

California State University, San Bernardino

**CSUSB ScholarWorks**

---

Theses Digitization Project

John M. Pfau Library

---

2006

## Solicitation Management System

Yu-Luen Lin

Follow this and additional works at: <https://scholarworks.lib.csusb.edu/etd-project>



Part of the [Databases and Information Systems Commons](#)

---

### Recommended Citation

Lin, Yu-Luen, "Solicitation Management System" (2006). *Theses Digitization Project*. 2976.  
<https://scholarworks.lib.csusb.edu/etd-project/2976>

This Project is brought to you for free and open access by the John M. Pfau Library at CSUSB ScholarWorks. It has been accepted for inclusion in Theses Digitization Project by an authorized administrator of CSUSB ScholarWorks. For more information, please contact [scholarworks@csusb.edu](mailto:scholarworks@csusb.edu).

SOLICITATION MANAGEMENT SYSTEM

---

A Project  
Presented to the  
Faculty of  
California State University,  
San Bernardino

---

In Partial Fulfillment  
of the Requirements for the Degree  
Master of Science  
in  
Computer Science

---

by  
Yu-Luen Lin  
June 2006

SOLICITATION MANAGEMENT SYSTEM

---


A Project  
Presented to the  
Faculty of  
California State University,  
San Bernardino

---

by  
Yu-Luen Lin  
June 2006

Approved by:

  
\_\_\_\_\_  
David Turner, Chair, Computer Science

  
\_\_\_\_\_  
Arturo Concepcion

  
\_\_\_\_\_  
Tong Lai Yu

*5/30/2006*  
\_\_\_\_\_  
Date

## ABSTRACT

Solicitation Management System (SMS) is a system built to facilitate the processing of grant proposal solicitations for the Office of Technology Transfer and Commercialization (OTTC). This system is built with five user roles in mind. They are the administrator, officers, staff members, evaluators, and applicants. The administrator creates officer and staff member accounts; evaluators and applicants register themselves into the system.

In addition to basic Web page form processing, the system contains a document conversion server that converts Word documents to post script documents. The document conversion service is implemented in Windows, and is accessed from the Web application using TCP sockets.

The system was implemented using the Spring and Hibernate frameworks, which helped to reduce the lines of code needed to implement desired functionality, and provided a means to implement the user interface following a small set of well-defined patterns.

## ACKNOWLEDGMENTS

I would like to acknowledge my advisor, Dr. David Turner for all the efforts that he had devoted to make this project possible. I would also like to thank him for all the time that he put into this project and for helping me whenever I ran into a problem. I would also like to thank Dr. Concepcion and Dr. Yu for serving on my committee and for giving me guidance along the way.

## TABLE OF CONTENTS

ABSTRACT .....	iii
ACKNOWLEDGMENTS .....	iv
LIST OF TABLES .....	ix
LIST OF FIGURES .....	x
CHAPTER ONE: INTRODUCTION	
1.1 Purpose .....	1
1.2 Scope .....	1
1.3 Definition and Abbreviations .....	1
1.4 Overview .....	4
CHAPTER TWO: OVERALL DISCRPTION	
2.1 Product Perspective .....	6
2.1.1 System Interfaces .....	6
2.1.2 User Interfaces .....	7
2.1.3 Hardware Interfaces .....	20
2.1.4 Software Interfaces .....	20
2.1.5 Communications Interfaces .....	21
2.1.6 Memory Constraints .....	21
2.1.7 Operations .....	21
2.1.8 Site Adaptation Requirements .....	22
2.2 Product Functions .....	22
2.3 User Characteristics .....	22
2.4 Constraints .....	28
2.5 Assumptions and Dependencies .....	30
2.6 Apportioning of Requirements .....	30

CHAPTER THREE: SOFTWARE REQUIREMENTS  
SPECIFICATION

3.1 External Interface Requirements .....	31
3.1.1 User Interfaces .....	31
3.1.2 Hardware Interfaces .....	32
3.1.3 Software Interfaces .....	32
3.1.4 Communications Interfaces .....	33
3.2 Functions .....	33
3.2.1 User Class: System Administrator .....	33
3.2.2 User Class: Officer .....	34
3.2.3 User Class: Staff .....	41
3.2.4 User Class: Evaluator .....	44
3.2.5 User Class: Applicant .....	47
3.3 Performance Requirements .....	54
3.4 Local Database Requirement .....	55
3.5 Design Constraints .....	55
3.6 Software System Attributes .....	55
3.6.1 Reliability .....	55
3.6.2 Availability .....	55
3.6.3 Security .....	56
3.6.4 Maintainability .....	56
3.6.5 Portability .....	56

CHAPTER FOUR: DATABASE DESIGN

4.1 Data Analysis .....	57
4.2 Database Schema Logical Model - Relational Schema .....	59
4.3 Data Type and Details .....	64

