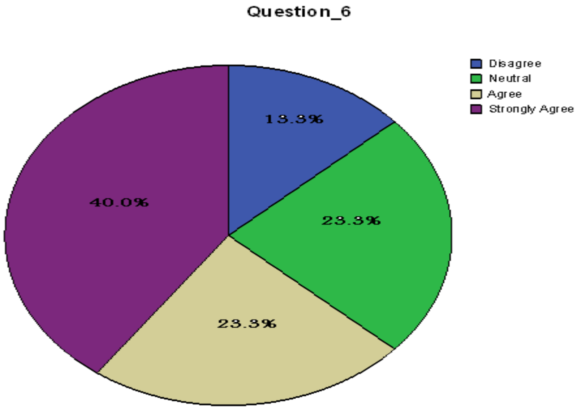


Figure 5.6: Question 6 statistical diagram

4. Question 7 (Using **OPCRS** would easy to understand each of the links on the home page) was when the mean= 4.1333 from 5. And the following figure shown demonstrates that 46.7 % of the respondents Strongly Agree, 23.3 % are Agree, 26.7% are Neutral and 3.3 % disagree.

Because the scale of the Strongly Agree percentage value (23.3 %) for the above question is greater than the other scales percentage value, it can be concluded the participants believe that the system is easy to understand for each contents and links in the home page.

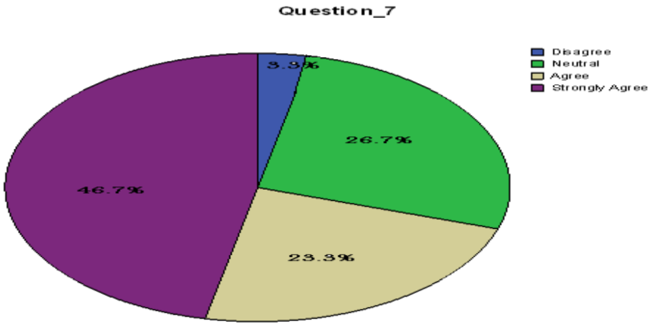


Figure 5.7: Question 7 statistical diagram

5. Question 8 (I would find **OPCRS** secure) was when the mean= 4.1000 from 5. And the following figure shown demonstrates that 36.7 % of the respondents Strongly Agree, 40.0 % are Agree, 20.0% are Neutral and 3.3% disagree.

Because the scale of the Agree percentage value was (40.0 %) for the above question is greater than the other scales percentage value, it can be concluded the participants believe that the system is more secure. The student must have correct username and password to enter to system. The student also must to grant approval to register programming language.

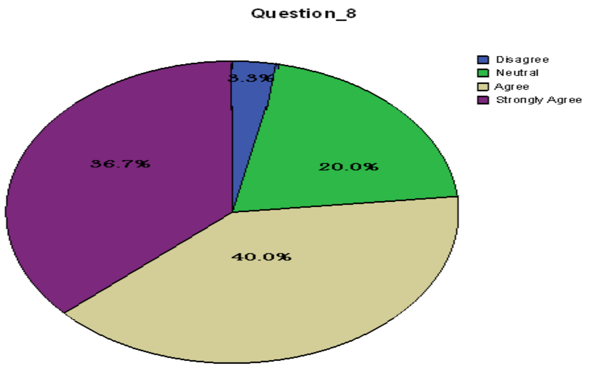


Figure 5.8: Question 8 statistical diagram

5.3.2.2 Perceived Ease Of Use

This second section to evaluate the perceived ease of use, the following table describes the descriptive for these questions.

Table 5.3: descriptive statistics for perceived ease of use

		Question_9	Question_10	Question_11	Question_12	Question_13
N	Valid	30	30	30	30	30
	Missing	0	0	0	0	0
Mean		4.2333	4.3667	4.0000	3.7000	3.9667
Std. Deviation		1.04000	.92786	1.11417	1.14921	.88992
Sum		127.00	131.00	120.00	111.00	119.00

According to the table above**,** that shown the main schema (N, Mean and Std deviation).The system evaluation measure the perceived ease of use of using the Online programming courses registration system to allow for the students to register programming course and to download material at anytime and anywhere. According o the result above that gave the proposed solution a high perceived Ease of use from the Student. In reliability analysis (Appendix A), the Cronbach's Alpha value for independent variables (question 9, question 10, question 11, question 12, and question 13) is 0.704. That value in the .704 is considered as acceptable (Sekaran, 2003). However the result of:

1. Question 9 (Learning to operate **OPCRS** would be easy for me) was when the mean= 4.2333 from 5. And the following figure shown demonstrates that 56.7 % of the respondents Strongly Agree, 20.0% are Agree, 13.3 % are Neutral and 10.0 % disagree.

Because the scale of the Strongly Agree percentage value was (56.7 %) for the above question is greater than the others scale percentage value, it can be concluded. The participants believe that the operate system is more easy to learning for the students. Especially the prototype include clear words and easy to use. In addition, it has more characteristics as easy to learn and design.

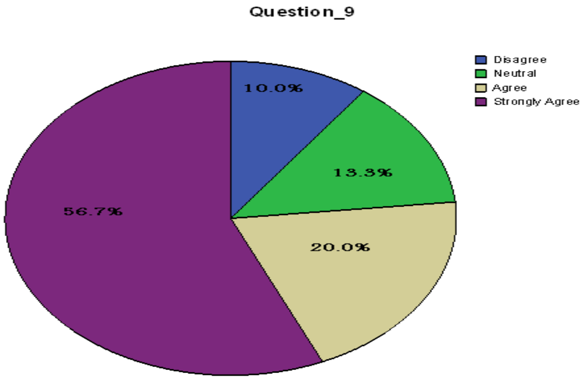


Figure 5.9: Question 9 statistical diagram

2. Question 10 (Using I would find it easy to get **OPCRS** to do what I want it to do) was when the mean= 4. 3667 from 5. And the following figure shown demonstrates that 66.7 % of the respondents Strongly Agree, 20.0% are Agree, 10.0% are Neutral and 3.3% disagree.

Because the scale of the Strongly Agree percentage value was (66.7%) for the above question is greater than the other scales percentage value, it can be concluded. The participants believe that the system is easy to learning for the students. Especially the meets all the needs of students as registration, bay fees, and files download.

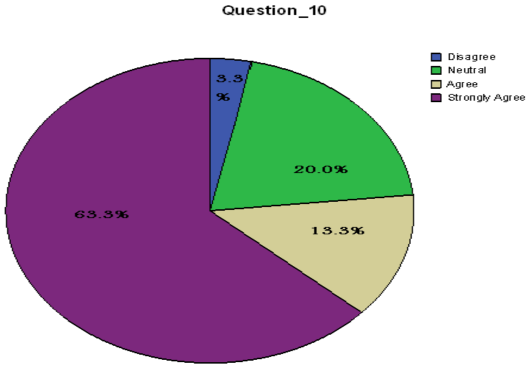


Figure 5.10: Question 10 statistical diagram

3. Question 11 (interaction with **OPCRS** would be clear and Understandable) was when the mean= 4.000 from 5. And the following figure shown demonstrates that 50.0 % of the respondents Strongly Agree, 10.0% are Agree, 20.0% are Neutral and 10.0% disagree.

Because the scale of the Strongly Agree percentage value was (50.0 %) for the above question is greater than the other scales percentage value, it can be concluded. The participants believe that the system is cleared and more understandable for the end-user. Especially the user interface was developed by where the student can understand prototype when he want use it. Otherwise the registration manually didn’t have any characteristic to interaction with end-user.

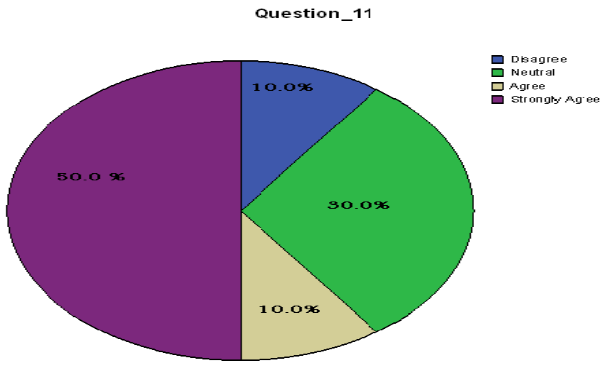


Figure 5.11: Question 11 statistical diagram

4. Question 12(I would find **OPCRS** to be flexible to interact with) was when the mean= 3.7000 from 5. And the following figure shown demonstrates that 23.4 % of the respondents Strongly Agree, 23.3 % are Agree, 23.3 % are Neutral and 20 % disagree.

Because the scale of the Strongly Agree percentage value was (46.7%) for the above question is greater than the other scale percentage value. it can be concluded, the participants believe that the system is flexible and friendly to interact with the end-user.

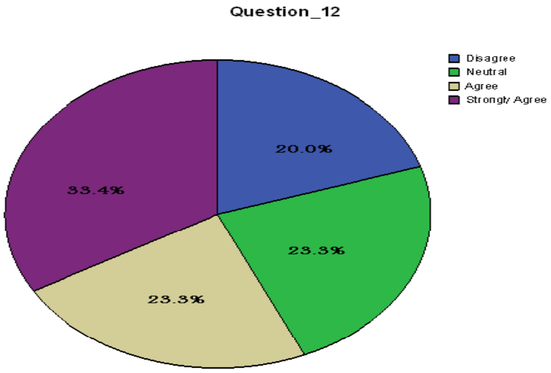


Figure 5.12: Question 12 statistical diagram

5. Question 13 (I would find **OPCRS** easy to use) was when the mean= 3. 9667 from 5. And the following figure shown demonstrates that 30.0 % of the respondents strongly Agree, 43.3 % are Agree, 20.0% are Neutral, and 6.7 are disagree.

Because the scale of the Strongly Agree percentage value was (30.0 %) for the above question is greater than the other scales percentage value, it can be concluded. The participants believe that the system was found ease to use from the end-user and more. Especially the prototype was written the link by English language, and everything in this prototype easier than manually registration, because the manually registration is follow the manual way to registration and give the instructions.

