

# **Automation of Blanket Credit Registration**

by

Shwetha Somashekar

A REPORT

submitted to

Oregon State University

in partial fulfillment of  
the requirements for the  
degree of

Master of Science

In

Computer Science

Presented 2006

Commencement June 2007

## AN ABSTRACT OF THE PROJECT OF

Shwetha Somashekar for the degree of Master of Science in Computer Science presented  
on \_\_\_\_\_.

Title: Automation of Blanket Credit Registration

Abstract Approved: \_\_\_\_\_

Timothy Budd

Almost every student in the School of EECS undergoes the process of *Blanket Credit Registration* wherein the student has to fill out the form for registration, meet the concerned professor and obtain his approval. The staff of the department has to maintain the details of the student and the Professor who has approved his/her registration and remind the faculty to grade the student at the end of every quarter. The goal of this project is to automate the currently manual registration process. In particular, our proposed system provides the students with the list of Courses that they could register and the list of Professors they could contact. Once the student makes the choice the system generates an email that is sent to the Professor. The email has a link that the Professor would use to approve/disapprove the request. If the Professor approves the request, an email is sent to the student informing him/her about the approval and the details of the Professor and the Student are stored in a database. The staff are also notified about the approval. If the Professor rejects the request, the student is notified about the decision. At the end of the quarter, the staff would login to the system and choose a Professor and the system would send a reminder email to the Professor with the list of students registered with him/her enabling the Professor to grade the students. This system was tested on a small sample of faculty, students and the staff in the school of EECS.

## **ACKNOWLEDGEMENTS**

I express my sincere thanks to my academic advisor and major professor, Dr. Timothy A. Budd for his tremendous support and the constant encouragement he provided without which none of this would have been possible. I am particularly thankful to him for guiding me throughout this project.

I express my gratitude towards Dr. Bella Bose and Dr. Carlos Jensen for sparing their valuable time, willing to serve on my committee. I would also like to thank the CS office staff for their help over the past 2 years.

I am grateful to Dr. Margaret Burnett, Dr. Mike Bailey, Nancy, Paul, Tyler, Kavitha, Vidya and Flora for evaluating my system and providing me with their valuable feedback. I greatly appreciate Shoan for providing me with the required information about the tools that are critical to this project.

I am also thankful to the wonderful set of friends that I had over the past two years including Vidya, Kavitha, Vasu, Feni, Vaishu and all others for their support and for making me enjoy my graduate school life to the fullest. Last but not the least I am indebted to Sriraam for his constant support, tremendous encouragement and motivation that helped me in completing my tasks effectively.

My special thanks to my family, mom, dad and my brother for their love and their invaluable and unflinching support.

# TABLE OF CONTENTS

CHAPTER 1 .....	1
Introduction.....	1
1.1 Existing System .....	1
1.2 Issues in the Existing System.....	6
1.3 Organization of the report.....	9
CHAPTER 2 .....	10
Project Requirements.....	10
2.1 User Roles and Responsibilities.....	10
2.1.1 Student .....	10
2.1.2 Faculty.....	11
2.1.3 Staff.....	12
2.2 Functional Requirements .....	12
2.2.1 Requirements pertaining to the Students .....	12
2.2.2 Requirements pertaining to the Faculty .....	13
2.2.3 Requirements pertaining to the Staff .....	13
2.3 Objective.....	14
CHAPTER 3 .....	15
System Architecture.....	15
3.1 Model View Controller (MVC) pattern .....	15
3.1.1 Overview.....	15
3.1.2 Advantages.....	17
3.1.3 Drawbacks of MVC .....	18
3.2 Detailed Architecture .....	18
3.3 Tools .....	19
3.3.1 Smarty .....	19
3.3.2 Propel .....	21

CHAPTER 4 .....	29
Implementation .....	29
4.1 Students.....	29
4.2 Faculty.....	32
4.3 Staff.....	35
 CHAPTER 5 .....	 37
Evaluation .....	37
5.1 User Evaluation.....	37
5.2 Experimental Setup.....	37
5.3 Results.....	38
5.3.1 Staff.....	38
5.3.2 Students.....	39
5.3.3 Faculty.....	40
 CHAPTER 6 .....	 41
Scripting Languages.....	41
6.1 PHP .....	41
6.2 PHP vs ASP .....	42
6.3 PHP vs JSP.....	43
 CHAPTER 7 .....	 45
Conclusion .....	45

## TABLE OF FIGURES

Figure 1: Link to access the forms .....	1
Figure 2: Link to access the Blanket Credits form in the current system .....	2
Figure 3: Blanket Credit Registration form .....	3
Figure 4: Form filled by a graduate student to request for registration .....	3
Figure 5: Instructions to the faculty .....	4
Figure 6: Instructions to the staff .....	5
Figure 7: Email sent by the staff to the student .....	5
Figure 8: Instructions to student, faculty and staff .....	6
Figure 9: Model View Controller .....	16
Figure 10: Template .....	20
Figure 11: Template .....	21
Figure 12: Link directs students to the Blanket Credits form in the new system .....	29
Figure 13: Form displayed to the students .....	30
Figure 14: Email sent to the Professor .....	33
Figure 15: Form to approve/disapprove .....	33
Figure 16: Email sent to the student .....	34
Figure 17: Login form for the Staff .....	35
Figure 18: To list the students under a particular professor .....	36
Figure 19: List of students .....	36

# CHAPTER 1

## Introduction

Students conducting research must register each term for blanket credits. The current system for requesting blanket credits is manual, labor-intensive and error-prone. In this report we describe an alternate web-based system that eliminates many of the problems associated with the current system. In this section, we provide an overview of the currently manual process of Blanket Credit Registration. We outline the problems with the current system and motivate the need for automating the process.

### 1.1 Existing System

Every student pursuing an undergraduate, graduate or a PhD degree in Computer Science or Electrical Engineering can register for a certain number of credits every term. These are termed Blanket Credits. Blanket Credits enable them to pursue their research/ project without having the need of taking courses. These credits can be registered under any of the three categories – *Research, Reading and Conference* and *Project*.

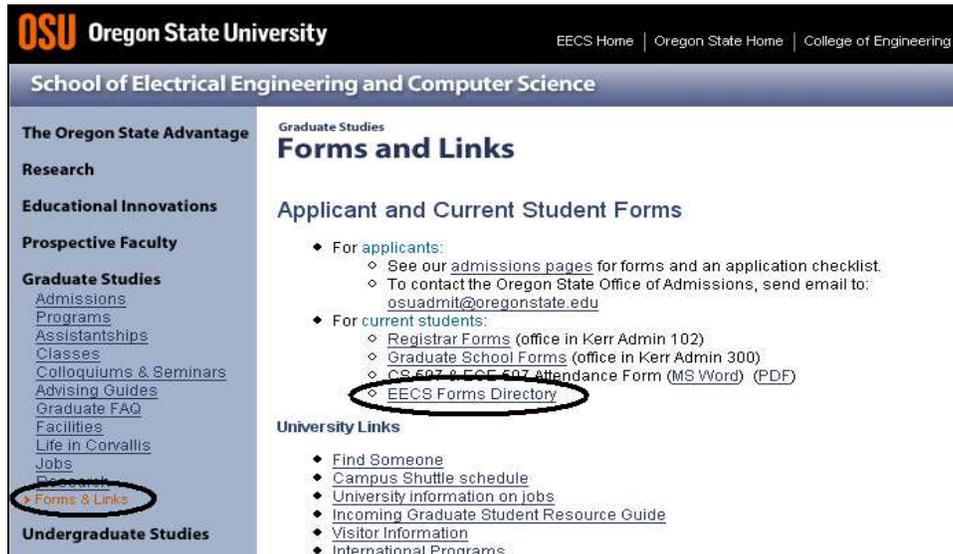


Figure 1: Link to access the forms

To register for blanket credits, the students use the process of Blanket credit registration. Currently, to support this process the students are provided with a set of instructions on a web page. The students are provided with a link on the university website, called EECS Forms Directory, as shown in Figure 1. Clicking on this link the students are directed to the commonly used forms as shown in Figure 2.

<b>Index of /graduate/forms</b>				
<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>	
 <a href="#">EECS507 attendance.doc</a>	09-Jan-2006 08:57	30k		
 <a href="#">EECS507 attendance.pdf</a>	09-Jan-2006 09:17	66k		
 <a href="#">blanket.txt</a>	17-Sep-2004 10:49	1k		
 <a href="#">crsequiv.html</a>	20-Aug-2004 14:49	13k		
 <a href="#">crsequiv current stu..&gt;</a>	19-Sep-2003 14:40	8k		
 <a href="#">domestic.pdf</a>	20-Feb-2003 19:04	67k		
 <a href="#">event.schedule.xls</a>	02-Mar-2005 10:05	23k		
 <a href="#">exit checklist eeecs.doc</a>	16-Sep-2004 09:19	37k		
 <a href="#">phd qual eval.doc</a>	07-Jun-2004 10:59	35k		
 <a href="#">thesis.sty.zip</a>	04-Feb-2004 13:27	25k		

**Figure 2: Link to access the Blanket Credits form in the current system**

The student should select the *blanket.txt* link from this directory. The student is then presented with the blanket credits registration form which is presented in Figure 3. The student has to then copy and paste the form on to the email editor. This form is then filled out by the students with the required information to acquire approval from the respective professor to register for the required number of credits.

The form lists all the course numbers to which a student can register depending on whether they are Undergraduate, Graduate or PhD students such as CS or ECE 401 for undergraduate, CS or ECE 501 for MS and CS or ECE 601 for PhD students. The students follow a set of instructions provided on the form to complete this approval process. The students are required to copy and paste the entire text information on to their email editor and fill out their information for term, Student Name, Student ID, Student

Email, CS or ECE, Course Number and the number of credits fields provided. The form filled out by a Graduate Computer Science student is presented in Figure 4.

```

GETTING APPROVAL TO REGISTER FOR CS/ECE BLANKET CREDITS
Blanket Course Numbers:
Undergraduate:
  Research (CS or ECE 401)
  Reading and Conference, (CS or ECE 405)
  Projects (CS or ECE 406)
M.S.:
  Research (CS or ECE 501)
  Reading and Conference (CS or ECE 505)
  Projects (CS or ECE 506)
Ph.D.:
  Research (CS or ECE 601)
  Reading and Conference (CS or ECE 605)
  Projects (ECE 606)
=====
Procedure: Follow these steps in order to register for ECE/CS blanket credits
or CS thesis credits:
The EECS staff needs email approval from the faculty member you are going to be
taking the blanket or thesis credit(s) with before the registration restriction
will be lifted and you can register for the credits.
STUDENT:
Step 1: Copy and paste this entire email into your email editor.
Step 2: Fill in the information below and email to faculty for their approval.
Term:
Student Name:
Student ID#:
Student email:
CS or ECE:
Course number:
Number of credits:
FACULTY
Step 3: To approve, forward student's email to Paul Bollmann at
paul.bollmann@oregonstate.edu and include your approval.
STAFF
Step 4: Paul receives faculty's approval and lifts restriction.
Step 5: Paul sends email to student - "restriction is lifted and you can
NOW register for (list course #)_____."