CHAPTER FIVE: SYSTEM ARCHITECTURE

5.1 Database Mappings .....	77
5.1.1 One-to-One Mapping .....	77
5.1.2 Many-to-Many Mapping .....	78
5.1.3 Many-to-One and One-to-Many Mapping .....	78
5.2 SMS Controllers .....	79
5.2.1 The Controller Class .....	79
5.2.2 The ParameterizableViewController Class .....	79
5.2.3 AbstractWizardFormController .....	79
5.2.4 SimpleFormController .....	80

CHAPTER SIX: SECURITY RELATED ISSUES

6.1 Security Focus .....	81
6.1.1 Protecting User Password .....	81
6.1.2 Blocking Unnecessary Access .....	82
6.1.3 Login .....	83

CHAPTER SEVEN: MAINTENANCE MANUAL

7.1 Prerequisite Software Installation .....	85
7.1.1 Linux Operating System .....	86
7.1.2 Java Runtime Environment .....	86
7.1.3 Oracle Database .....	86
7.1.4 Tomcat .....	87
7.1.5 Microsoft Windows XP Professional .....	87
7.1.6 Microsoft Office 2003 .....	88
7.1.7 Adobe Universal PostScript Driver .....	88
7.2 Installation of This System .....	89



7.3 Changing the System .....	89
CHAPTER EIGHT: CONCLUSION AND FUTURE DIRECTIONS	
8.1 Conclusion .....	90
8.2 Future Directions .....	91
REFERENCES .....	92

LIST OF TABLES

Table 1. Structure of Table App\_AppGroup ..... 65

Table 2. Structure of Table Applicant ..... 66

Table 3. Structure of Table Application ..... 67

Table 4. Structure of Table ApplicationGroup ..... 68

Table 5. Structure of Table Application\_OttcAward ..... 68

Table 6. Structure of Table Application\_TechArea ..... 69

Table 7. Structure of Table BusArea ..... 69

Table 8. Structure of Table Evaluation ..... 70

Table 9. Structure of Table Evaluator ..... 71

Table 10. Structure of Table Evaluator\_BusArea ..... 72

Table 11. Structure of Table Evaluator\_TechArea ..... 72

Table 12. Structure of Table OttcAward ..... 73

Table 13. Structure of Table Proposal ..... 73

Table 14. Structure of Table Sol\_AppGroup ..... 74

Table 15. Structure of Table Solicitation ..... 74

Table 16. Structure of Table  
Solicitation\_OttcAward ..... 75

Table 17. Structure of Table TechArea ..... 75

Table 18. Structure of Table Users ..... 76

LIST OF FIGURES

Figure 1. Deployment Diagram ..... 7

Figure 2. Solicitation Management System Login  
Page..... 9

Figure 3. Registration Page .....10

Figure 4. Evaluator Register Wizard, Page 1 .....11

Figure 5. Evaluator Register Wizard, Page 2 .....12

Figure 6. Evaluator Register Wizard, Page 3 .....13

Figure 7. Evaluator Register Wizard, Page 4 .....14

Figure 8. Evaluator Register Wizard, Page 5 .....15

Figure 9. Administrator Home Page .....16

Figure 10. Officer Home Page .....17

Figure 11. Staff Home Page .....18

Figure 12. Evaluator Home Page .....19

Figure 13. Applicant Home Page .....20

Figure 14. Use Case Diagram - Administrator .....23

Figure 15. Use Case Diagram - Officer .....24

Figure 16. Use Case Diagram - Staff .....25

Figure 17. Use Case Diagram - Applicant .....26

Figure 18. Use Case Diagram - Evaluator .....26

Figure 19. Officer List Solicitations .....35

Figure 20. Officer Create Solicitation Wizard  
Page 1.....36

Figure 21. Officer Create Solicitation Wizard,  
Page 2.....37

Figure 22. Officer Create Solicitation Wizard,  
Page 3.....38

Figure 23. Officer Edit Assign Evaluators .....	39
Figure 24. Officer View Application .....	40
Figure 25. Staff List Solicitations .....	42
Figure 26. Staff Generate Reports .....	43
Figure 27. Evaluator List Assigned Work .....	45
Figure 28. Evaluator Evaluates Application .....	46
Figure 29. Evaluator Views Evaluation .....	47
Figure 30. Applicant List Open Solicitations .....	48
Figure 31. Applicant Create Application Wizard Page 1 .....	49
Figure 32. Applicant Create Application Wizard, Page 2 .....	50
Figure 33. Applicant Create Application Wizard, Page 3 .....	51
Figure 34. Applicant Create Application Wizard, Page 4 .....	52
Figure 35. Applicant Create Application, Final Page .....	53
Figure 36. Applicant View Application .....	54
Figure 37. Entity Relationship Diagram .....	58
Figure 38. Database Relational Schema .....	62
Figure 39. Relations Between Tables .....	64
Figure 40. Login Sequence Diagram .....	84

## CHAPTER ONE

### INTRODUCTION

#### 1.1 Purpose

This project was done with three purposes in mind. First, an effort is made to make the new version of the SMS more user friendly, so that it is easier for users to use and comprehend. Second, the system undergoes a makeover so that the internal processing of data is sped up. Third, newly required functions are added to the new version, so that it can facilitate the processing of the solicitation.