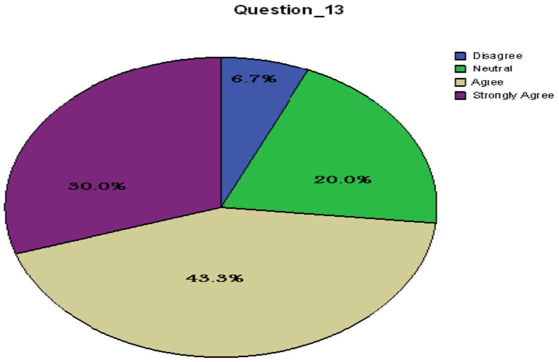


Figure 5.13: Question 13 statistical diagram

5.3.2.3 Attribute Of Usability

This third section to evaluate the Attribute of Usability, the following table describes the descriptive for these questions.

Table 5.4: Descriptive Statistics for Attribute of Usability

		Question_14	Question_15
N	Valid	30	30
	Missing	0	0
Mean		3.9000	3.9333
Std. Deviation		1.02889	1.04826
Sum		117.00	118.00

According to the table above**,** that shown the main schema (N, Mean and Std deviation).The system evaluation measure the Attribute of Usability of using the Online programming courses registration system to allow for the students to register programming course and to download material at anytime and anywhere. According o the result above that gave the proposed solution a high perceived usability from the Student. In reliability analysis (Appendix A), the Cronbach's Alpha value for independent variables (question 14, and question 15) is 0. 864. That value in the 0. 864 are considered as good (Sekaran, 2003). However the result of:

1. Question 14 (I found it easy to interact between lecturer and his/her students by using **OPCRS**) was when the mean= 3.9000 from 5. And the following figure shown the demonstrates that 30.0 % of the respondents Strongly Agree, 43.3 % are Agree, 16.7 % are Neutral, 6.7% are disagree, and strongly disagree are 3.3.

Because the scale of the Strongly Agree percentage value was (30.0%) for the above question is greater than the other scales percentage value, it can be concluded. The participants believe that the system was found ease to interact between the lectures and students. We know the manual registration system didn't have way to inform the students for everything new. Therefore, in this prototype we developed way to inform the students by send E-mail for everyone have these classes. In addition, the prototype provided many services for each student and lecturer to improve interact between them.

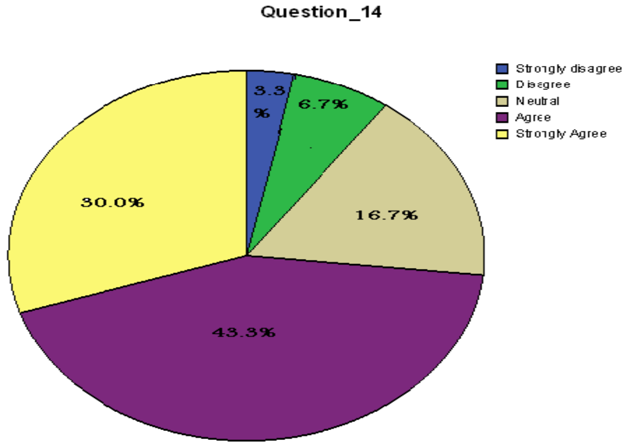


Figure 5.14: Question 14 statistical diagram

2. Question 15 (The procedure through **OPCRS** prototype by online is clear) was when the mean= 3. 9333 from 5. And the following figure shown demonstrates that 33.3 % of the respondents Strongly Agree 40.0 % are Agree , 16.7 % are Neutral, 6.7% are disagree, and 3.3 are strongly disagree.

Because the scale of the Agree percentage value was (40.0 %) for the above question is greater than the other scales percentage value, it can be concluded. The participants believe that the procedures in the OPCRS system were found clear and easier to usability For the student as registration, after registration can enter to prototype by login. He also registers any available programming course and pay fees it.

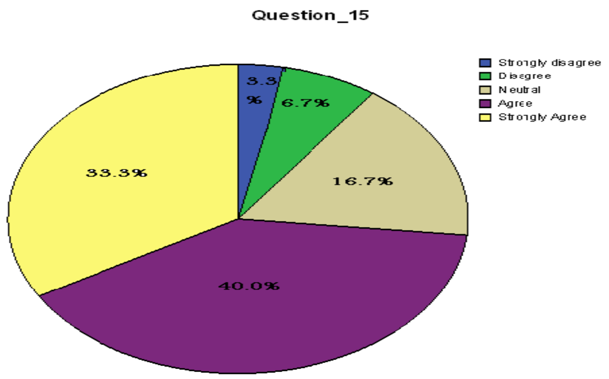


Figure 5.15: Question 15 statistical diagram

5.3.2.4 User Satisfaction

This last section to evaluate the User Satisfaction, the following table describes the descriptive for these questions.

Table 5.5: Descriptive Statistics for User Satisfaction

		Question_16	Question_17
N	Valid	30	30
	Missing	0	0
Mean		4.3667	3.8000
Std. Deviation		.96431	.55086
Sum		131.00	114.00

According to the table above that shown the main schema (Minimum, Maximum and the Mean).The system evaluation measure the User Satisfaction of using the Online programming courses registration system to allow for the students to register programming course and to download material at anytime and anywhere. According o the result above that gave the proposed solution a high User Satisfaction from the Student. In reliability analysis (Appendix A), the Cronbach’s Alpha value for independent variables (question 15, and question 16) is 0. 770. That value in the 0. 770 are considered as acceptance (Sekaran, 2003). However the result of:

1. Question 16 (I completely satisfied in using **OPCRS** prototype) was when the mean= 4.3667 from 5. And the following figure shown demonstrates that 63.3 % of the respondents strongly Agree, 16.7 .7% are Agree, 13.3 % are Neutral and 6.7 % disagree.

   Because the scale of the Strongly Agree percentage value was (63.3 %) for the above question is greater than the other scales percentage value, it can be concluded. The participants complete satisfied when using OPCRS prototype.

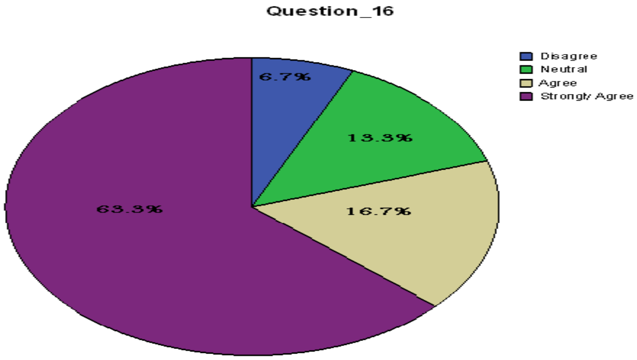


Figure 5.16: Question 16 statistical diagram

2. Question 17 (I feel very confident in using **OPCRS** Prototype) was when the mean= 3.8000 from 5. And the following figure shown demonstrates that 3.3 % of the respondents strongly Agree, 76.7% are Agree, 16.7 % are Neutral, and 3.3 % are disagree.

Because the scale of the Agree percentage value was (3.3%) for the above question is greater than the other scales percentage value, it can be concluded. The fell of the participants very confident when they used OPCRS and all the participants wishes to implement this prototype .due they have boring fell when using the manual system.

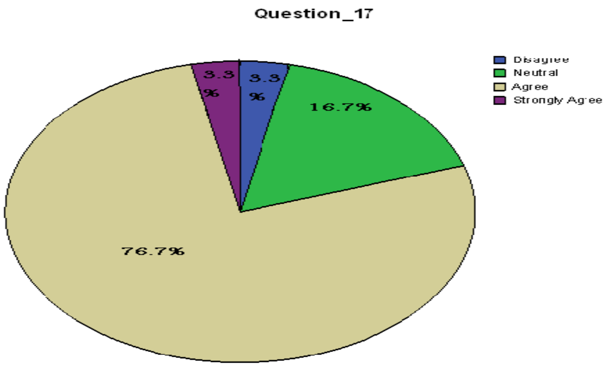


Figure 5.17: Question 17 statistical diagram

6.4 CONCLUSION

Evaluation takes part in an important part in the development process and can uncover usability deficits early during the design. In further works, more usability tests for the re-design application with real student should be conducted. Interviews with these test persons and evaluation to reach more people will help to shape application and better meet the user's opinion, requirements and expectations. The overall results were encouraging but improvement is definitely needed. The Cronbach's Alpha values range from .704 to .800 and are all above 0.7 which is considered as acceptance.

CHAPTER SIX

CONCLUSION

6.0 INTRODUCTION

The main aims of this chapter discuss the conclusion and recommendation of this study. The Conclusion will explain how this study achieved the goals, according to the objectives and problems and limitations encountered during the development of this project. Finally, brief recommendations will be given as contributions to future enhancements will also be discussed.

6.1 The Research Objectives

1. Identify the basic requirements for building an online registration system:

The prototype requirements for Online Programming courses Registration system prototype have been specified and illustrated in chapter Four. By specifying the prototype requirements, they have been used to solve the problem. In order to implement the suggested prototype. This study also includes the prototype analysis and design which is later used to develop the prototype of the desired prototype.

2. To propose model of online registration system by using Java Server Page (JSP):

After determined the requirements of the prototype, we moved to development phase .In this phase we designed and developed the prototype by Java server Page (JSP) language with My-SQL data base. Due the JSP have some of characteristics like security, compatible with any browser, and integrated.

3. To evaluate the user acceptance of the prototype:

After the prototype has been developed, it has been evaluated by 30 students and staff in the College of Arts and Sciences (CAS) in University Utara Malaysia (UUM) using on the usability test and the functionality evaluation.

6.2 Problems And Limitations

1) Learn JSP language:

The time frame did not enough to learn G language and understand it well and due to limited materials and tutorials in internet for JSP language, I face many difficulties to fix syntax errors which faced me during programming.

2) Design required class for the system:

I faced problem in how to identify the required class, in addition, how to interact with class after I identified it, where there are many of relationship between that classes who represent this system.

6.3 Recommendations And Future Work

After accomplishing the main phase of the project, there is a suggest additional features that if added will improve the quality and flexibility of this project. It is worth mentioning that the main objective in this study is to introduce new ideas for online Registration and e-payment systems to be used in educational applications and other fields.