```

Figure 3: Blanket Credit Registration form

https://webmail.oregonstate.edu - Mail :: Message Composition - Mozilla Firefox

Message Composition - Sun Jun 18 08:53:18 2006

Send Message Save Draft Cancel Message

From somashsh@eeecs.oregonstate.edu

To professor@eeecs.oregonstate.edu

Cc

Bcc

Subject Registration

Options

Address Book Expand Names Spell Check Special Characters Attachments

Save a copy in "sent-mail"

Step 2: Fill in the information below and email to faculty for their approval.

Term: Summer 2006

Student Name: Example

Student ID#: 948563458

Student email: example@eeecs.oregonstate.edu

CS or ECE: CS

Course number: 506

Number of credits: 4

FACULTY

Step 3: To approve, forward student's email to Tyler Deadder at tyler.deadder@oregonstate.edu and include your approval.

STAFF

Step 4: Tyler receives faculty's approval and lifts restriction.

Step 5: Tyler sends email to student - "restriction is lifted and you can NOW register for (list course #)\_\_\_\_\_."

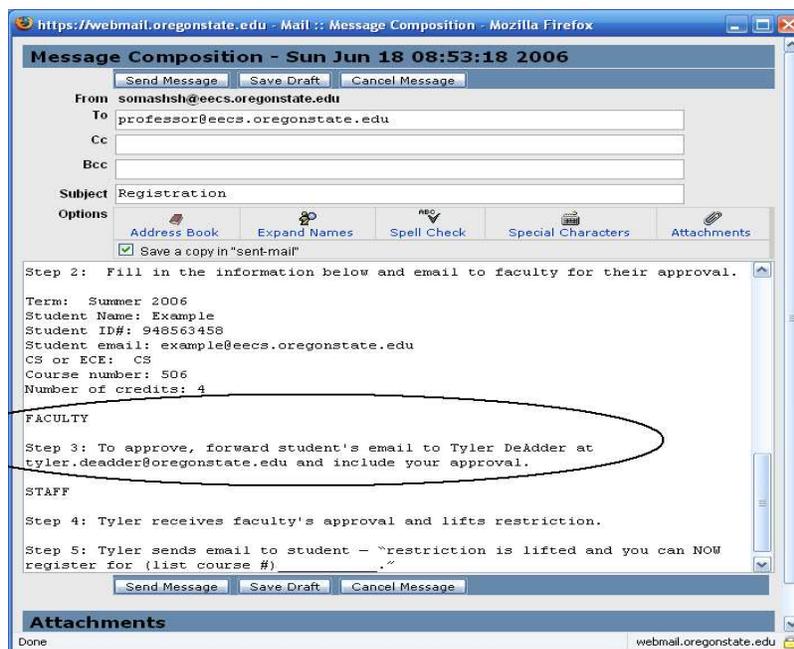
Send Message Save Draft Cancel Message

Attachments

Done webmail.oregonstate.edu

Figure 4: Form filled by a graduate student to request for registration

They are required to email this information to the professor from whom they are requesting approval. Once the student has emailed the request to the respective professor the student needs to wait until a decision is made prior to registering for the credits. This form also lists a set of instructions to the students, faculty and the staff that needs to be carried out to complete this approval process. Once the email is forwarded to the respective professor, the professor needs to go through all the information provided in the email. He then needs to follow the instructions provided for the faculty members, embedded among the same set of information as shown in Figure 5.



**Figure 5: Instructions to the faculty**

According to the instruction, on approval, the faculty member is required to forward the student's email along with their approval to the staff as can be seen from Figure 6. The staff, on receiving the email from the professor with their approval allows a particular student to register for the required number of credits. The staff has to remove the restriction for that student to allow him/her to register for the requested number of credits for that respective course. He then emails that student about the approval informing that the student can now register for those credits as shown in Figure 7.

Though the existing system provides a way to register for the credits, this process is carried out in an informal and a complicated manner. Allowing such an informal manner at the university level involves inconvenience to the students, faculty and the staff. The department would greatly benefit from an automated blanket registration system which is the goal of this project.

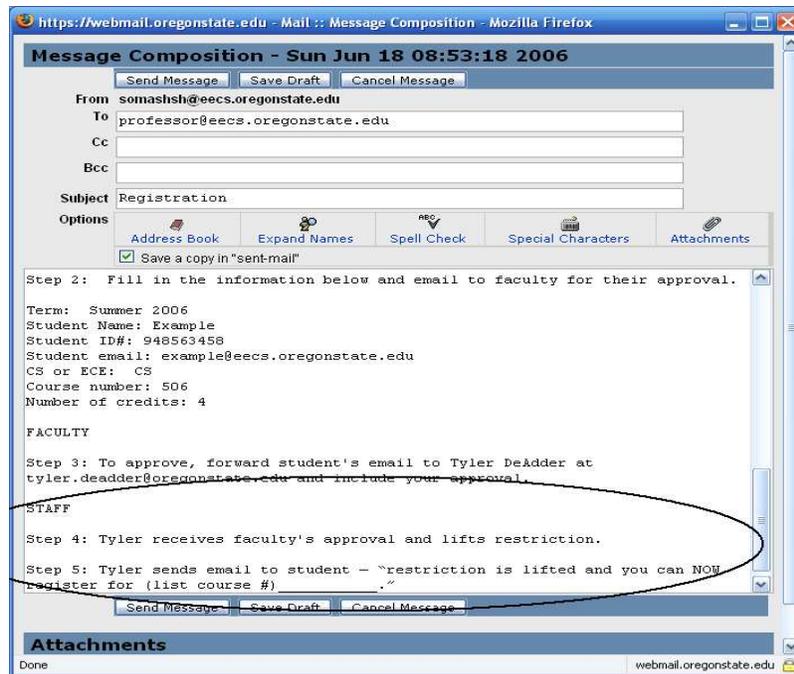


Figure 6: Instructions to the staff

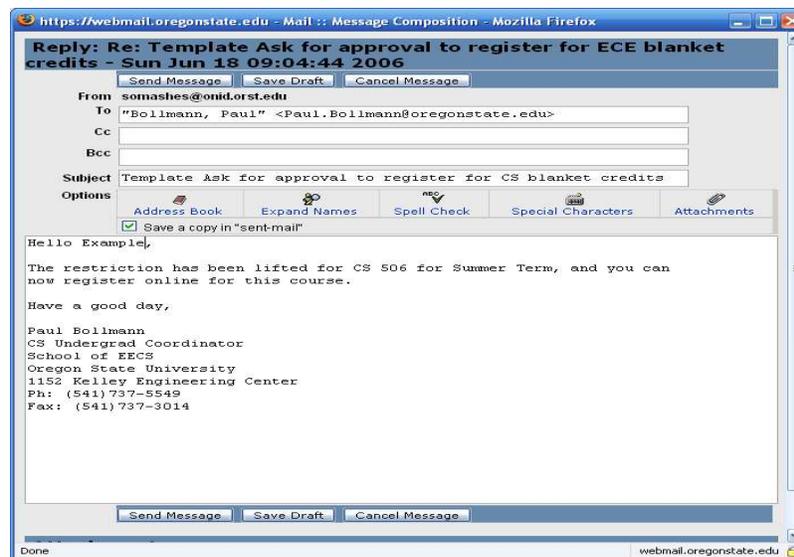


Figure 7: Email sent by the staff to the student

## 1.2 Issues in the Existing System

All the instructions to the students, faculty and the staff, as shown in Figure 8, are provided on the same form that is forwarded among the student, faculty and the staff members. This leads to cluttering of information on the form which is used only by the students to provide their information to the professors. There is no necessity for the student to understand the lifecycle of this process. Under the existing system, the student has to understand all the complicated procedures followed in this process.

Procedure: Follow these steps in order to register for ECE/CS blanket credits or CS thesis credits:

The EECS staff needs email approval from the faculty member you are going to be taking the blanket or thesis credit(s) with before the registration restriction will be lifted and you can register for the credits.

```

STUDENT:

Step 1: Copy and paste this entire email into your email editor.

Step 2: Fill in the information below and email to faculty for their approval.

Term:
Student Name:
Student ID#:
Student email:
CS or ECE:
Course number:
Number of credits:

FACULTY

Step 3: To approve, forward student's email to Paul Bollmann at
paul.bollmann@oregonstate.edu and include your approval.

STAFF

Step 4: Paul receives faculty's approval and lifts restriction.

Step 5: Paul sends email to student - "restriction is lifted and you can
NOW register for (list course #) _____."

```

**Figure 8: Instructions to student, faculty and staff**

Instead, the information pertaining only to the students can be provided in a more understandable form to ease the process. The students are not provided with the names of the professors under whom they could get approval for registration. As a result of which

the students need to find out by themselves about the professor who will be willing to allow them to register for the credits under them. This is done by physically going around and requesting them or by sending out emails to the professors requesting them to provide them with the permission for registration. Once they have received the permission from the professor for registering for the credits, they need to follow the instruction provided on the form and send an email to the professor for a formal registration of the credits.

On approval or disapproval of a student the faculty members are provided with instructions to forward the email sent by the student to the staff. The staff is responsible for lifting the restriction to allow the students to register for the course. But the email sent by the student consists of some unnecessary information as some students tend to be verbose. The faculty members need to go through this information to get to know the student who is requesting for the approval. Also, it only provides them with the information as to what they need to do if they approve for the registration but does not provide them with any information about the procedure to follow if they disapprove the request. The faculty members may not be able to provide approval to the students due to the amount of work load varying each term. Under such circumstances they would want to provide the students with the comments about their disapproval. They do not know the level of detail to which they need to respond to the student.