#### 1.2 Scope

The scope of this project includes rewriting the system using the Spring and Hibernate frameworks, so that it includes all the functionalities of the current system plus some additional reports and modifications to the user interface as requested by OTTC staff. Thus, the project is an updated version of the SMS.

#### 1.3 Definition and Abbreviations

SMS - Solicitation Management System.

OTTC - Office of Technology Transfer and Commercialization.

It is an office that assists in transitioning promising new technologies from government and

academic laboratories alike into full commercialization.

Spring Framework - The Spring Framework is a layered Java/J2EE application framework. Spring is an Inversion of Control (IoC) container designed to work with JavaBeans and is based upon org.springframework.beans package.

Hibernate - Hibernate is an Object-relational-mapping (ORM) framework for Java. It lets developers develop persistent classes following common Java idiom.

Apache - A public-domain open source Web server developed by a loosely-knit group of programmers. The source code is freely available, anyone can adapt the server for specific needs, and there is a large public library of Apache add-ons. In many respects, development of Apache is similar to development of the Linux operating system. The original version of Apache was written for UNIX, but there are now versions that run under OS/2, Windows and other platforms.

API - Application Programming Interface.

HTML - Hyper Text Markup Language.

Java - An object oriented language developed by Sun Microsystems, Java programs are capable of running on

most popular computer platforms without the need for recompilation.

Java Servlet - A Java application that runs in a web server or application server. It provides server-side processing which usually is used to access a database or perform e-commerce processing.

JSP - Java Server Page, an extension of the Java server technology from Sun that provides a simple programming vehicle for displaying dynamic content on a web page.

JDBC - (Java Database Connectivity) is a programming interface that allows Java applications to access database through the SQL language.

MySQL - MYSQL is a software that delivers a very fast, multi-threaded, multi-user, and robust SQL (Structured Query Language) database server.

Tomcat - Tomcat is an application server from the Apache Software Foundation that executes Java servlets and renders Web pages that include Java Server Page coding. Described as a "reference implementation" of the Java Servlet and the Java Server Page specifications, Tomcat is the result of an open collaboration of developers and is available from the Apache Web site in both binary and source versions.

UML - Short for Unified Modeling Language, a general-purpose notational language for specifying and visualizing complex software, especially large, object-oriented projects. UML builds on previous notational methods such as Booch, OMT, and OOSE.

Use Case Diagrams - A diagram provided by the UML to facilitate the process of requirements gathering. The use-case diagram models the interactions between the system's external clients and the use cases of the system. Each use case represents a different capability that the system provides to its clients.

CCAT - Center for the Commercialization of Advanced Technology.

#### 1.4 Overview

The Solicitation Management System (SMS) was built to facilitate the processing of grant proposal solicitations. The system created within this project comprises the third version of SMS, and was developed collaboratively with Dr. David Turner, Yu-Luen Lin, Xinsheng Chen, and Lu-Yi Wu.

Version one of the SMS was built by Dr. David Turner, Rajiv Sadagopan and Sathya Prasad. This version was used in a small demonstration project by the Office of Technology Transfer and Commercialization (OTTC) to run a grant proposal solicitation. The subsequent version,



version 2, was built by Dr. David Turner, Rick Pallow and Jonathan Wang. SMS version 2 is currently the system in production and is used to run regular grant proposal solicitations.

Version 2 has been in production for some time now and some feedback was given by the users. Based upon this feedback, version 3 of the SMS is written with newly requested components are added.

## CHAPTER TWO

### OVERALL DISCRIPTION

The purpose of this software is to aid the processing of a solicitation for the Office of Technology Transfer and Commercialization.

#### 2.1 Product Perspective

This system was initially built to aid the solicitation procedure for OTTC. Within this project, we have expanded the system, so that it also serves the needs of the OTTC's sister organization CCAT. In fact, SMS has the potential to serve as a general purpose solicitation management system that can be used by other organizations as well.

This system is currently being used by OTTC to manage the submission and evaluation of grant proposals. An improvement in processing speed along with newly added functionalities can make this system more user friendly and more productive. The value of this system lies in the amount of time that can be saved for the staff of OTTC while processing a solicitation.