Due to the time frame that is not sufficiently enough to assure the entire functionalities of the system. Future works can be carried out to overview the researches that came upon during the work of this project. It would be more suggestive to advice the one who needs to pursue some future works to follow every single step included in the project. Online Programming Courses Registration System has the following features that will benefit the students in the College of Arts and Sciences (CAS) in University Utara Malaysia (UUM) and Staff in COPED center:

- Check available services by online Registration system, to make it easy and more flexibility to deal with by the students and staff.
- As should do training for students and staff how to use online registration application.
- Provides direct, simple access to the focused valuable content via few keystrokes or text entry only.
- Reduces the amount of vertical scrolling by simplifying the text to display.

6.4 CONCLUSION

As was explained though chapter one, the objectives of this study are to develop the Web-application and do usability testing. As well as producing requirement model for Online Programming Registration System in the College of Arts and Sciences (CAS) in University Utara Malaysia (UUM*)* by Using JSP language. In order to make requirements more understandable the requirements have been modeled by using some of UML diagrams such as use case diagram ,use case specification and sequence diagram to design the system requirement in order to illustrate the research objectives. However, the other features have been added to provide students and staffs with the appropriate enquire. Finally the application has been implemented using WML technology according with Online Programming Registration System in the College of Arts and Sciences (CAS) in University Utara Malaysia (UUM*)* by Using JSP language. Implementing online Registration and e-payment to support service delivery to the fact that reduce human errors and increase efficiency.

REFERENCES

Ahmad, H., and Yusoff, R. (2001). *Integrated process design for e-learning*: a case study. 488 – 491.

Ahn, T., Ryu, S. & Han, I. (2004). *The Impact of the Online and Offline Features on the User Acceptance of Internet Shopping Mal*l. Electronic Commerce Research and, 3 (2004), 405-420.

AL-Bastaki , Y.(2005). *A framework for a WAP-based course registration system*.44(3) , Pp 327 - 342 .

Boroson, E. (2003). "Course Registration System, the on-line registrar, the on-line registrar."Retrieved March 04,2008, from http://www.cs.brown.edu/courses/cs190/old.dhl2003/ asgns/2-7/tmp/eboroson_specs.pdf.

Ciebiera, k., Mincer-Daszkiewicz.J , Walen, T. (2004) . *New Course Registration Module for the University Study-Oriented System*, EUNIS’2004, the 10th International Conference of European University Information Systems, Bled, Slovenia, pp. 247–252.

Carat, G. (2002). *E-Payment Systems database – Trends and Analysis*, Electronic Payment Systems Observatory (ePSO), Institute for Prospective Technological / European Commission.

Chan, P. (2008). *Dynamic Web Service Composition*: A New Approach in Building Reliable Web Service. Pp 20 – 25.

Cui, H. (2005). Digitized *OA system construction of University based on web service technology*.113: 698 – 702.

David, F.D. (2004). *Perceived Usefulness, Perceived Ease of Use and End-User Acceptance of Information technology*. London: Prentice Hall

George, C., and Scerri, J. (2007). *User-Generated Content Online:* Legitimate power or the Wild West? BELETa: 2007 annual conference Hertfordshire.

Gunawardana .J.M.N.C, Ishara . G.P, Ragel .R.G & Radhakrishnan .S.(2008) . *Course Registration System for the Faculty of Engineering in University of Peradeniya* . Proceedings of the Peradeniya University Research Sessions, Sri Lanka, Vol.13, Part II.

Holden.E & Weeden.E (2003) . *The Experience Factor in Early Programming Education* . SIGITE'04, October 28-30, 2004, Salt Lake City, Utah, USA.

Hyland.E & Clynch.G (2002) . *Initial experiences gained and initiatives employed in the teaching of Java programming in the Institute of Technology* . Principles and Practice of Programming in Java.pp.101

James Turner. (2002). Mysql *And Jsp Web Applications*: Data-driven Programming Using Tomcat And **Mysql**. Retrieved from : http://www.biblio.com/isbn/9780672323096.html

Jing,Y .(2009) . *On-line Payment and Security of E-commerce* . Proceedings of the 2009 International Symposium on Web Information Systems and Applications (WISA'09). *Nanchang, P. R. China, May 22-24, 2009, pp. 046-050.* March 04, 2008, from http://www.cs.brown.edu/courses/cs190/old.dhl2003/asgns/2-7/tmp/eboroson_specs.pdf.

John ,J. Micheal, A. Irina, A. and Ramazan, B. (2004), *Online Interactive Home Work Grading System* . Retrieved from : http://www.iiisci.org/journal/CV$/sci/pdfs/P792477.pdf

Junping, W. (2007). *Risk Control and Estimation of Online Services Market Segmentation*. 1 – 5.

Kerner, S. (2006). *Is PHP the Cure for the 'Broken' Web*, internetnews.com.

Knight A., and Dai, N. (2002). *Objects and the Web*. IEEE Software. pp.51-59. IEEE.

Lankton, N. (2007). *Antecedents and Dimensions of Online Service Expectations*." 54(4): 776 – 788.

Laudon, K. C, & Laudon, J. P . (2000) . *Management Information Systems*: Prentice Hall PTR Upper Saddle River, NJ, USA.

Lee, K. and Jae, Y. (2008). *A Survey on Internet Payment Systems*. Software Ecosystem.

McCarth , B . *An Evaluation of an Online Programming Course*. Communications of the IBIMA . Volume 5, 2008.

Naini, V., and Tanik, M. (2008). *A Web-based interactive Student Advising system using Java frameworks*. 172 – 177.

Nijaz, (2000), *Dynamic web-based application development*. New York: Prentice Hall.

Nielsen, J. (1993), *Usability engineering*, Boston: AP professional. (chapter 6.5: test tasks, 185-186)

Nielsen, Jakob (2000). *"Why You Only Need to Test With 5 Users*".19 March. Retrived from : http://www.useit.com/alertbox/20000319.html

Nielsen, J. (1998). International Standard, Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs). Switzerland: Int. Organization for Standardization Geneva.

Raghu Ramakrishnan and Johannes E. Gehrke 3003) . *Database Management Systems*, Third Edition. McGraw Hill.

Rayport and Jaworski,. (2004) *Best Face Forward, Why Companies Must Improve Their Service Interfaces with Customers*. Harvard Business School Press, January 2005

Roy, T., and Richard, N. (2002). *Principled Design of the Modern Web Architecture*. ACM Transactions on Internet Technology, Vol. 2, No. 2, May 2002, Pages 115–150.

Satzinger, J., R. Jackson & S**.** Burd ***(***2002***).*** Systems Analysis and Design in a Changing World **(**2nd Ed.), Boston**,** MA***:*** Course Technology**.**

Schmitt (2006). Engaging a Rich Web Experience. Greenwich: Manning Publications Co.

Sekaran, U. (2003). Research Methods for Business: A skill building approach. New York : John Wiley & Sons.

Silva, P. & Paton, N. (2003) .*Unified Modeling Language for Interactive application* Retrievedfrom:http://scholar.google.com/scholar?q=UMLi:%20Unified%20Modeling%20language%20for%20Interactive%20Application&hl=en&lr=&oi=scholar.

Singh sumanjeet .(2009), *emergence of payment systems in the age of electronic commerce: the state of art.* Global Journal of International Business Research Vol. 2. No. 2.

Steven, J., & Templeton, E. (2008). *Detecting Spoofed Packets*, retrieved on 3 Oct 2009, U.C. Davis.

Sun (2007),*JavaServer Pages[tm] Technology* - White Paper. Retrieved from : http://java.sun.com/products/jsp/ White Paper /index.html

Sun technology. (2008). *Java Server Pages technology work, Retrieved from:* http://java.sun.com/products/jsp/ White Paper /index.html

Treiber, M. (2007). *Active Web Service Registries*. 11(5): 66 – 71.

Tunckan, E. (2007), *Structural Dimensions And Functions Of Student Centers In The Open Education Faculty Practices*, *TOJDE* October 2007 Volume: 8 Number: 4 Article 13 Retrieved December 15,2008 From http://tojde.anadolu.edu.tr/

Uebersax J. S. *(2006). Likert scales: Dispelling the confusion. Statistical Methods for Rater Agreement Web site.* Available at ourworld.compuserve.com/homepages/jsuebersax/likert2.htm. Accessed December 18, 2007.

Vaishnavi, V. & Kuechler, B. (2004). *Design Research in information system*. Retrieved 28, March, 2008, from http://www.isworld.org/Researchdesign/drisISworld.htm

Woodley.M & Kamin .S. K (2007) . *Programming Studio: A Course for Improving Programming Skills in Undergraduates.* SIGCSE'07, March 7–10, 2007, Covington, Kentucky, USA.

Yasuhiko, H., Tsutomu, A., and Syun, T. (2006). *Construction and operation of a registration system in a university*. 37(10): 69 – 80.

Zainal, A., and Hasibuan, H. (2005). *The Use of E-Learning towards New Learning Paradigm: Case Study Student Centered E-Learning Environment at Faculty of Computer Science-University of Indonesia*. 1026 – 1030.

Zane , V. (2004) . *Wireless Student Testing* , Proceeding of the International Conference on Pervasive Computer and Communication , Las Vegas , Nevada . Retrived on 14 may 2008.

APPENDIX (A)
INTERVIEW

Online Programming Courses Registration System (OPCRS)

MSc (IT)

College of Arts and Sciences

NAME: Saleh Sulaiman Affash Alqatan

E-MAIL: Saleh_q_2001@yahoo.com

CONTACT: 0175500916

SUPERVISOR: Rafidah Bt Abd Razak

About this questionnaire

This study aims to build a Prototype for the lectures and students, to help them by provide them with the appropriate learning services.

The purpose of this questionnaire is to help me gain an understanding of the user who will use Online Programming Courses Registration, and to get any additional feedback or comments about it. All the information you provide is confidential. Your name is not stored with this questionnaire, and the information you provide will not be used for any other purposes.

The main objective of this study is to improve the interaction between the students of FTM College in UUM and the professional Enrichment & Development Division (CoPED) Center. In order to achieve that the following objectives will be pursued:

- To identify the basic requirements for building an online registration system.
- To propose model of online registration system by using Java Server Page (JSP).
- To evaluate the user acceptance of the prototype.

A_1 : Introduction:

This questionnaire consists of 17 questions in two parts

1. General information.
2. System aspects.

Please answer **ALL** questions in **ALL** parts

<u>Part 1: General information</u>

1. **Gender:**

   [ ] male [ ] female

2. **Age:** ________ Years

3. **Educational background:**

[ ] Diploma [ ] Degree [ ] Master [ ] PhD

Part 2: System aspects

This part is intended to obtain your views on some aspects of the Online Programming Courses Registration System (OPCRS). Please mark [√] your answers.