The students are not directed to provide a specific header to the email. So if any student provides a general header to the email and emails it to the professor, this email may go unnoticed by the professor who receives numerous emails every day. This causes a delay in the approval process. On the other hand if the emails are sent without any header they may also be discarded as spam mails by the spam filters that the professors use. The student needs to wait for an email either from the staff or the professor on the decision without being aware of the time that is required for the entire process to be completed. Since, there are deadlines for registering for the courses such delays causes problems to the students.

When the faculty member provides approval for a student and forwards their email to the staff, the staff is required to keep track of the students' names and the professor under whom they have been provided approval. He then sends an email to the respective student informing him/her about the approval from the professor. In the email students are also informed that the restriction has been lifted and they will now be allowed to register for the requested credits online. There can be number of students registering for blanket credits every term, considering undergraduate, graduate and PhD students. Manually keeping track of all the student names and the professor who has approved their registration is a tedious process. Also, each student can be approaching different professors at different times. For every student, the professor has to send an email to the staff approving the student's registration. It is clear that, manually tracking the Professor-Student approval is a time-consuming and an interruptive process. Also every time when an email is sent to the staff they have to process it as early as possible as there are deadlines set by which the students can register for the courses. At times when the staff member is not able to process the emails due to their work load or due to their absence, this job needs to be allocated to another person who will be present during this period to keep track of the names of the students and the professors. All these issues rise in the process of manually recording the information.

There is another set of manual task involved at the end of the term when it is necessary to email the professors reminding them to grade the students. In the existing system the staff looks over the list maintained in the form of an Excel sheet to go through the professor's names and the students registered under them. The staff member then provides the professors with the list of student's names they need to provide a grade for the blanket credits they have registered under them. With the manual tracking of these records there could be problems due to missing names and thus causing problems in the grading system. It could turn out that the student will know about this only at the end of the term when the grades come out. At that point it is necessary for the staff to go through the names thoroughly to make sure that the student was provided approval and then again inform the professor to allot a grade.

With all the above problems being encountered in the existing system, a new system is required that tackles all the above issues, making the task of Blanket Credit Registration for the students, faculty and the staff members convenient. The proposed system aims at considering all the above issues recognized and tackle them with the intention to ease the task of registration for the blanket credits.

### **1.3 Organization of the report**

The next chapter provides the requirements for the project and introduces the roles and responsibilities of the users of the system. The third chapter presents the architecture of the system and the tools used in the implementation of the project. Chapter 4 discusses the implementation details of this project. Chapter 5 presents the results of the evaluations that were conducted using a sample of three users from each of the user category of Student, Professor and Staff. Chapter 6 justifies the use of PHP as our scripting language against the several alternatives that we had considered. The final chapter concludes the report by outlining some possibilities for future enhancements. The Appendix section presents the forms that were used for evaluation.

## CHAPTER 2

### Project Requirements

The process for registering for Blanket Credits involves three different users: namely, the Students, Faculty and the Staff. The new system designed to facilitate the automation of the process should aim at simplifying the activities carried out by the users. The goal of the system is to provide the students with a more interactive version of the process, for example filling out a form online as opposed to the current process of copying and pasting information on to the email. The new system should also eliminate redundant efforts performed by the faculty in the approval/disapproval process. The system should also cater to the interests of the staff and provide a better method for recording and maintaining the information. This can be done by automatically storing the information as and when the students are approved.

#### 2.1 User Roles and Responsibilities

All the three users are identified based on the people interacting with the system to register for blanket credits. There are interactions among all the three users to complete this process of registration. The aim of the project is to streamline the actions carried out by each of these users by providing necessary information at each step of the process. Also, the system should help the users to proceed with and complete the process with less effort when compared to the existing system.

##### 2.1.1 Student

The students should be presented with a form to fill out to register for the credits instead of copying and pasting the form available online in to an email message. This could prevent the students from missing any information while copying the form. Three different categories of students: namely, the *Undergraduate*, *Graduate* and *PhD* students, can register for blanket credits under different options such as *Research*, *Project* and

*Reading and Conference.* The forms designed should clearly distinguish the courses to which these categories of students can register. The form should display only the information on the form that is in the interest of the students. This will avoid any ambiguities in the process of registration. It should provide the students with an interactive form to simplify their process of finding information about the professor they should email to. This can be achieved by listing the names of the professors who will be accepting students to register for blanket credits. All the known information such as the term, course numbers and the professors' names can be presented in the form via a drop down list for the students to select from. Emails should be automatically sent to the respective professor on submission of the forms by the students. This will avoid the task of finding the respective professor's email address to forward the email to. The students should also be provided with immediate notification about the subsequent step in the process. In this way they are informed about the current process.

### **2.1.2 Faculty**

Emails sent to faculty for the registration should have suitable header information to assist the faculty to recognize the email and respond to it. The message content should be simple and clear. They should be provided with only the relevant information. There should be precise information provided as to what needs to be done next to approve/disapprove the registration. Hyperlinks can be provided in each of the automatically generated emails, clicking on which will direct them to a form to make a decision. The process of conveying their decision to the students or staff should be automated. Manually sending emails is time-consuming. On the other hand if emails are generated automatically once they have made their decision, then the task is simplified. This can be a much easier process to make their decision instead of the present process of replying to each of the emails and forwarding the emails to both the students and the staff.

### **2.1.3 Staff**

The staff has to perform two set of tasks to complete this process. First, on approval for registration, they need to maintain the information of the students and the professors. Second, at the end of the term, they need to email the professors with the list of students they need to grade. The new system should aim at automating this process. It will be useful if the system stores the details of the student and the professor upon the approval of the registration. It will also be convenient if the system provides an easy way to notify the professors with the student names they need to grade.

## **2.2 Functional Requirements**

The requirements are categorized on the functionalities carried out by the three groups of user. Though these three groups of users perform different actions, there are interactions among the users to complete the entire process successfully. Each group of user communicates with the other group of users in the process. Thus, the first step of requirements gathering is to understand the functionality of each of these users. Next there should be a clear understanding of the interaction among all the users with each of the other group of users carrying out varied functionality.

### **2.2.1 Requirements pertaining to the Students**

Since the system involves registering for courses, students play an important role in the working of the system. They are the users who initiate the working of the system. Hence, performing a survey and gathering information about the existing system is the first step in the requirements gathering process. In talking to students who use this process on a regular basis and me being a student undergoing this process every term gives a better perspective about the process involved. Since the students have to undergo this process every term, there is a need for a change in the existing system. The existing system to register for Blanket credits is a complicated process which is not suitable for repeated

execution. The students pointed out that the procedure of copying and pasting the form available online on to the email as an undesirable job. Since the form is already available online, they prefer an automated system which will make the process easier and more accurate. They also find it difficult to determine the professors to be approached for the approval. With processing deadlines to register for the courses, they will find it easier if there can be a way to pace the process. It will also be convenient if there is a way by which they can be ensured that all the information is reaching the right people on time.

### **2.2.2 Requirements pertaining to the Faculty**

The Faculty are at the intermediate level of the system. Based on the decision made by the faculty, it is possible to take any further steps in the registration process. The faculty feels that they might not read the emails due to improper subject information under the current process. They might even sometimes discard the emails as spam messages when there is no subject provided. Hence, they feel the necessity of a system that notifies the arrival of request for registration with appropriate header information. They also find it very inconvenient to answer to emails due to the heavy volume of emails that they receive everyday.

### **2.2.3 Requirements pertaining to the Staff**

They form the core user for the functioning of the system. This is because they have to deal with the information gathered from this system for an entire term. Any incorrect record of information could lead to undesirable situations. Discussing with the staff about this process of Blanket Credit Registration provided an insight of the actual working of the existing system. They find it a very tedious process. Lot of effort and time is consumed in the process of categorizing the student information as and when they receive emails from the professors. They have to maintain an excel spreadsheet to record the details of the professor and the students registered under them. Emails from different professors arrive at different times. Categorizing them is a time consuming and a lengthy process. It is also necessary to go through the information in every email very carefully to

avoid occurrence of any mistakes. Later, at the end of the term, the staff emails the professors with the student names that they need to assist a grade to which is again quite a wearisome process. Also, the staff has to maintain a check list at the end of the term to record as to which professor they have already notified and which professor they have not. They must make sure that all the professors are notified and that they do not send redundant emails.

### **2.3 Objective**

The goal of the new system of Blanket Credit Registration is to simplify the task of the students, faculty and the staff in this process. There are three primary objectives for the upgraded system. First, to provide the students with a better system to aid in the registration process and to provide a better way to the students in finding the required information with less effort and in a much accessible manner. Second, provide the faculty with an interactive system to fasten the process. Third, to provide the staff with a better maintainable system to keep track of the students registering for the courses. Also then the goal is to provide help in administering their grades towards the end of the term.

Providing the students with an interactive system should be made unambiguous. It could help the students in acquiring the required information without overlooking at them. Providing the students with information of the ongoing procedure as they proceed with the registration process is more informative. They should be updated with the next steps to take and make sure that they are aware of the next step in the process. Maintaining a database to keep track of the students and their relevant information is an efficient way to manage information. Maintaining the information when the students register could be used later for some other inferences.

## CHAPTER 3

### System Architecture

In this section, we present the architecture of our proposed system. First we present our design pattern for the system and the pros and cons of the design pattern. Then, we outline the detailed architecture and then provide an overview of the tools that were used in the project namely *Smarty* and *Propel*. *Smarty* is used to design the UI in our system while *Propel* is used for database operations. We also present examples to explain the use of these tools in our system.