1 = strongly disagree, 2 = Disagree, 3 = Natural, 4 = Agree, 5 = strongly agree

PERCEIVED USEFULNESS		1	2	3	4	5
1	**OPCRS** Prototype will enable the student to register programming courses and pay fees quickly.	○	○	○	○	○
2	Using **OPCRS** will reduce students effort					
3	Using **OPCRS** will reduce students time	○	○	○	○	○
4	Using **OPCRS** would easy to understand each of the links on the home page.					
5	I would find **OPCRS** secure.	○	○	○	○	○

PERCEIVED EASE OF USE		1	2	3	4	5
6	Learning to operate **OPCRS** would be easy for me.					
7	I would find it easy to get **OPCRS** to do what I want it to do.	○	○	○	○	○
8	interaction with **OPCRS** would be clear and Understandable.					
9	I would find **OPCRS** to be flexible to interact with.	○	○	○	○	○
10	I would find **OPCRS** easy to use.	○	○	○	○	○

ATTRIUTE OF USABILITY		1	2	3	4	5
11	I found it easy to interact between lecturer and his/her students by using **OPCRS**.					
12	The procedure through **OPCRS** prototype by online is clear.	○	○	○	○	○

USER SATISFACTION		1	2	3	4	5
13	I completely satisfied in using **OPCRS** prototype.					
14	I feel very confident in using **OPCRS** Prototype	. ○	○	○	○	○

\

Thank you for your cooperation and attention

A_2: Frequency Table

Question_4

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Neutral	5	16.7	16.7	16.7
	Agree	9	30.0	30.0	46.7
	Strongly Agree	16	53.3	53.3	100.0
	Total	30	100.0	100.0	

Question_5

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Disagree	3	10.0	10.0	10.0
	Neutral	6	20.0	20.0	30.0
	Agree	8	26.7	26.7	56.7
	Strongly Agree	13	43.3	43.3	100.0
	Total	30	100.0	100.0	

Question_6

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Disagree	4	13.3	13.3	13.3
	Neutral	7	23.3	23.3	36.7
	Agree	7	23.3	23.3	60.0
	Strongly Agree	12	40.0	40.0	100.0
	Total	30	100.0	100.0	

Question_7

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Disagree	1	3.3	3.3	3.3
	Neutral	8	26.7	26.7	30.0
	Agree	7	23.3	23.3	53.3
	Strongly Agree	14	46.7	46.7	100.0
	Total	30	100.0	100.0	

Question_8

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Disagree	1	3.3	3.3	3.3
	Neutral	6	20.0	20.0	23.3
	Agree	12	40.0	40.0	63.3
	Strongly Agree	11	36.7	36.7	100.0
	Total	30	100.0	100.0	

Question_9

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Disagree	3	10.0	10.0	10.0
	Neutral	4	13.3	13.3	23.3
	Agree	6	20.0	20.0	43.3
	Strongly Agree	17	56.7	56.7	100.0
	Total	30	100.0	100.0	

Question_10

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Disagree	1	3.3	3.3	3.3
	Neutral	6	20.0	20.0	23.3
	Agree	4	13.3	13.3	36.7
	Strongly Agree	19	63.3	63.3	100.0
	Total	30	100.0	100.0	

Question_11

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Disagree	3	10.0	10.0	10.0
	Neutral	9	30.0	30.0	40.0
	Agree	3	10.0	10.0	50.0
	Strongly Agree	15	50.0	50.0	100.0
	Total	30	100.0	100.0	

Question_12

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Disagree	6	20.0	20.0	20.0
	Neutral	7	23.3	23.3	43.3
	Agree	7	23.3	23.3	66.7
	Strongly Agree	10	33.4	33.4	100.0
	Total	30	100.0	100.0	

Question_13

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Disagree	2	6.7	6.7	6.7
	Neutral	6	20.0	20.0	26.7
	Agree	13	43.3	43.3	70.0
	Strongly Agree	9	30.0	30.0	100.0
	Total	30	100.0	100.0	

Question_14

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly disagree	1	3.3	3.3	3.3
	Disagree	2	6.7	6.7	10.0
	Neutral	5	16.7	16.7	26.7
	Agree	13	43.3	43.3	70.0
	Strongly Agree	9	30.0	30.0	100.0
	Total	30	100.0	100.0	

Question_15

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly disagree	1	3.3	3.3	3.3
	Disagree	2	6.7	6.7	10.0
	Neutral	5	16.7	16.7	26.7
	Agree	12	40.0	40.0	66.7
	Strongly Agree	10	33.3	33.3	100.0
	Total	30	100.0	100.0	

Question_16

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Disagree	2	6.7	6.7	6.7
	Neutral	4	13.3	13.3	20.0
	Agree	5	16.7	16.7	36.7
	Strongly Agree	19	63.3	63.3	100.0
	Total	30	100.0	100.0	

Question_17

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Disagree	1	3.3	3.3	3.3
	Neutral	5	16.7	16.7	20.0
	Agree	23	76.7	76.7	96.7
	Strongly Agree	1	3.3	3.3	100.0
	Total	30	100.0	100.0	

A_E : Reliability Analysis

A) RELIABILITY /VARIABLES=Question_4 Question_5 Question_6 Question_7 Question_8.

Reliability Statistics

Cronbach's Alpha	N of Items
.704	5

B) RELIABILITY /VARIABLES=Question_9 Question_10 Question_11 Question_12 Question_13.

Reliability Statistics

Cronbach's Alpha	N of Items
.704	5

C) RELIABILITY /VARIABLES=Question_14 Question_15.

Reliability Statistics

Cronbach's Alpha	N of Items
.864	2

D) RELIABILITY /VARIABLES=Question_16 Question_17.

Reliability Statistics

Cronbach's Alpha	N of Items
.770	2

APPENDIX (B)
INTERVIEW

UNIVERSITY UTARA MALAYSIA

Computing Professional and Development Division

College of Arts and Science

To Whom It May Concern

Dear Sir/Madam

After the briefing and discussion about the student proposal (Saleh Sulaimen) the online programming courses registration system for FTM faculty by using Mobile , we collected some ideas presented in the discussion. This website will be useful the students when them respond with the website.

However, we confirm that this proposal will benefit and useful to both parties (students and staff (COPED)) in order to improve the website for (COPED center) and enhance the interaction, communication between the students and the staff(COPED).

Program Manager

HAIRULNIZA ABD. RAHMAN
PENSYARAH
UUM College of Arts and Sciences
Universiti Utara Malaysia

1- Are you face any difficult to announcement in the programming courses?

Yes .

2- Are you face any difficult for registration the students in the courses?

Yes .

3- Are you think this prototype for website will help to inform the class members if the lecturer absents in their classes?

Yes .

4- Are you think this Prototype will increase number of registered students in these courses, what your opinion about that?

Yes .

5- Are you think this prototype will improve the process of registration courses, what your opinion about that?

Yes .

6- Are you think this prototype will enhance the performance of staff and introduce the facilities and efficient interaction between this center and students, what your opinion about that?

Yes . more faster .

7- Are you think this prototype will introduce the facilities for the students in payment the fees for the programming courses, what your opinion about that?

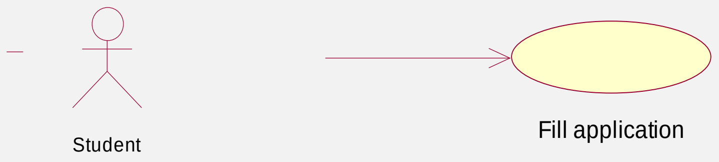
Yes . Easier for them .

HAIRULNIZA ABD. RAHMAN
PENSYARAH
UUM College of Arts and Sciences
Universiti Utara Malaysia

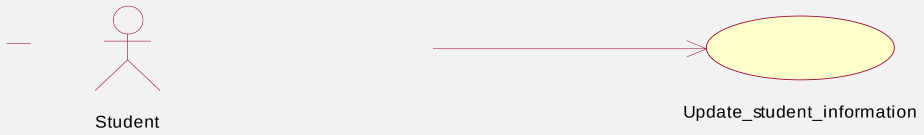
APPENDIX (C)
USE CASE SPECIFICATION

1. Use Case Specifications

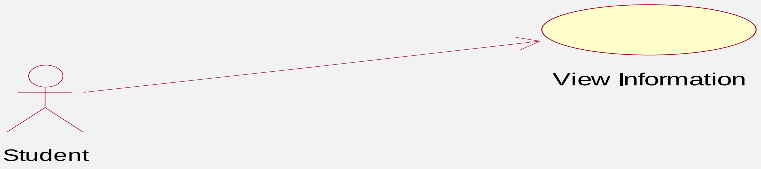
1.1. USE CASE: FILL APPLICATION

	
Brief Description	• This use case is initiated by the student. This use case will enable the student to fill the application forms and to apply.
Characteristic of Activation	• Event Driven by (Student).
Basic Flow	• This use case begins when the student wants to fill application by press registration Link. • The web site wills respond to the student order and will open the application page. • After that the student will fill all the required information in application form such as metric no, and student name, selected programming courses, password, and Retype password. • Student will press Submit button.
Alternative Flow	• The student may press back link to cancel the application form, which will close the page of the application form. • The student may press Reset button to empty all fields in case of wrong data entered.
Exceptional Flow	• When the student presses Submit button the system will display an error message as "the birth of date is required to be filed", as a validation**,** in case of missing any field to be filled.
Pre-condition	• Student must click the link of Apply for programming courses registration to open the application form.
Post-condition	• All required field must be filled correctly in valid data format to avoid any error messages.
RULE(S)	• All required field must be filled correctly in valid data format to avoid any error messages.

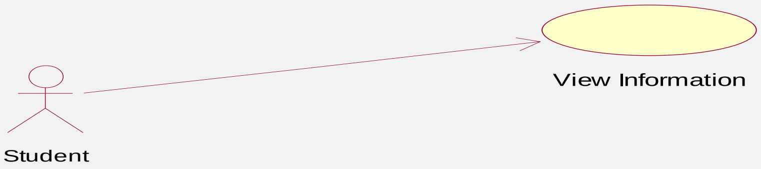
1.2. USE CASE: UPDATE STUDENT INFORMATION

 	
Brief Description	• This use case is initiated by the Student. This use case will enable the Student to update his/her information.
Characteristic of Activation	• Event Driven (Student).
Basic Flow	• This use case is initiated by the Student when they want update his/her information by press Update Information link. • The web site will respond to the student and will open the Update student information page. • The student insert new information such as student number, student name, IC/NO , program , semester and password. • After finish the student can press submit link.
Alternative Flow	• The student may press back link to cancel the application form, which will close the page of the application form. • The student may press Reset button to empty all fields in case of wrong data entered.
Exceptional Flow	• None
Pre-condition	• The Student must be already logged into his/her account.
Post-condition	• His/her information must update.
RULE(S)	• None

1.3. USE CASE: view information

	
Brief Description	- This use case is initiated by the Student. This use case will enable the Student to View his/her information.
Characteristic of Activation	- Event Driven (Student).
Basic Flow	- This use case is initiated by the Student when they want View his/her information by press View Information link. - The web site will respond to the student and will open the View Student Information page.
Alternative Flow	- None
Exceptional Flow	- None
Pre-condition	- The Student must be already logged into his/her account.
Post-condition	- The system must to appear his/her information.
RULE(S)	- None

1.4. USE CASE: REGISTER COURSES

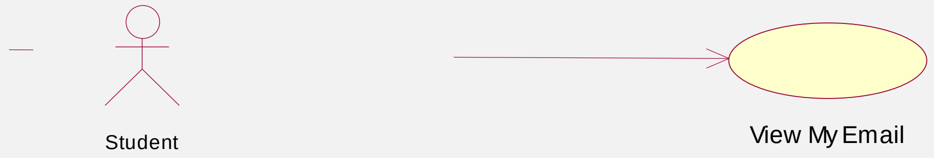


Brief Description	- This use case is initiated by the student. This use case will enable the student to register programming courses.
Characteristic of Activation	- Event Driven (Student).
Basic Flow	- This use case is initiated by the Student when they want Register Programming courses by press Register Course link. - The web site will respond to the student and will open the Register course page. - The student will insert Course Code that he/she want register it. - After that the student will press submit button, the system will respond by show course name and credit on course Information page. - And then the student will press submit button to register programming course, the system will respond by show message ("course registered successfully").
Alternative Flow	- The student may press back link to cancel the application form, which will close the page of the application form. - The student may press Reset button to empty all fields in case of wrong data entered.
Exceptional Flow	- The student may insert wrong course code, the system will respond by show message ("Invalid course code ").
Pre-condition	- The Student must be already logged into his/her account.
Post-condition	- The system must show information about this programming courses (course name and credit).
RULE(S)	- The course code should be valid.

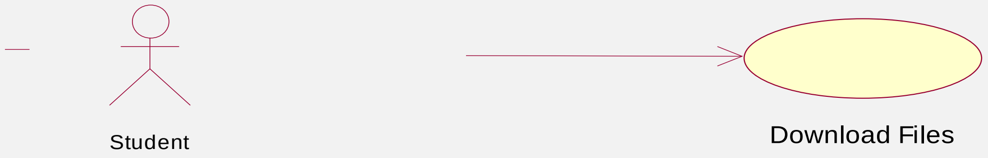
1.5. USE CASE: SEARCH COURSES

Student search course	
Brief Description	• This use case is initiated by the student. This use case will enable the student to search programming course.
Characteristic of Activation	• Event Driven (Student).
Basic Flow	• This use case is initiated by the Student when they want search Programming courses by press search Course link. • The web site will respond to the student and will open the search page. • The student will insert Course Code that he/she want search about it. • After that the student will press submit button, the system will respond by show course name and credit.
Alternative Flow	• The student may press back link to cancel the application form, which will close the page of the application form. • The student may press Reset button to empty all fields in case of wrong data entered.
Exceptional Flow	• The student may insert wrong course code, the system will respond by show message (“Invalid course code “).
Pre-condition	• The Student must be already logged into his/her account.
Post-condition	• The course code should be valid.
RULE(S)	• The course code should be valid.

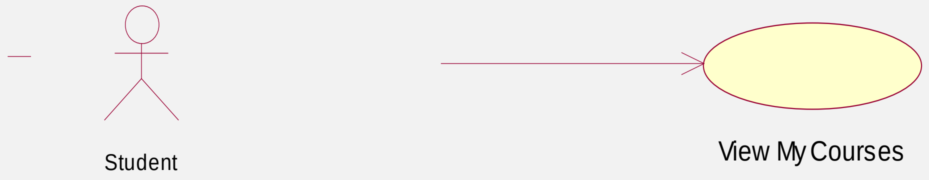
1.6. USE CASE: VIEW MY E-MAIL

	
Brief Description	• This use case is initiated by the student. This use case will enable the student to view his/her messages.
Characteristic of Activation	• Event Driven (Student).
Basic Flow	• This use case is initiated by the Student when they want View his/her message by press View my message link. • The web site will respond to the student and will open the View email page that contains sender, message subject and message content.
Alternative Flow	• None
Exceptional Flow	• None
Pre-condition	• The Student must be already logged into his/her account.
Post-condition	• None
RULE(S)	• None

1.7. USE CASE: DOWNLOAD FILES

	
Brief Description	• This use case is initiated by the student. This use case will enable the student to download files which upload lecturer for him/her.
Characteristic of Activation	• Event Driven (Student).
Basic Flow	• This use case is initiated by the Student when they want download his/her files by press download files link. • The web site will respond to the student and will open the view course page that contains files name. • The student will press file name which he want download it, the system will respond by open Download page that contains file title, file description and button to download this file. • The student will press download button to download this file.
Alternative Flow	• The student may press back link to cancel the application form, which will close the page of the application form.
Exceptional Flow	• None
Pre-condition	• The Student must be already logged into his/her account.
Post-condition	• The system must allow to student to determine file location
RULE(S)	• None

1.8. USE CASE: VIEW MY COURSES

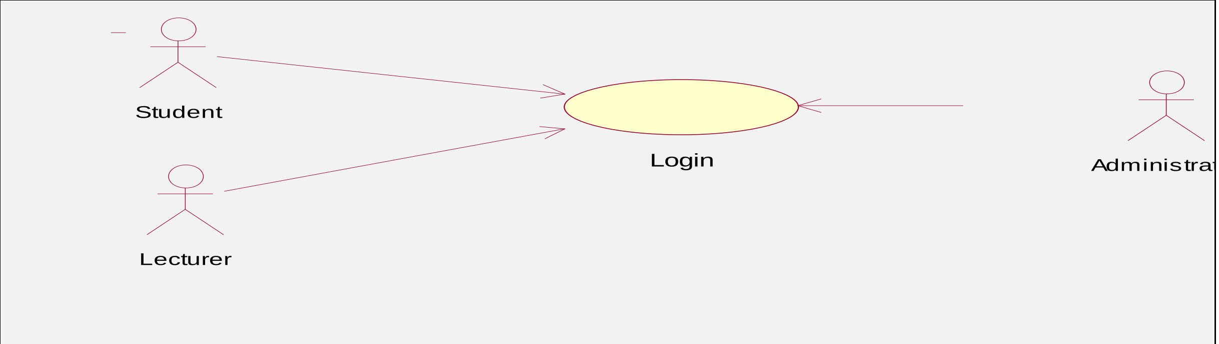
	
Brief Description	• This use case is initiated by the student. This use case will enable the student to view his/her course which he paid fees for it.
Characteristic of Activation	• Event Driven (Student).
Basic Flow	• This use case is initiated by the Student when they want View his/her courses by press View my course link. • The web site will respond to the student and will open the View courses page that contains programming course which he already paid fees it.
Alternative Flow	• None
Exceptional Flow	• None
Pre-condition	• The Student must be already logged into his/her account.
Post-condition	• The system must show all programming course which paid fees it only
RULE(S)	• None

1.9. USE CASE: PAY FEES

  Student View My Courses	
Brief Description	• This use case is initiated by the student. This use case will enable student to pay fees for his/ her programming course which he already registered it.
Characteristic of Activation	• Event Driven (Student).
Basic Flow	• This use case is initiated by the Student when they want pay fees for his/her programming courses by press Payment link. • The web site will respond to the student and will open the View courses page which he already registered it to pay fee it. • The system will press pay link, the system will respond by open payment page. • After that the student will insert his/her credit number. • The student will press submit, the system will verify for credit number and, the system will respond by open Deposit page. • The student presses submit. • The student inserts deposit value. • After that the student will press submit, the system will respond by showing message ('your payment has been done successfully").
Alternative Flow	• The student may press back link to cancel the application form, which will close the page of the application form. • The student may press Reset button to empty all fields in case of wrong data entered.
Exceptional Flow	• The student may enter wrong account number, the system will display message "Invalid Credit No". • The student may do not have enough balance, which will display an error message "No enough balance".

Pre-condition	- The Student must be already logged into his/her account.
Post-condition	- The system must show all programming course which paid fees it only
RULE(S)	- The system must update his/her balance.

1.10. USE CASE: LOGIN

	
Brief Description	- This use case is initiated when the (Student or Lecturer or Administrator) wants to enter to their accounts to use the task that they are able to do.
Characteristic of Activation	- Event Driven (Student or Lecturer or Administrator).
Basic Flow	- This use case begins when The (Student or Lecturer or Administrator) to access their accounts by press student Area Link or lecturer Area Link or Administrator Area Link, the system will respond by open Login Panel page. - The user (student or lecturer or Administrator) insert his/her Username and Password in the login page, the system will verify the username and password. - The system will respond by open Main Page for Student or lecturer or administrator

Alternative Flow	- The student may press Reset button to empty all fields in case of wrong data entered. - The student may press main link to cancel the application form, which will return to Home page.
Exceptional Flow	- The student may enter wrong combination of username and password in the login page and press login button, the system will display an error message stating "Invalid username or password".
Pre-condition	- The (Student or Lecturer or Administrator) must have their login information (correct Username and Password).
Post-condition	- In case of accepting login a new page should be viewed to the student his/her account. - In case of invalid user name or password a message should be viewed to alert him - In case of null user name a message should be viewed to alert him of not entering
RULE(S)	- Only the authorized users can login to their accounts by inserting the correct username and password.