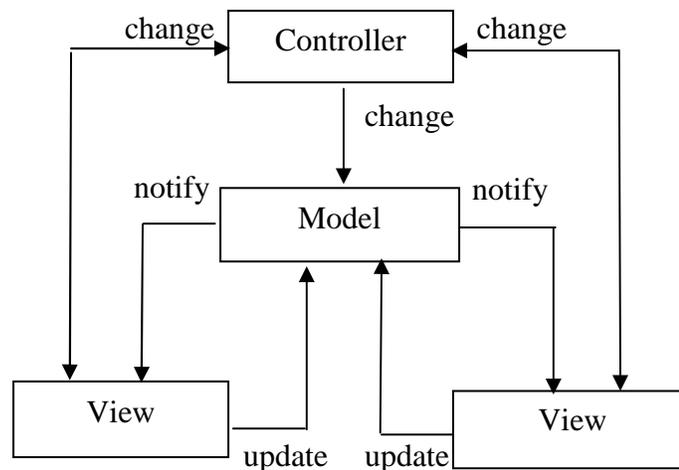
#### 3.1 Model View Controller (MVC) pattern

A design pattern describes a proven solution to a recurring design problem, placing particular emphasis on the context and forces surrounding the problem, and the consequences and impact of the solution [8]. The reasons to use design patterns are:

- **Proven:** The experience, knowledge and insights of developers who have used these patterns successfully in their own work can be tapped
- **Reusable:** When a problem recurs, a new solution need not be invented; instead the pattern can be followed and adapted as necessary
- **Expressive:** Design patterns provide a common vocabulary of solutions, which can be used to express larger solutions succinctly

##### 3.1.1 Overview

Model View Controller is a popular Object Oriented pattern and is presented in Figure 9. The purpose of the MVC design pattern is to separate the application object (model) from the way it is presented to the user (view) and from the way in which the user controls it (controller). Before MVC, user interface designs tended to couple objects together. MVC decouples them thus allowing greater flexibility and possibility for re-use. MVC also provides a powerful way to organize systems that support multiple presentations of the same information [8]. We now explain the important components of the MVC design pattern that is presented in Figure 9.



**Figure 9: Model View Controller**

The model object contains information about all the data that need to be displayed. It also knows about all the operations that can be applied to transform that object. However, it has no knowledge about the GUI, the manner in which the data are to be displayed, or the GUI actions that are used to manipulate the data. The data are accessed and manipulated through methods that are independent of the GUI. The model represents the enterprise data and the business rules that govern access to and updates of this data.

The view object refers to the model. It uses the query methods of the model to retrieve data from the model and then displays the information. It accesses enterprise data through the model and specifies how that data should be presented. The view object is also responsible for maintaining consistency in its presentation when the model changes.

The controller object knows about the physical means by which users manipulate data within the model, for example, clicking a mouse, typing on the keyboard etc. A controller translates interactions with the view into actions to be performed by the model. In a GUI for example, the controller object would receive mouse clicks or keyboard input which it would translate into the manipulator method that the model understands. In GUIs, views and controllers often work very closely together. For example, a controller is responsible for updating a particular parameter in the model which is then displayed by a view. In

some cases a single object may function as both a controller and a view. Each controller-view pair is associated with only one model. However a particular model can have many view-controller pairs.

### 3.1.2 Advantages

The MVC architecture has the following benefits:

**Multiple views:** The separation of model and view allows multiple views to use the same enterprise model.

**Easier support for new types of clients:** To support a new type of client, a view and controller can be written for it and wire them into the existing enterprise model.

**Clarity of design:** The model's method list makes it easy to comprehend how to control the model's behavior. When designing the application, this trait makes the application easier to implement and maintain.

**Efficient modularity:** The design allows any of the components to be swapped in and out as the user or programmer desires. Changes to one aspect of the program aren't coupled to other aspects. Also, development of the various components can progress in parallel, once the interface between the components is clearly defined.

**Ease of growth:** Controllers and views can grow as the model grows; and older versions of the views and controllers can still be used as long as a common interface is maintained.

**Substitutable user interface:** Different views and controllers can be substituted to provide alternate user interfaces for the same model.

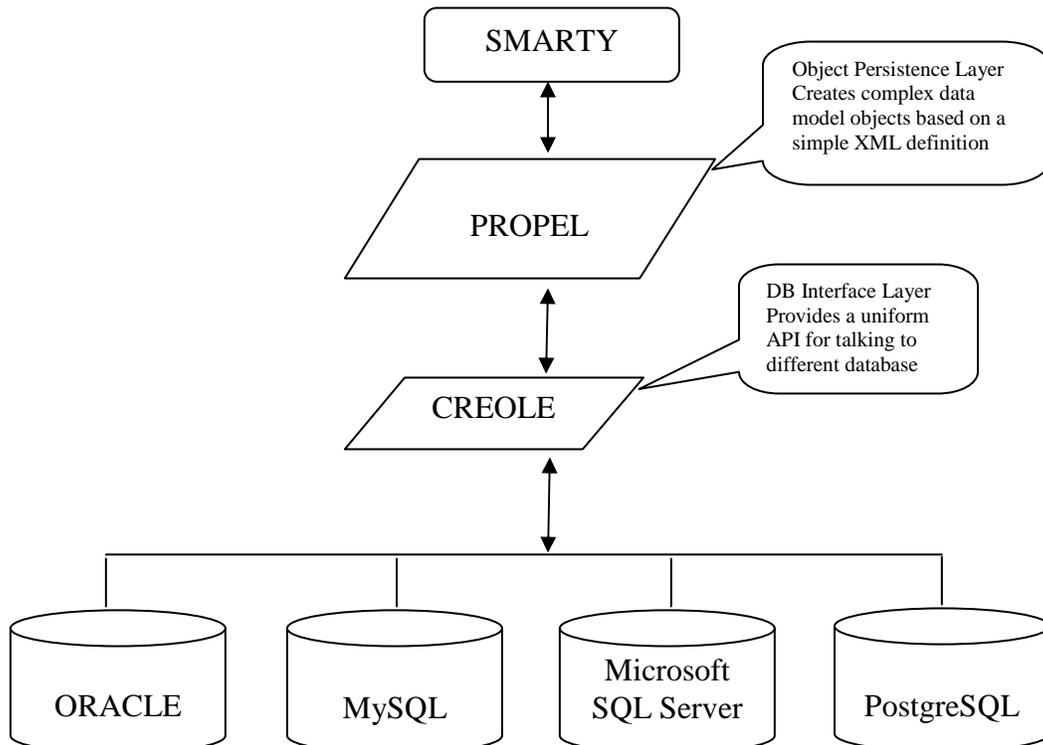
**Easier user interface changes:** Changes affecting just the user interface of the application logic become easier to make.

### 3.1.3 Drawbacks of MVC

- Close coupling of views and controllers to model. Changes to the model interface require parallel changes in the view and may require additional changes to the controller. Certain code changes become more difficult.
- Potential for excessive updates. The change propagation mechanism can be inefficient when frequent model changes require many change notifications. This is not prevalent if a passive model is used.
- Close coupling between view and controller. Strict separation is difficult if not impossible.

### 3.2 Detailed Architecture

To incorporate MVC pattern in PHP coding, a set of tools are used. Smarty describes the “Template/Presentation” framework for PHP programming. It automates the tasks dealt at the presentation layer of an application. Propel, a full-service object persistence and query toolkit for PHP forms the model layer of the architecture. MySQL is used to provide the database component for the application. Creole provides a uniform database API or data abstraction for PHP.



## 3.3 Tools

### 3.3.1 Smarty

#### 3.3.1.1 Overview

*Smarty* is a Template/Presentation Framework. It provides the programmer and the template designer with a set of tools to automate tasks commonly dealt with at the presentation layer of an application. It focuses on providing quick development and deployment of an application while maintaining high-performance, scalability, security and future growth [4].

One of the purposes of *Smarty* is to facilitate the separation of application code from the presentation. The application code consists of the business logic of the application written and maintained in PHP. The presentation is the way the content is presented to the end user and it is written and maintained in the template files. Smarty, at its most basic function as the application code collects content, assigns it to the template engine and displays it. The application code has no concern as to how this content will be presented in the template.

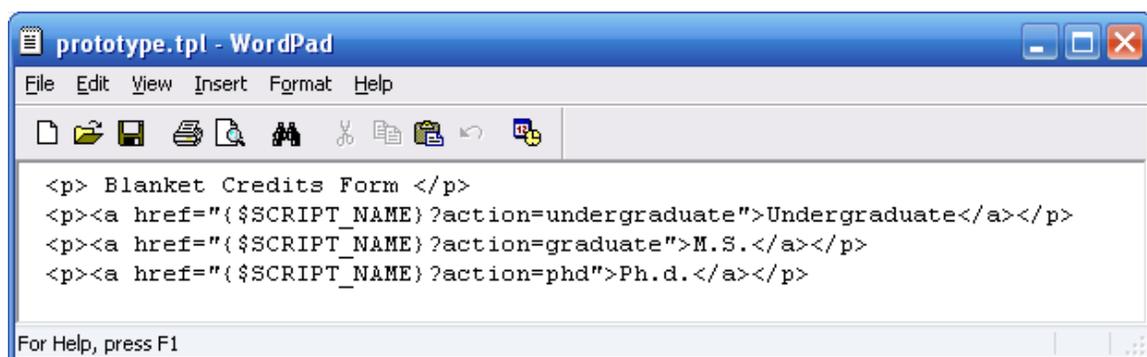
#### 3.3.1.2 Advantages

- Designers can't break application code. They can mess with the templates all they want, but the code stays intact. The code will be tighter, more secure and easier to maintain.
- Errors in the templates are confined to the Smarty's error handling routines, making them as simple and intuitive as possible for the designer.
- With presentation on its own layer, designers can modify or completely redesign it from scratch, all without intervention from the programmer.
- Programmers cannot mess with templates. They can go about maintaining the application code, changing the way content is acquired, making new business rules without disturbing the presentation layer.

- *Smarty* can assign variables extracted from configuration files. Template designers can maintain values common to several templates in one location without intervention from the programmer, and config variables can easily be shared between the programming and presentation portions of the application.
- Templates do not contain PHP code. Therefore, a template designer need not understand the entire functionality of PHP, but only the subset of functionality made available to them from the programmer (application code.)
- Web page designers are not dealing with PHP code syntax, but instead an easy-to-use templating syntax not much different than plain HTML. The templates are a very close representation of the final output, shortening the design cycle.
- The programmer has complete control of the template output and the compiled template content with pre-filters, post-filters and output-filters.
- *Smarty* compiles templates into PHP code behind the scenes, eliminating run-time parsing of templates.