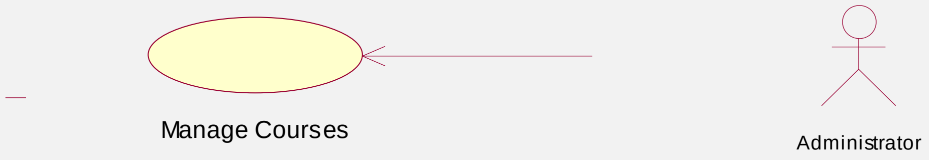
1.11. USE CASE: Send Message

Lecturer Send Message	
Brief Description	• This use case is initiated when the Lecturer wants to send E-mail to his/her Student.
Characteristic of Activation	• Event Driven (Lecturer).
Basic Flow	• This use case is initiated by the lecturer when they want send his/her student by press Send message link, then the system will respond by open message page. • The Lecturer will insert message subject and description. • The Lecturer will send the message by press send message button.
Alternative Flow	• The student may press back link to cancel the application form, which will close the page of the application form.
Exceptional Flow	• None
Pre-condition	• The Lecturer must be already logged into his/her account.
Post-condition	• The system must send message to all his/her student.
RULE(S)	• None.

1.12. USE CASE: Upload File

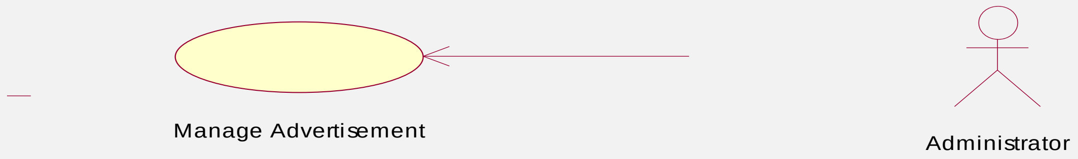
Lecturer Loading Files	
Brief Description	• This use case is initiated when the Lecturer wants to load files such as PDF to his/her Student.
Characteristic of Activation	• Event Driven (Lecturer).
Basic Flow	• This use case is initiated by the Lecturer when they want upload files such as DFP or Doc to his/her student by press Loading Files link, then the system will respond by open upload page. • The Lecturer will selecting files that he want load to students in his/her class by press click here to upload file Link. • The lecturer inserts title and description for this file. • The lecturer select course numbers which he wants upload this file for it. • The lecturer will press submit button.
Alternative Flow	• The student may press back link to cancel the application form, which will close the page of the application form.
Exceptional Flow	• The Lecturer may press Cancel button to cancel load those files, which will close the page of the load form
Pre-condition	• The Lecturer must be already logged into his/her account.
Post-condition	• The lecturer may do loading files large size (more than 10MB), which will display an alter message
RULE(S)	• The system must load those files into database.

1.13. USE CASE: MANAGE COURSES

 	
Brief Description	• This use case is initiated when the administrate wants to Manage programming courses.
Characteristic of Activation	• Event Driven (Administrator).
Basic Flow	• This use case is initiated by the administrator when they want Manage programming courses by press courses link, then the system will respond by open courses page. • The administrators can new programming courses by press new link, the system will respond by open new course page. • The administrator will insert programming course information such as course number, course name, hour, description and price. • The administrator will press submit, the system will return back to Main course page.

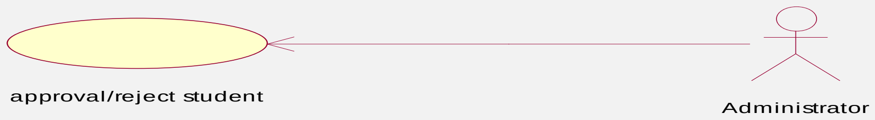
Alternative Flow	- The administrators can update programming courses information by press Update link, the system will respond by open update course page. - The administrators can Delete programming course by press Delete link, the system will respond by open Delete course page - The administrators can view programming courses by press View link, the system will respond by open View course page. - The administrators may press Main link to return to main page, which will close the page of the application form. - The administrators may press Logout link to exist from his/her account, which will close the page of the application form. - The administrators may press back link to cancel the application form, which will close the page of the application form. - The administrators may press Reset button to empty all fields in case of wrong data entered.
Exceptional Flow	- when the administrator presses Add new programming course button the system will display an error message as “the course name is required to be filed”, as a validation**,** in case of missing any field to be filled.
Pre-condition	- The administrator must be already logged into his/her account.
Post-condition	- The system will create new record in database.
RULE(S)	- None

1.14. USE CASE: MANAGE ADVERTISEMENT

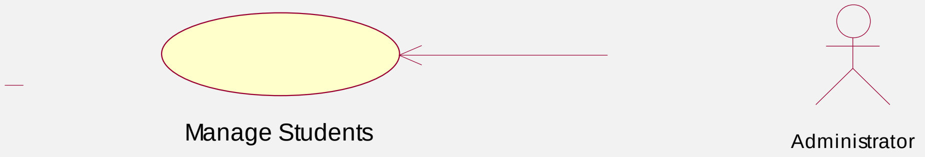
 	
Brief Description	• This use case is initiated when the administrate wants to make Advertisement for new programming course.
Characteristic of Activation	• Event Driven (Administrator).
Basic Flow	• This use case is initiated by the administrator when they want Manage Advertisement by press Advertisement link, then the system will respond by open Advertisement page. • The administrators can new Advertisement by press new link, the system will respond by open new Advertisement page. • The administrator will insert Advertisement information such as Advertisement title, description. • The administrator will press submit, the system will return back to main Advertisement page.

Alternative Flow	• The administrators can Advertisement courses information by press Update link, the system will respond by open update Advertisement page. • The administrators can Delete Advertisement by press Delete link, the system will respond by open Delete Advertisement page. • The administrators can Advertisement by press View link, the system will respond by open View Advertisement page. • The administrators may press Main link to return to main page, which will close the page of the application form. • The administrators may press Logout link to exist from his/her account, which will close the page of the application form. • The administrators may press back link to cancel the application form, which will close the page of the application form. • The administrators may press Reset button to empty all fields in case of wrong data entered.
Exceptional Flow	• When the administrator presses Add Advertisement link the system will display an error message as "Advertisement description is required to be filed", as a validation, in case of missing any field to be filled.
Pre-condition	• The administrator must be already logged into his/her account.
Post-condition	• The system will create new record in database.
RULE(S)	• None

1.15. USE CASE: Approval Application

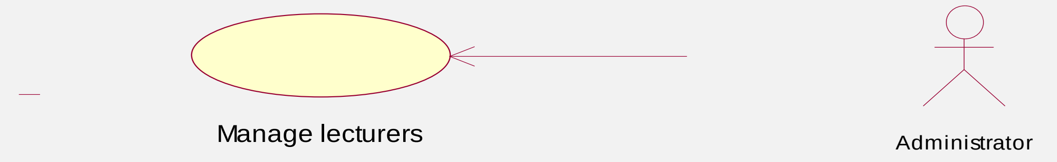
	
Brief Description	• This use case is initiated by the administrator when he receives the students applications on his/her account
Characteristic of Activation	• Event Driven (Administrator).
Basic Flow	• This use case is initiated by the administrator when they want approve/reject to student application by press Approve/Reject Student link, then the system will respond by open Approve/Reject page. • The administrators can make approval to student application by press Approve link and student status will change to Approve.
Alternative Flow	• The administrators can View student application by press View link, the system will respond by open view page. • The administrators may press Main link to return to main page, which will close the page of the application form. • The administrators may press Logout link to exist from his/her account, which will close the page of the application form. • The administrators may press back link to cancel the application form, which will close the page of the application form. • The administrators may press Reset button to empty all fields in case of wrong data entered.
Exceptional Flow	• None
Pre-condition	• The administrator must be already logged into his/her account.
Post-condition	• The system will update student status.
RULE(S)	• None

1.16. USE CASE: MANAGE STUDENTS

	
Brief Description	• This use case is initiated when the administrate wants to Manage Students
Characteristic of Activation	• Event Driven (Administrator).
Basic Flow	• This use case is initiated by the administrator when they want Manage Students information by press Student link, then the system will respond by open student page. • The administrators can add new Student by press new link, the system will respond by open new Student page. • The administrator will insert Student information such as Student number, student name, IC/No, program, select Semester and password. • The administrator will press submit, the system will return back to Main student page.

Alternative Flow	• The administrators can update student information by press Update link, the system will respond by open update student page. • The administrators can Delete student by press Delete link, the system will respond by open Delete student page. • The administrators can view student by press View link, the system will respond by open View student page. • The administrators may press Main link to return to main page, which will close the page of the application form. • The administrators may press Logout link to exist from his/her account, which will close the page of the application form. • The administrators may press back link to cancel the application form, which will close the page of the application form. • The administrators may press Reset button to empty all fields in case of wrong data entered.
Exceptional Flow	• when the administrator presses Add new Student button the system will display an error message as "the student name is required to be filed", as a validation, in case of missing any field to be filled
Pre-condition	• The administrator must be already logged into his/her account.
Post-condition	• The system will create new record in database.
RULE(S)	• None

1.17. USE CASE: MANAGE LECTURERS

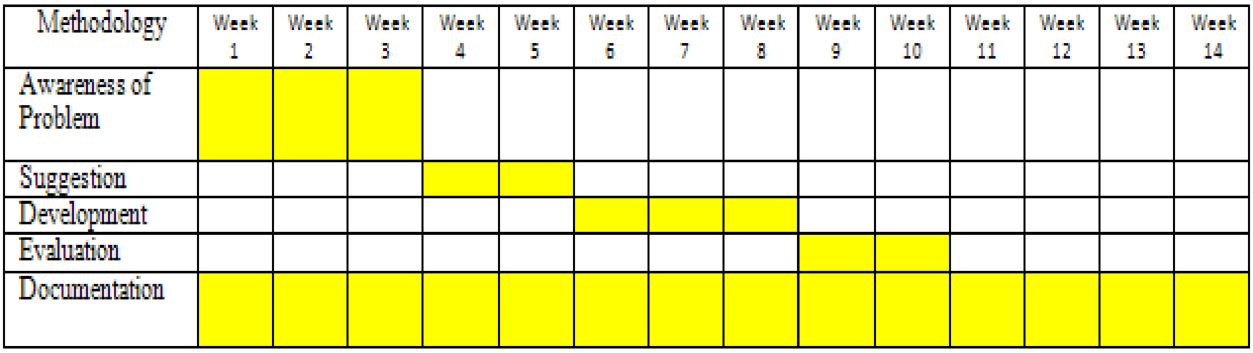
	
Brief Description	• This use case is initiated when the administrate wants to Manage lecturers.
Characteristic of Activation	• Event Driven (Administrator).
Basic Flow	• This use case is initiated by the administrator when they want Manage lecturer information by press lecturer link, then the system will respond by open lecturer page. • The administrators can new lecturer by press new link, the system will respond by open new lecturer page. • The administrator will insert lecturer information such as lecturer number, lecturer name and password. • The administrator will press submit, the system will return back to Main student page.

Alternative Flow	• The administrators can update lecturer information by press Update link, the system will respond by open update lecturer page. • The administrators can Delete lecturer by press Delete link, the system will respond by open Delete lecturer page. • The administrators can view lecturer by press View link, the system will respond by open View lecturer page. • The administrators may press Main link to return to main page, which will close the page of the application form. • The administrators may press Logout link to exist from his/her account, which will close the page of the application form. • The administrators may press back link to cancel the application form, which will close the page of the application form. • The administrators may press Reset button to empty all fields in case of wrong data entered.
Exceptional Flow	• when the administrator presses Add new lecturer button the system will display an error message as “the lecturer name is required to be filed”, as a validation, in case of missing any field to be filled.
Pre-condition	• The administrator must be already logged into his/her account.
Post-condition	• The system will create new record in database.
RULE(S)	• None

APPENDIX D

Project Work Flow

Project Work Flow

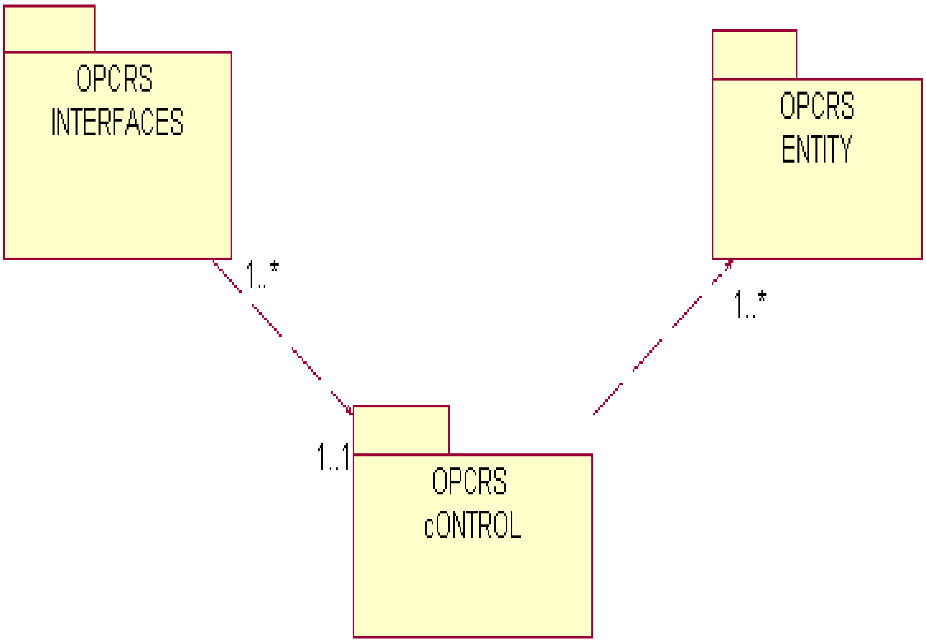


Methodology	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14
Awareness of Problem	🟨	🟨	🟨											
Suggestion				🟨	🟨									
Development						🟨	🟨	🟨						
Evaluation									🟨	🟨				
Documentation	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨