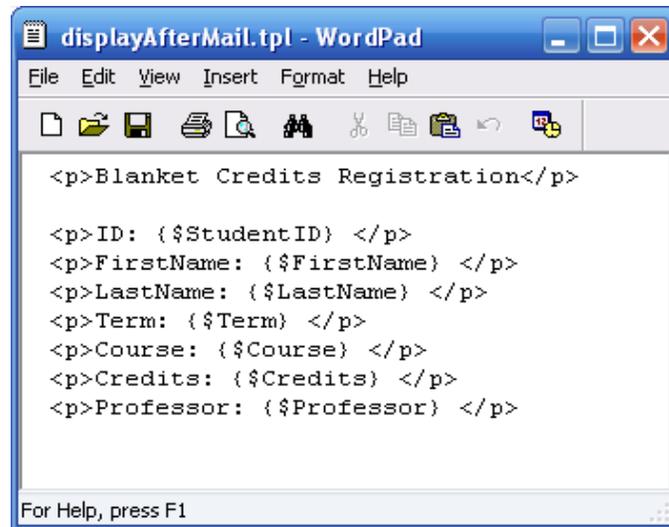
### 3.3.1.3 Role of Smarty in the project – An example

*Smarty* is used to separate the UI and the application logic thus keeping the UI simple and making modifications to the UI easier. Templates are used to display the information. Figure 10 and Figure 11 present two example interfaces in the project corresponding to the initial page displaying the different categories of the students and the page displaying the details of the student after an email has been sent to the professor. As can be observed from the figure, the interface does not manipulate the data but only serves to display the information to the user.



```
prototype.tpl - WordPad
File Edit View Insert Format Help
[Icons]
<p> Blanket Credits Form </p>
<p><a href="{$_SCRIPT_NAME} ?action=undergraduate">Undergraduate</a></p>
<p><a href="{$_SCRIPT_NAME} ?action=graduate">M.S.</a></p>
<p><a href="{$_SCRIPT_NAME} ?action=phd">Ph.d.</a></p>
For Help, press F1
```

Figure 10: Template



**Figure 11: Template**

As an example, consider the following code that assigns the data retrieved from the database to the template.

```
$this->tpl->assign("FirstName", $firstName);
$this->tpl->assign("LastName", $lastName);
$this->tpl->assign("StudentID", $studentid);
$this->tpl->assign("Term", $term);
$this->tpl->assign("Course", $course);
$this->tpl->assign("Credits", $credits);
$this->tpl->assign("Professor", $professor);
$this->tpl->display('../templates/displayAfterMail.tpl');
```

Once the data has been assigned to the template, then the values can be accessed in the template via these variables. But these variables are not manipulated by Smarty in the interface themselves.

## 3.3.2 Propel

### 3.3.2.1 Overview

Propel is the model layer of the traditional MVC. Propel provides database abstraction and object persistence services for PHP. It helps in eliminating the use of SQL statements and database connections. Classes and objects are used to access data in a convenient manner. Propel provides a system for storing objects to a database and for searching for and restoring the objects from the database. Propel enables the user to perform complex

queries and database manipulations without writing SQL queries. Propel makes applications easier to write, easier to deploy, and a lot easier to migrate if the need should arise. Propel provides a smart and comprehensive data management service at minimal performance cost to the PHP application [5].

Propel has two main components that work together to provide efficient object persistence customized for the RDBMS:

- A generator engine for building the classes and SQL files (propel-generator)
- A runtime environment that provides tools for building SQL queries, executing compiled queries, and tools for managing connections to multiple databases simultaneously (propel)

The Generator is based on a simple XML schema describing the database and tables, the generator will build PHP classes to interact with the data model. It also builds SQL definition files to create the tables, keys, sequences for the specific RDBMS. The use of a build system to create RDBMS-specific SQL results in a database-independent system without sacrificing performance.

The Runtime framework provides the ability to use these generated classes in the PHP scripts to transparently handle reading from and writing to the database. The runtime environment provides a layer of database abstraction and encapsulation for business logic rules. Propel's classes are designed to encapsulate any data level validation needed by the application.

### **3.3.2.2 Tools used**

Propel consists of the following tools to provide a complete abstraction layer to the application. Phing is a project build system based on Apache Ant [7]. It is a highly flexible build framework. Phing has the following features:

- Simple XML buildfiles
- Rich set of provided tasks
- Easily extendable via PHP classes

- Platform-independent: works on UNIX, Windows, MacOSX
- No required external dependencies
- Built & optimized for ZendEngine2/PHP5

Phing uses XML *buildfiles* that contain the set of instructions. The buildfile is structured into targets that contain the actual commands to execute a database query. So, to use Phing, a buildfile is written and then Phing is run, specifying the target in the buildfile that needs to be executed.

Creole is a database abstraction layer for PHP5. It abstracts PHP's native db-specific API to create more portable code while also providing developers with a clean fully Object-Oriented interface base [6].

Creole provides the following distinguishing features:

- Built for PHP5: New object model, exceptions
- Fully Object-Oriented API
- Provides classes and methods for scrolling: next, previous, first, last, relative, and absolute row fetching.
- Classes and methods provide an additional means for iterating over recordsets.
- Comprehensive database metadata (database, table, full column info, primary keys, foreign keys, indexes) using simple OO API.
- Type-specific methods handle any necessary conversions and escaping for inserting and retrieving values.
- Native (no emulation) handling of autoincrement / sequence id generation.

### 3.3.2.3 Features

**Easily extensible:** Use of XML schemas makes it possible to easily integrate with other components which are driven by the data model or to extend the schema definition to include attributes for other components.

**Easily customizable:** The generated object model is built to be customized. All internal calls are made to empty stub classes which extend the classes with generated logic. Hence it is possible to over ride the behavior in the stub classes or provide own custom business logic without sacrificing the ability to rebuild the object model at a later time.

**Truly independent:** The use of a build process to create customized SQL and PHP classes means that the database interactions with Propel are database-independent, taking advantage of the strengths of the RDBMS and providing emulation for things the RDBMS doesn't support. Tools are also provided to easily move data from one RDBMS to another.

**Intuitive, Predictable:** Propel's data model definition follows very closely to the structure of the underlying database. This provides the advantage of making Propel easy to understand and also of behaving predictably with very complex data models.

**Advanced:** Propel makes the difficult relational problems easy and efficient. For example, Propel provides the ability to populate related (foreign key) objects from a single query. Propel also supports basic inheritance and other advanced OO features.

### 3.3.2.4 Advantages

- Propel gives the database a well-defined API. Instead of creating complex SQL statements for finding and manipulating data, developers make calls to a simple API.
- Propel allows the user to include business logic for altering the underlying database.
- The database schema can be easily changed without affecting the code used to access it. Propel is very flexible when changing RDBMS environments. It can access several RDBMS systems and move data between them very easily, giving the ability to change the underlying database or modify the database at a later time.

- Easy to get started: Because Propel uses an XML schema to describe the data model, migrating from one RDBMS to another to another is as easy as changing a configuration property
- Easy to use: Using Propel in a PHP application is very simple. Once the database configuration is specified and the object model is built, using the generated classes to manipulate the data model is as simple as including the desired PHP class in the script and working with it like any other class. Persistence is handled transparently when the save() method of the object is invoked.

### 3.3.2.5 Role of Propel in the project – An example

Propel is used to define the schema in XML format. Once the XML schema is defined, the object model can be built using Propel. An example XML schema is presented below.

```
</table>  
<table name="faculty" description="Faculty Table">  
  <column  
    name="id"  
    required="true"  
    primaryKey="true"  
    autoIncrement="true"  
    type="INTEGER"  
    description="Faculty Id"/>  
  <column  
    name="name"  
    required="true"  
    type="VARCHAR"  
    size="128"  
    description="Faculty Name"/>  
  <column
```

```

name="email"
type="VARCHAR"
size="128"
description="E-Mail Address"/>
</table>

```

Each table description used in the system is specified in the XML file which is used to describe the tables and the relationship between the tables in the database.

A file called “build.properties” must be created that would contain the information about the project that propel needs in order to build the classes and SQL files, such as project name, database etc. An example “build.properties” file segment for the registration database is presented below.

```

propel.project      = registration
propel.database    = mysql
propel.database.url = mysql://root@localhost/registration

```

On building the object model for the system registration, the following directory structure is created automatically.

```

build – classes – registration – map – om - Faculty.php
      |      | FacultyPeer.php
      |      | Student.php
      |      | StdentPeer.php
      |      | - - - - - BaseFaculty.php
      |      |           BaseFacultyPeer.php
      |      |           BaseStudent.php
      |      |           BaseStudentPeer.php
      | - - - - - FacultyMapBuilder.php
                        StudentMapBuilder.php

```

*BaseFaculty.php*, *BaseFacultyPeer.php*, *BaseStudent.php*, *BaseStudentPeer.php* are used to add a student, faculty or find a student and faculty in the database. The PHP classes contain the definition to *Save*, *Delete* a record and other methods such as to *Access* data by *primaryKey* or *indexNumber* which can be used later for data access. It would be worth recalling that these methods are generated automatically as well. Propel also generates *Faculty.php*, *FacultyPeer.php*, *Student.php*, *StudentPeer.php* which are empty subclasses that can be used to customize the behavior in the project without making changes to the generated code. These classes allow the simplified code in the controller to access data from the database.

The following segment of code present an example on the creation of a student object after obtaining the information from the forms.

```
$a = new Student();
    $a->setFirstName($sfn);
    $a->setLastName($sln);
    $a->setStudentid($si);
    $a->setTerm($st);
    $a->setCourse($cn);
    $a->setCredits($noc);
    $a->setEmail($to);
    $a->setProfessor($profname);
    $a->setFacultyid($id);
$e = $a->save();
```

The properties of the object is set and the object is saved in the database with a call to the method *save()*. It can be observed that the database operations are simplified when compared to the default method of writing PHP scripts for database operations (presented below).

```
$mysql_handle = mysql_connect($dbhost, $dbuser, $dbpass)
    or die("Error connecting to database server");
mysql_select_db($dbname, $mysql_handle)
```

```
or die("Error selecting database: $dbname");  
$query = "INSERT INTO records VALUES ('$prof', '$sn', '$si', '$cn', '$noc', '$to_student')";  
mysql_query($query);  
mysql_close($mysql_handle);
```

Without the MVC pattern the PHP coding to carry out the same functionality involves writing SQL statements. It is convenient to program when there is less format specific statements to be generated.

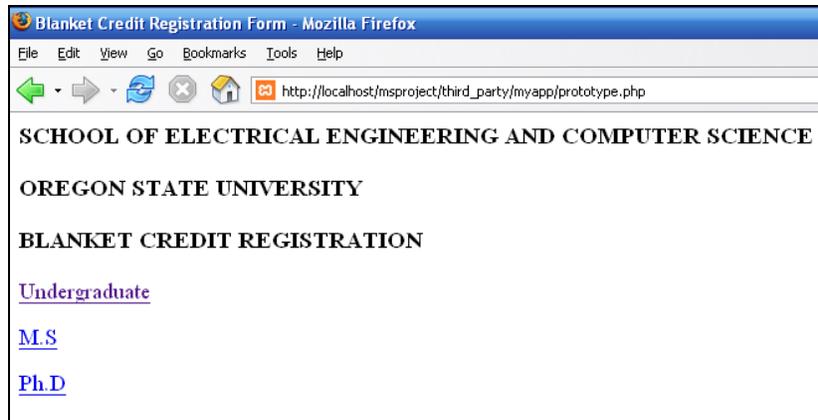
## CHAPTER 4

### Implementation

The Blanket Credit Registration System is designed to overcome the difficulties observed in the current system. It is developed using PHP to ease the process of blanket credit registration. The system is intended to cater to the needs of all the three category of users namely the students, staff and the faculty. Web forms are designed in the new system to facilitate this process. In this section, we provide the details of our implementation by presenting the modules corresponding to the three users: *Students*, *Staff* and *Faculty*.

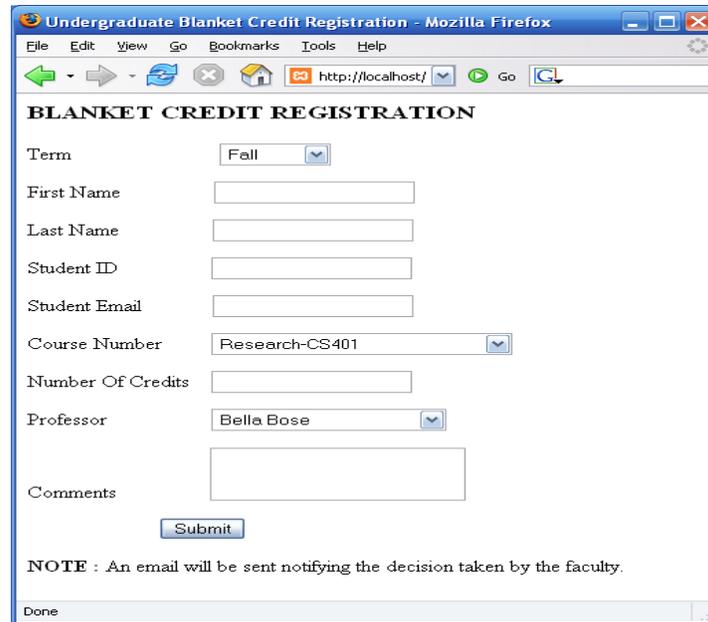
#### 4.1 Students

With the new students they need not meet the professors personally to register. The students are provided with a link that is easily noticeable on the university web page. Clicking on this link directs the students to a form presented in Figure 12.



**Figure 12: Link directs students to the Blanket Credits form in the new system**

This form presents the students with three different links for *Graduate*, *undergraduate* and *PhD* students. This distinction is made to direct the different program students to their respective form. Thus effectively, we are providing them only with relevant information. Clicking on the link directs the students to a form shown in Figure 13.



The screenshot shows a web browser window titled "Undergraduate Blanket Credit Registration - Mozilla Firefox". The browser's address bar shows "http://localhost/". The main content area displays a form titled "BLANKET CREDIT REGISTRATION". The form contains the following fields:

- Term: A dropdown menu with "Fall" selected.
- First Name: A text input field.
- Last Name: A text input field.
- Student ID: A text input field.
- Student Email: A text input field.
- Course Number: A dropdown menu with "Research-CS401" selected.
- Number Of Credits: A text input field.
- Professor: A dropdown menu with "Bella Bose" selected.
- Comments: A text area.

At the bottom of the form is a "Submit" button. Below the form, a note reads: "NOTE : An email will be sent notifying the decision taken by the faculty." The browser's status bar at the bottom shows "Done".

**Figure 13: Form displayed to the students**

To proceed with the process of blanket credit registration, it is required that the students fill out this form. The fields on the form are as follows:

*Term:* Select the term they want to register in.

*Student Name:* Name of the student registering for the blanket credits

*Student ID:* OSU ID, required to authenticate the students registering for the credits.

*Student Email:* Email address of the students to notify them about the approval/disapproval decision taken by the faculty.

*Course Number:* Drop down list of courses that the students can register for blanket credits. These courses are listed based on the program (Undergraduate, Graduate or PhD) the students are pursuing.

Undergraduate students accessing this form have the following course options:

Research (CS or ECE 401)

Reading and Conference (CS or ECE 405)

Projects (CS or ECE 406)

While Graduate students accessing this form have the following course options:

Research (CS or ECE 501)

Reading and Conference (CS or ECE 505)

Projects (CS or ECE 506)

And PhD students accessing this form have the following course options:

Research (CS or ECE 601)

Reading and Conference (CS or ECE 605)

Projects (CS or ECE 606)

Thus, listing only the courses pertaining to the program the student is pursuing avoids ambiguities in the process.

*Number of Credits:* The number of credits they desire to obtain approval for registration.

*Professor:* Drop down list of names of Professors who are willing to accept students to register for the blanket credits.

*Comments:* Additional information that the students would want to provide to the professors.

.

The information gathered on submission of the form by the students is stored in the database to allow the transfer of information along the process to different sets of users. A note is provided at the end of the form to inform the students about the next step in the process. The note gives them the information that they will receive an email informing them about the decision taken by the professor.

This is the process undergone by the students to register for the blanket credits. Unlike the existing system, the students are provided access to this form very easily without undergoing a lot of search for the link. They are also not required to perform the

cumbersome process of copying and pasting the form in an email. Instead they are only required to fill out a form and submit it. They are relieved of the work of personally meeting Professors who are willing to accept students. The new system provides the students with a list of Professor's names. This is implemented by maintaining a database of the Professor's names. The dropdown list provided on the form is populated with the Professor's names. This gives information to the students about the professors they can approach. The students need not look for the respective Professor's email address from the department webpage to send them an email. Instead, in the new system, on submitting the form an email is sent automatically to the professor they have selected from the drop down list.

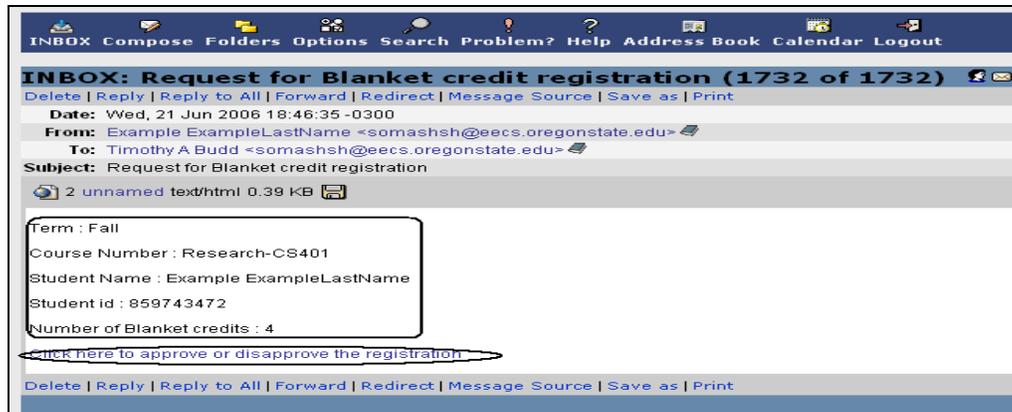
By filling out the form and submitting it, the students are guaranteed to provide the required information. Keeping the form simple and precise reduces the chances of them missing out on details the absence of which could potentially delay the process. This can happen in the existing system as it involves copying and pasting the form into the email editor. Information to be provided can be overlooked due to cluttering of sentences.

The new system provides messages that assure the students of them receiving an email about the professor's decision. This provides an answer to their question about the next step in the process. Hence, the new system brings down the task of the students to just filling out a form when compared to the elaborate process involved in the existing system.

## **4.2 Faculty**

On submission of the form by the students, an email is sent to the respective professor automatically. This email is generated on the submission of the form as a part of the process. Hence, the email header and content are chosen appropriately. The header of the email generated is "*Request for Blanket Credit Registration*". This makes it easier for the faculty members to recognize it as an email sent to them by the students requesting for the permission to register for blanket credits. This would make sure that the email will be answered by the professors and not discarded as spam emails. The content of the email is

also kept simple pertaining to the task. The email is not burdened with any kind of unnecessary information.



**Figure 14: Email sent to the Professor**

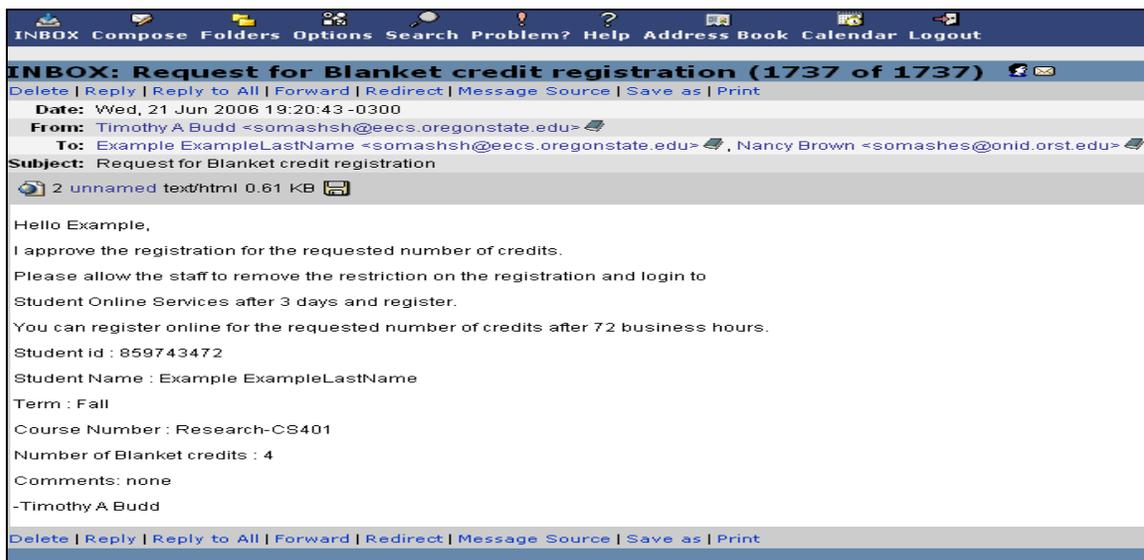
To keep the content short and precise, information about the term, course number and the number of credits they are registering for and the name and id of the student is provided. Along with this information, the faculty members are also provided with a link in the email shown in Figure 14. They need to access this link to make their decision. Clicking on the link directs the professors to the form presented in Figure 15.