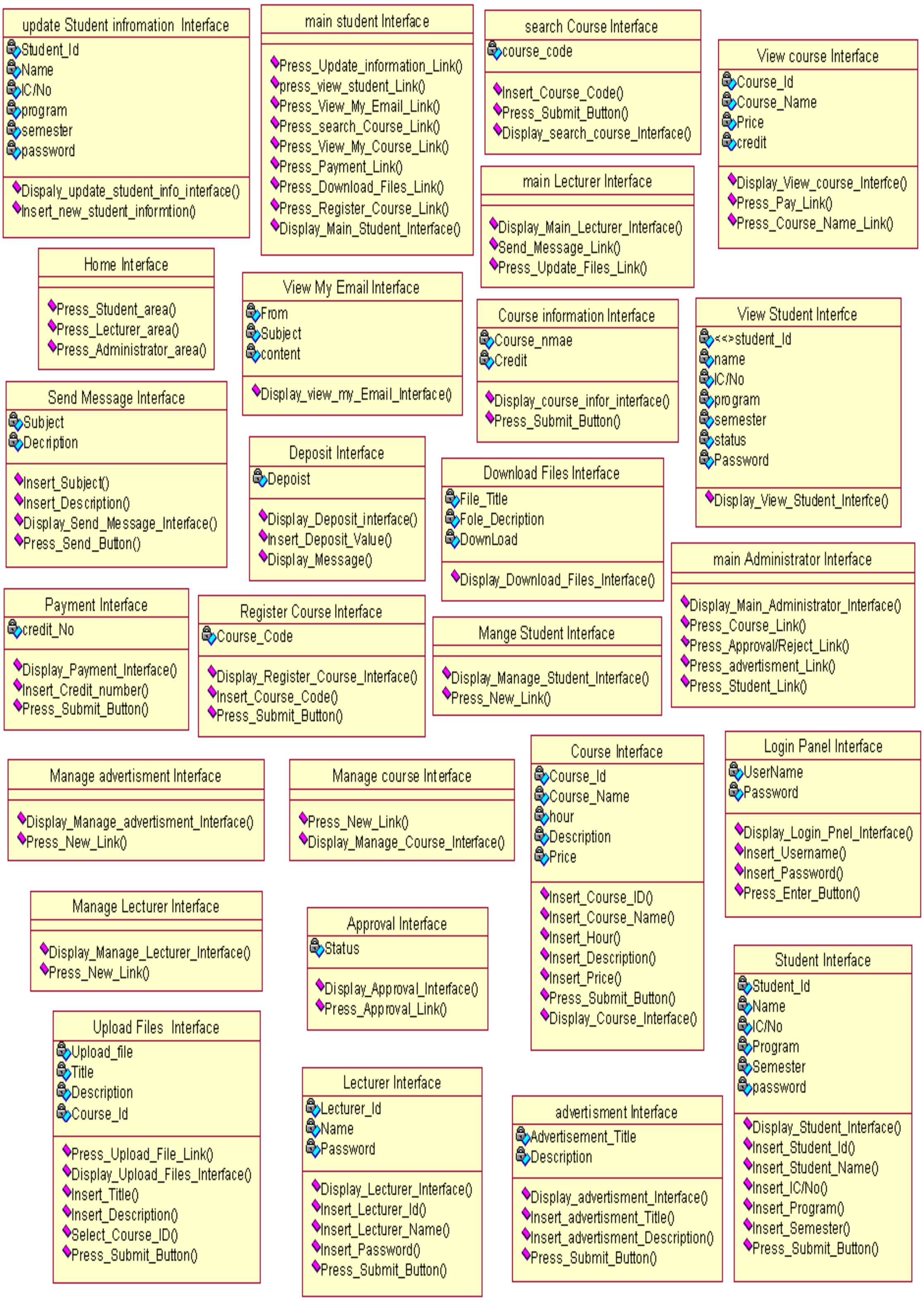
APPENDIX E

Class Diagram

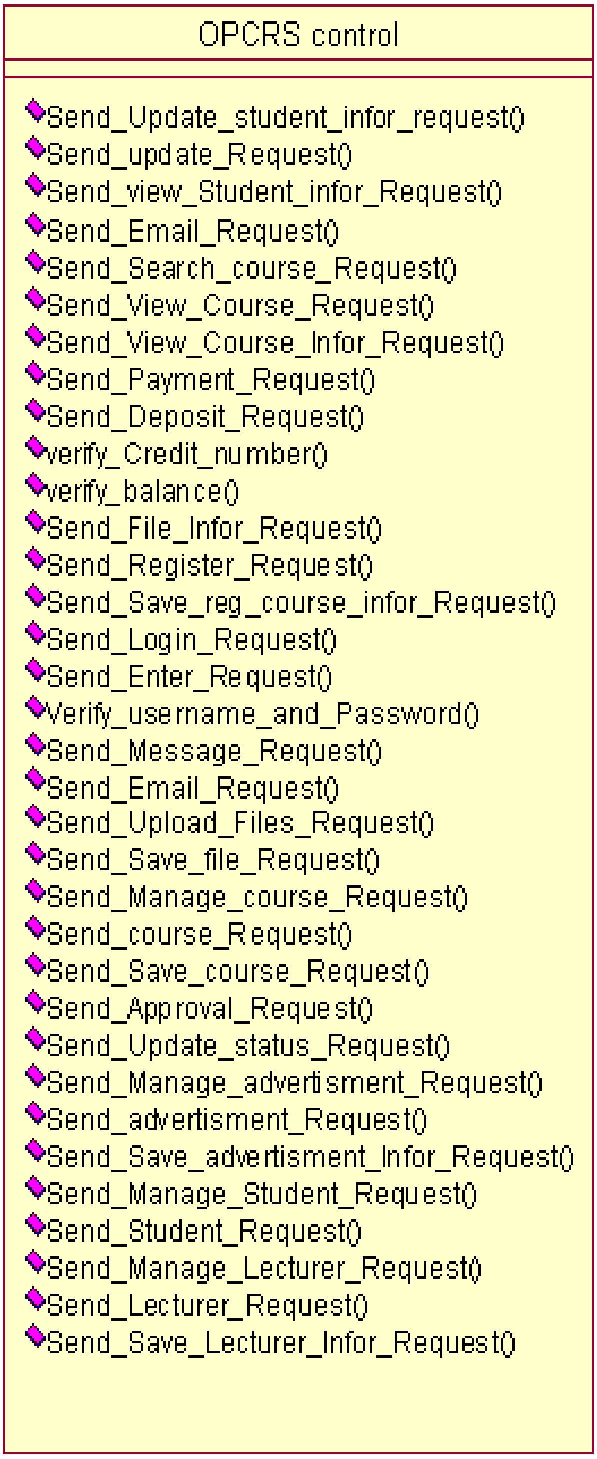
Package Diagram



OPCRS Interface

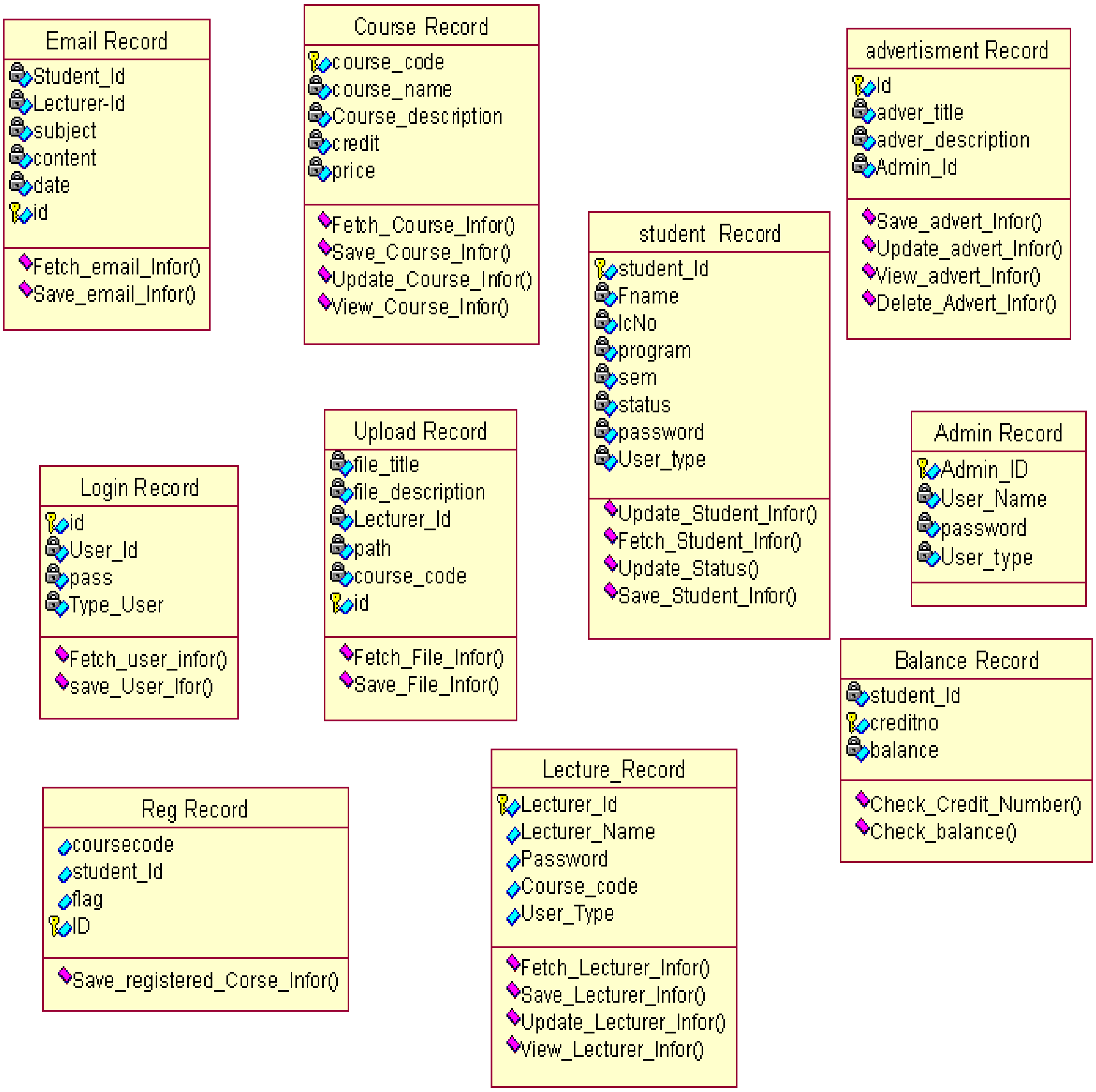

update Student infromation Interface
Student_Id
Name
IC/No
program
semester
password
Dispaly_update_student_info_interface()
Insert_new_student_informtion()
main student Interface
Press_Update_information_Link()
press_view_student_Link()
Press_View_My_Email_Link()
Press_search_Course_Link()
Press_View_My_Course_Link()
Press_Payment_Link()
Press_Download_Files_Link()
Press_Register_Course_Link()
Display_Main_Student_Interface()
search Course Interface
course_code
Insert_Course_Code()
Press_Submit_Button()
Display_search_course_Interface()
View course Interface
Course_Id
Course_Name
Price
credit
Display_View_course_Interfce()
Press_Pay_Link()
Press_Course_Name_Link()
main Lecturer Interface
Display_Main_Lecturer_Interface()
Send_Message_Link()
Press_Update_Files_Link()
Home Interface
Press_Student_area()
Press_Lecturer_area()
Press_Administrator_area()
View My Email Interface
From
Subject
content
Display_view_my_Email_Interface()
Course information Interface
Course_nmae
Credit
Display_course_infor_interface()
Press_Submit_Button()
View Student Interfce
<<>student_Id
name
IC/No
program
semester
status
Password
Display_View_Student_Interfce()
Send Message Interface
Subject
Decription
Insert_Subject()
Insert_Description()
Display_Send_Message_Interface()
Press_Send_Button()
Deposit Interface
Depoist
Display_Deposit_interface()
Insert_Deposit_Value()
Display_Message()
Download Files Interface
File_Title
Fole_Decription
DownLoad
Display_Download_Files_Interface()
main Administrator Interface
Display_Main_Administrator_Interface()
Press_Course_Link()
Press_Approval/Reject_Link()
Press_advertisment_Link()
Press_Student_Link()
Payment Interface
credit_No
Display_Payment_Interface()
Insert_Credit_number()
Press_Submit_Button()
Register Course Interface
Course_Code
Display_Register_Course_Interface()
Insert_Course_Code()
Press_Submit_Button()
Mange Student Interface
Display_Manage_Student_Interface()
Press_New_Link()
Course Interface
Course_Id
Course_Name
hour
Description
Price
Insert_Course_ID()
Insert_Course_Name()
Insert_Hour()
Insert_Description()
Insert_Price()
Press_Submit_Button()
Display_Course_Interface()
Login Panel Interface
UserName
Password
Display_Login_Pnel_Interface()
Insert_Username()
Insert_Password()
Press_Enter_Button()
Manage advertisment Interface
Display_Manage_advertisment_Interface()
Press_New_Link()
Manage course Interface
Press_New_Link()
Display_Manage_Course_Interface()
Manage Lecturer Interface
Display_Manage_Lecturer_Interface()
Press_New_Link()
Approval Interface
Status
Display_Approval_Interface()
Press_Approval_Link()
Upload Files Interface
Upload_file
Title
Description
Course_Id
Press_Upload_File_Link()
Display_Upload_Files_Interface()
Insert_Title()
Insert_Description()
Select_Course_ID()
Press_Submit_Button()
Lecturer Interface
Lecturer_Id
Name
Password
Display_Lecturer_Interface()
Insert_Lecturer_Id()
Insert_Lecturer_Name()
Insert_Password()
Press_Submit_Button()
advertisment Interface
Advertisement_Title
Description
Display_advertisment_Interface()
Insert_advertisment_Title()
Insert_advertisment_Description()
Press_Submit_Button()
Student Interface
Student_Id
Name
IC/No
Program
Semester
password
Display_Student_Interface()
Insert_Student_Id()
Insert_Student_Name()
Insert_IC/No()
Insert_Program()
Insert_Semester()
Press_Submit_Button()

OPCRS Control



OPCRS control
Send_Update_student_infor_request()
Send_update_Request()
Send_view_Student_infor_Request()
Send_Email_Request()
Send_Search_course_Request()
Send_View_Course_Request()
Send_View_Course_Infor_Request()
Send_Payment_Request()
Send_Deposit_Request()
verify_Credit_number()
verify_balance()
Send_File_Infor_Request()
Send_Register_Request()
Send_Save_reg_course_infor_Request()
Send_Login_Request()
Send_Enter_Request()
Verify_username_and_Password()
Send_Message_Request()
Send_Email_Request()
Send_Upload_Files_Request()
Send_Save_file_Request()
Send_Manage_course_Request()
Send_course_Request()
Send_Save_course_Request()
Send_Approval_Request()
Send_Update_status_Request()
Send_Manage_advertisment_Request()
Send_advertisment_Request()
Send_Save_advertisment_Infor_Request()
Send_Manage_Student_Request()
Send_Student_Request()
Send_Manage_Lecturer_Request()
Send_Lecturer_Request()
Send_Save_Lecturer_Infor_Request()

OPCRS Entity


Email Record
Student_Id
Lecturer-Id
subject
content
date
id
Fetch_email_Infor()
Save_email_Infor()
Course Record
course_code
course_name
Course_description
credit
price
Fetch_Course_Infor()
Save_Course_Infor()
Update_Course_Infor()
View_Course_Infor()
advertisment Record
Id
adver_title
adver_description
Admin_Id
Save_advert_Infor()
Update_advert_Infor()
View_advert_Infor()
Delete_Advert_Infor()
student Record
student_Id
Fname
IcNo
program
sem
status
password
User_type
Update_Student_Infor()
Fetch_Student_Infor()
Update_Status()
Save_Student_Infor()
Upload Record
file_title
file_description
Lecturer_Id
path
course_code
id
Fetch_File_Infor()
Save_File_Infor()
Login Record
id
User_Id
pass
Type_User
Fetch_user_infor()
save_User_Ifor()
Admin Record
Admin_ID
User_Name
password
User_type
Balance Record
student_Id
creditno
balance
Check_Credit_Number()
Check_balance()
Reg Record
coursecode
student_Id
flag
ID
Save_registered_Corse_Infor()
Lecture_Record
Lecturer_Id
Lecturer_Name
Password
Course_code
User_Type
Fetch_Lecturer_Infor()
Save_Lecturer_Infor()
Update_Lecturer_Infor()
View_Lecturer_Infor()