The screenshot shows a Mozilla Firefox browser window displaying a web form. The page title is "ELECTRICAL ENGINEERING AND COMPUTER SCIENCE OREGON STATE UNIVERSITY BLANKET CREDIT REGISTRATION". The form contains the following fields and options:
 

- Student Information:**
  - ID: 859743472
  - FirstName: Example
  - LastName: ExampleLastName
  - Term: Fall
  - Course: Research-CS401
  - Credits: 4
  - Professor: Timothy A Budd
- Comments:** [Text input field]
- Approve  Disapprove
-

**Figure 15: Form to approve/disapprove**

In the form, all the necessary information about the students such as name, id and email are provided. The faculty members can go through these details before they make their decision. To make a decision, the faculty member needs to just check approve or disapprove checkbox. If any of the faculty members desire to provide any comments about their decision they can do so by writing them in the space provided in the comments text area. On making their decision the faculty members need to submit the form. This completes the task of the faculty members in the process of blanket credit registration. They need not take care of emailing the student or the staff member about their decision. This task is performed automatically by the new system. Based on their decision, the concerned people are notified about their decision. If the professor approves a particular student, they check the checkbox “Approve”. When the professor submits the form after this action the student and the staff are sent email about their decision as shown in Figure 16. The content of the email provides information to the student about the approval of the professor. It also informs that the staff requires approximately three days to lift the restriction on the registration after approval and that they could login to Student Online Services after three days to register for the requested number of credits.



**Figure 16: Email sent to the student**

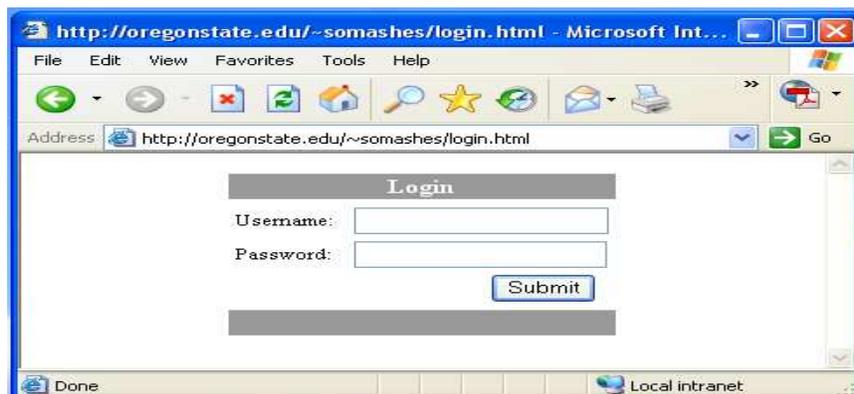
Apart from sending an email to the student, the staff is also sent an email immediately when the professor approves the registration. This is done to inform the staff to lift the

restriction on that particular student within 72 hours. In the case where the professor disapproves the registration the “Disapprove” checkbox is checked. On submitting the form, only the student is sent an email about their decision.

### 4.3 Staff

The staff is provided with the information of the student who is registering for the course. On receiving the email, the staff removes the restriction for that particular student. All the necessary information such as *Student name, id, course number, the number of credits*, required by the staff to lift the restriction on a course is provided. Unlike the old system, the staff need not maintain an excel sheet to keep track of the student’s information. This is taken care by the new system by maintaining a database of the names of the students registering for the credits, number of credits and the Professors under whom they are registering. This avoids the additional work performed by the staff to maintain the records of the students along with the task of lifting the restriction.

In the current system, the staff emails the professors at the end of the term reminding them to provide the grades to the students registered under them. This task is simplified in the new system by maintaining a database of the student records. The staff is provided with an authenticated login to access this database. Access to the database is provided via a form presented in Figure 17.



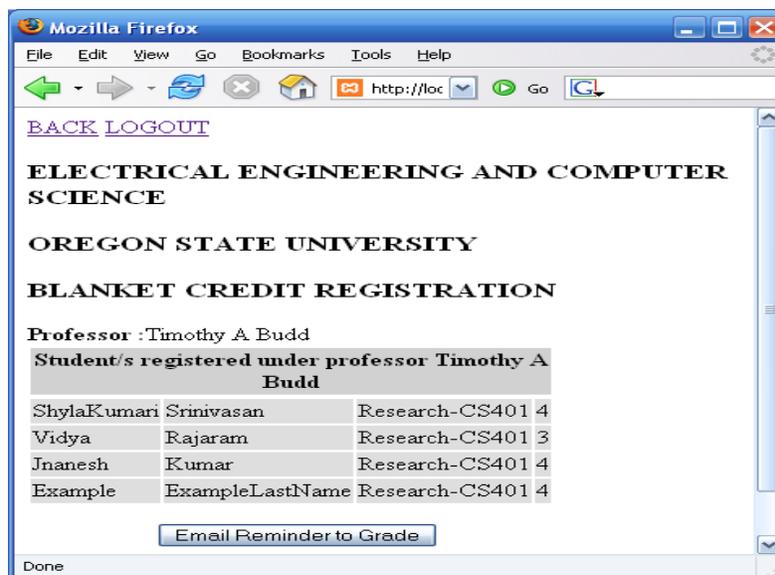
**Figure 17: Login form for the Staff**

On successful authentication the staff is directed to a form to access the list of the names of the professors as shown in Figure 18.



**Figure 18: To list the students under a particular professor**

The staff will then be able to select the names of the professors from the dropdown list. This would list the names of the students registered under that particular professor. An *Email Reminder to Grade* button is provided at the end of the list, as shown in Figure 19. Clicking on this button the professor is automatically sent an email reminding them to provide the grades to the students on the list.



**Figure 19: List of students**

In this way, the staff can send individual emails to the professors by selecting the required professor's name from the dropdown list. This change in the system of organizing the task of blanket credit registration by keeping record of the acquired necessary information eases the task of the staff in maintaining long records of students.

## **CHAPTER 5**

### **Evaluation**

In this section, we outline the experimental setup and present the results of the tests that were performed on our system using a sample of users. We evaluated our system on some students, faculty and staff in the school of EECS.

#### **5.1 User Evaluation**

Users prefer a system that is easy to learn and to use and that is effective, efficient and satisfactory. Though evaluation can be carried out at any step of development, in our system, it is carried out before deployment to ensure that all the requirements are met and that the functionalities required are carried out by the system. The evaluation is carried out on a sample of users from all the three groups of users involved in the system. The goal of the evaluation is to assess how well the design fulfills users' needs and how the users rate it. It is evaluated to determine the ease of usage of the system.

Questionnaires are a well-established technique for collecting demographic data and users' opinion [9]. They can be used to clarify and deepen the understanding of the system under evaluation. Evaluation of our system is carried out using questionnaires to determine whether the system satisfies the user requirements.

#### **5.2 Experimental Setup**

Three sets of questionnaires were prepared to evaluate the functionalities required by the three groups of users. Three users from each of the three groups namely, student, staff and faculty evaluated our system. For the evaluation, the users were provided with the system. On using the system to carry out their tasks, the users were given a set of questionnaires to complete regarding the system. The statements were designed to obtain feedback on the fulfillment of the requirements of the system. The evaluations were aimed at determining the usefulness of automating the current system. Along with understanding if the new system has fulfilled the requirements, statements were also used to determine the ease with which all the three groups of users were able to carry out their

work when compared to the current system. We present the questionnaires in the appendix section.

Likert Scale is used to evaluate the user satisfaction of the new system. A range of 1 – 5 numbers is provided for every question where 1 represents Strongly Disagree and 5 represents Strongly Agree. This scale is used to identify the opinions of the different users on the new system. Each set of the users had a different set of statements as the requirements vary between the users.

The questionnaire for the staff is set to evaluate the ease with which they can adapt to the new system and the ease with which they can work with new system. The questionnaire for the students was set to evaluate their satisfaction with the usage of the new system. Another goal is to evaluate the design of the system in terms of the information provided on the website that aids in completing the registration successfully. The questionnaire set for the faculty evaluates the understanding of the information provided to the faculty to complete the task.

Questions common to all the three group of users were framed to obtain the feedback form the users on the possible improvements if any needed for the system and their overall opinion about the system. If they have disagreed with any of the statements (i.e., rated less than 3), they were asked to explain the reason to enable us to better understand their requirements/expectations. These additional questions are posed to obtain a better understanding on the usage of the new system on its deployment and also to determine whether or not the system satisfies the requirements of the users to complete their task.

## **5.3 Results**

### **5.3.1 Staff**

We are thankful to the three staff members from the School of EECS who kindly evaluated our system. We briefly present the results of their evaluation here. The results indicate that the users were able to effectively complete their task effectively and quickly using the new system. It was also observed that they were also able to adapt to the new

system easily (with an average rating of 4.66 for all the statements). Overall, they were satisfied with our proposed system (average rating of 4.33). They felt that including some additional functionality would enhance the usefulness of the system. They also observed that it is important to deploy the system to evaluate it. Their opinion was that the interface was easy to work with.

Some suggested improvements include: modifications to the reminder email to grade the students that is sent to the faculty at the end of the term, an option of viewing the complete list of faculty and students who have registered with them and the number of credits that they have registered. Another interesting suggestion was to provide an option to include the professors from other departments in the blanket credit registration system thus enabling the students to register with professors from non-EECS departments.

### **5.3.2 Students**

Two students from the department of computer science and one from electrical and computer engineering evaluated the system. The users were very happy with the system and all the statements about the system were rated higher than 4. All of them strongly agreed (with a rating of 5) that the system was useful to effectively complete their work. Also, the results indicate that they felt that the website was easy to use and the information was easily accessible, easy to understand and was helpful in completing the task (with an average rating of 4.66). The results further show that the users felt that the system simplified the registration process and that it has all the functionalities expected out of it (average rating was 4.33). The users rated their overall satisfaction with the system with an average of 4.33.

The feedbacks indicated that the students were happy with the system. Some suggested improvements include modifications to the UI and the email sent to the professor when the student requests an approval to include some personal comments. Another suggestion was to automate the removal of the restriction process. It is unclear at this point whether implementing this suggestion is feasible as the staff would then not be able to control the process. Yet another interesting feature that was pointed out was to enable the student to

change the number of credits should the Professor instructs him/her to do so. Under the current system, the student has to restart the entire process to get the approval. A better solution could be to provide a link by which he/she can alter the number of credits directly and the Professor could approve it.

### **5.3.3 Faculty**

We are grateful to the three faculty members from the Department of Computer Science for evaluating the proposed system in spite of their busy schedules. We present the results of their evaluation here. All of them strongly agreed with an average rating of 5 that the required information is easy to find in the proposed system, information provided is easy to understand and that the system has all the functions and capabilities expected from it. All of them strongly agreed that overall, they were satisfied with the system (average rating of 5). Two of them strongly agreed that the information available is helpful to complete their task and that with the system they were able to complete their work quickly. One of the faculty members felt that this system accelerates the process for the staff more than the faculty.

Some useful feedbacks include changing the subject of the email sent to the faculty to reflect the request for approval and to include the course number in the email. Another suggestion was to display the set of students grouped by their course number. Yet another feedback was to provide the faculty with the option of changing the number of credits that the student requests.

## **CHAPTER 6**

### **Scripting Languages**

In this section, we justify the use of PHP as our scripting language to manipulate the database. We had considered alternatives such as ASP and JSP and decided to use PHP as our scripting language. We provide an overview of the differences between PHP and the other languages and explain the reasons for preferring PHP over these other languages in our project.

#### **6.1 PHP**

Hypertext Preprocessor - PHP is a scripting language that is suitable for web development. It is a server-side cross-platform language that can be embedded into HTML. Hence, they allow the flexibility to create and edit the pages in the same way as the HTML pages. This is particularly very useful for creating dynamic web pages where the content cannot be predetermined. Simple scripts can be placed directly in the HTML files. It is an open source product and freely available on the web. Thus it has an active developing environment. Although in this project MySQL is used as the database with PHP, PHP has the ability to support many databases such as Access, Oracle. This feature provides a lot of flexibility for the developer. It is easy to make changes to the system after its installation.

To perform any server side operations PHP is very convenient as the programs are very simple. When a PHP page is accessed, the PHP code is read and parsed by the server the page resides on. The output from the PHP functions on the page is typically returned as HTML code which can be read by the browser. Because the PHP code is transformed into HTML before the page is loaded, users cannot view the PHP code on a page.

Using PHP in web applications makes the job easier. All the web pages are developed using HTML. With applications such as Dreamweaver the development of these pages can be done easily. It is not required to design the pages by coding in HTML. Instead, we

can design the pages easily using an interactive interface. For the back end part of the processing of these pages, PHP is used. Since it can be embedded in the html coding it is not required to learn the entire language to start using it. It can be learnt incrementally.

MySQL database is used to maintain the information about the Professors and their email ids. Since PHP supports MySQL database, it is used as the database to process the forms. Processing the forms is made easy due to the usage of PHP. Generation of emails automatically on submission of forms is written in PHP by a set of variable initialization and switch statements. This information is collected dynamically and the content of the email is generated. Though the process seems complicated to keep track of the information every time the student processes the form, PHP makes the process simple by the usage of the “mailto” command which emails the content to the respective professor based on the information provided. Using PHP in this project has been an advantage to bring about the functionality of the system.

## **6.2 PHP vs ASP**

ASP (Active Server Pages) is Microsoft’s scripting technology for displaying dynamic web pages. PHP is relatively faster and more stable than ASP. ASP is built on a COM-based architecture and hence whenever a program tries to connect to a database or makes a call to the COM object there is an overhead in the server. There COM overheads slow the working of the system. On the other hand, in PHP everything runs in memory space. Hence PHP code runs faster because there is no overhead of communicating with different COM objects in different processes.

ASP supports multiple programming languages. This feature makes ASP a memory intensive application than PHP’s model because each ASP language compiler runs in its own process. So when an ASP detects a begin ASP tag (<%), it needs to do a context switch out of the HTML parser, and select the appropriate compiler. Then at the end ASP tag (%>) it will do a context switch back to the HTML parser. Hence, PHP has superior memory management.

ASP runs on the Internet Information Server which needs windows N.T/2000/2003 servers. PHP could run on Linux as well. PHP is secure. IIS server has many security issues. ASP mostly uses MS-SQL server as the back end. While PHP supports many databases and MySQL is one of them and is free.

ASP is mostly used with Windows. On the other hand, PHP programs run on UNIX, Linux, Solaris and Windows operating systems. Thus, PHP is more cross platform compatible than ASP. ASP is reliable only on Microsoft Windows-based web servers.

Also, the architecture of PHP is simpler thus making the dynamic generation of UI more flexible. PHP is also highly flexible because it allows the developer to extend the software and customize it for client's requirements. There is lot of code reuse and hence it is not necessary to reinvent the wheel when it used for a project. There is an extensive library of software written in PHP which is free and available in source code form.

Coding in PHP is more efficient than in ASP. Typically the same functionality can be accomplished with less number of lines of code with PHP than with ASP. This is because the HTTP GET and POST variables are created automatically by PHP as global variables, thus eliminating the step of extracting them from the ASP Request object. PHP's HTTP header manipulation functions are easier to use than in ASP.

The cost of running an ASP-based website implies a full Windows server platform. Overall the development of ASP pages leads to higher costs, expensive software licenses and speed. Security and flexibility are also sacrificed.

### **6.3 PHP vs JSP**

JSP (Java Server Pages): It enables web developers and designers to rapidly develop and easily maintain information-rich dynamic web pages. It enables rapid development of web-based applications that are platform independent. JSP technology separates the user interface from content generation, enabling designers to change the overall page layout

without altering the underlying dynamic content. Layout can be changed without affecting the logic.

When compared to ASP, the dynamic part of JSP is written in Java, not using Visual Basic or other MS-specific language, so JSP is more powerful and easier to use. JSP is portable to other operating systems and non-Microsoft Web servers.

Like PHP, JSP allows including the logic in the design. But to achieve this it involves including things such as scriptlets, declarations and directives in the implementation. This makes coding in JSP complex.

Logic and the content of JSP programs are not well separated. Scalability of JSP programs is not very good. JSP pages often run slower than similar pages such as PHP due to it being more robust and intensive. It offers no extensibility. It does not allow reusability. PHP supports any 32-bit or better platform, whereas JSP supports only platforms that have a Java virtual machine available.

## CHAPTER 7

### Conclusion

Almost every student in the School of EECS has to deal with the process of Blanket Credit Registration. Currently, the student has to go through a laborious process that involves manually filling up the form, meeting a faculty who would be ready to approve it and then turn it in to the department. Then, the staff of the department has to maintain the list of approved students and have to manually send the emails to the Professors at the end of the quarter reminding them to grade the student.

In this project, we motivated the need of automating this process by providing a detailed overview of the problems involved in the current system. To be precise, we outlined the details of the difficult tasks associated with each of the involving users namely the *Students*, *Faculty* and *Staff*. We then presented the responsibilities of each user in the proposed system and provided the requirements of each of the user in the proposed system and the objective of the system.

We then provided details about the architecture of the proposed system and explained the tools that were used in implementing our system. We also presented the implementation details of our system. We then justified the use of PHP as our scripting language against the alternatives that we had considered for developing our system. We also provided the feedback of a sample of users that includes the students, staff and faculty of the School of EECS who were used to test the system. Overall, the feedbacks of the users are positive and if approved, the system will be employed by the department.

The design of the system facilitates modification to one component without altering the other component of the system. For instance, the use of *Smarty* and *Propel* in the system makes it possible for future enhancements to the UI without modifying the underlying database schema and vice-versa. This would enable the future developers to work on a particular module without the need for understanding the details of all the other modules of the system.

An interesting future direction is to design several test scenarios and perform an exhaustive testing on these. Also, the inputs of the users can then be used to re-design the UI if needed. Another possible and exciting future enhancement is to integrate our system with the registration process of the university. This would involve maintaining a larger database of the users involved in each of the department. This would open up a number of interesting issues ranging from tracking the dynamics of the departments (for ex., changes in the faculty, staff and students in the department) to inter-departmental registrations of the Blanket Credits.

## APPENDIX

### Evaluation Questionnaires

Students:

Overall Reaction		1	2	3	4	5
1	The system simplifies the process					
2	Website is easy to use					
3	The work can be effectively completed using the website					
4	The information needed can be found easily					
5	The information provided is easy to understand					
6	The information is helpful in completing the task					
7	The website has all the functions and capabilities that is expected out of it					
8	Overall, satisfied with the system					

Faculty:

Overall Reaction		1	2	3	4	5
1	Easy to find the required information					
2	Information provided is easy to understand					
3	Information is helpful to complete the task					
4	Able to complete the work quickly					
5	Has all the functions and capabilities expected from it					
6	Overall, satisfied with the system					

Staff:

Overall Reaction		1	2	3	4	5
1	Able to effectively complete the work using the new system					
2	Able to complete the work quickly using the new system					
3	The new system has all the functions and capabilities that are expected from it					
4	Easy to adapt to the new system					
5	Overall satisfied with this system					

Feedback Questions:

- 1) Feedback about the system?
- 2) Does it meet all the requirements?
- 3) How do you think the system can be improved?
- 4) If you have rated any of the above below 3, explain

## References

- [1] PHP – Hypertext PreProcessor : <http://www.php.net/>
- [2] JSP – Java Server Pages : <http://java.sun.com/products/jsp/>
- [3] ASP – Active Server Pages : <http://www.developer.com/net/net/article.php/865861>
- [4] Smarty – Template Engine : <http://smarty.php.net/crashcourse.php>
- [5] Propel – Object Persistence Toolkit : <http://propel.phpdb.org/trac/>
- [6] Creole – Common Database API : <http://creole.phpdb.org/trac/>
- [7] Phing - Project Build System : <http://phing.info/trac/>
- [8] PHPMailer – PHP Email Class : <http://phpmailer.sourceforge.net/>
- [9] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides,  
Design Patterns:Elements of Reusable Object-Oriented Software
- [10] Preece, Rogers, Sharp, Interaction Design beyond Human Computer Interaction