##### 2.1.1 System Interfaces

The deployment diagram of the SMS is shown in Figure 1. It consists of four 4 components. These are: the client

machine, the web server, the database server and the Word-to-post script conversion server. When the user wants to access our system, the client machine connects to the web server through the HTTPS protocol. The web server then connects to the database server through the TCP protocol to verify login and process information. The web server also connects to the document conversion server through the application-specific protocol described in Section 3.1.3 when the user uploads a Word document.

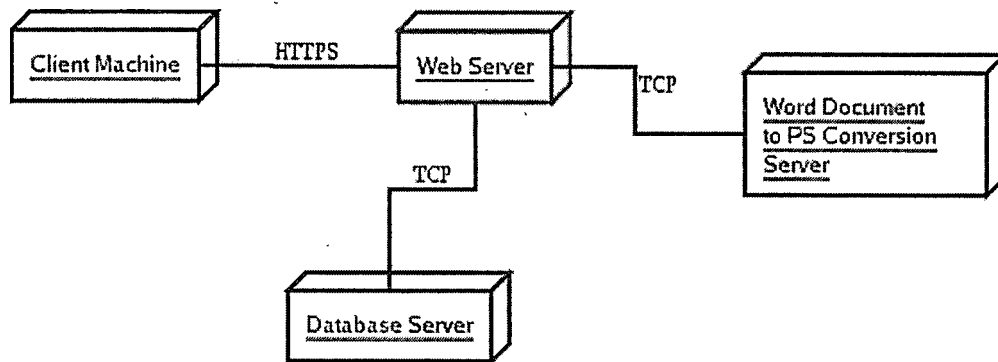


Figure 1. Deployment Diagram

### 2.1.2 User Interfaces

Each user logs in using a user name and password, and is redirected to his home page according to his role. From his home page, the user can access functions of the system

that are specific to his role. The user interface consists of roughly 150 web pages due to the large scale of this system.

2.1.2.1 Login Page. The main user login page, shown in Figure 2, is the main entry to the solicitation management system. The main page simply consists of a login page for all users. On the top of the login page, there is a banner that shows the agency logo. Users of the solicitation management system can log themselves into the system through this page. After a successful login, users are redirected to their homepage according to their user role. In situations that users forget their password, they can use the "forgot password" link to reset it.

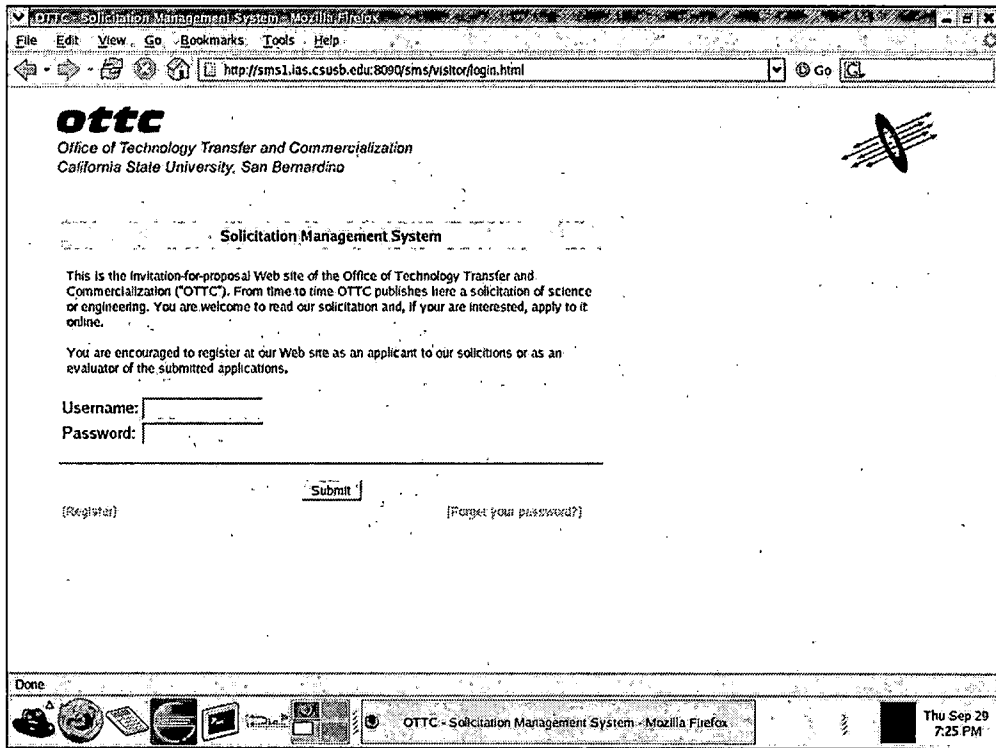


Figure 2. Solicitation Management System Login Page

2.1.2.2 Registration Page. Registration is the page (shown in Figure 3) where potential applicants and evaluators can register themselves to the solicitation management system. The user is prompted to fill in some personal information during the registration process. After the user successfully registers, the user will be redirected to his home page.

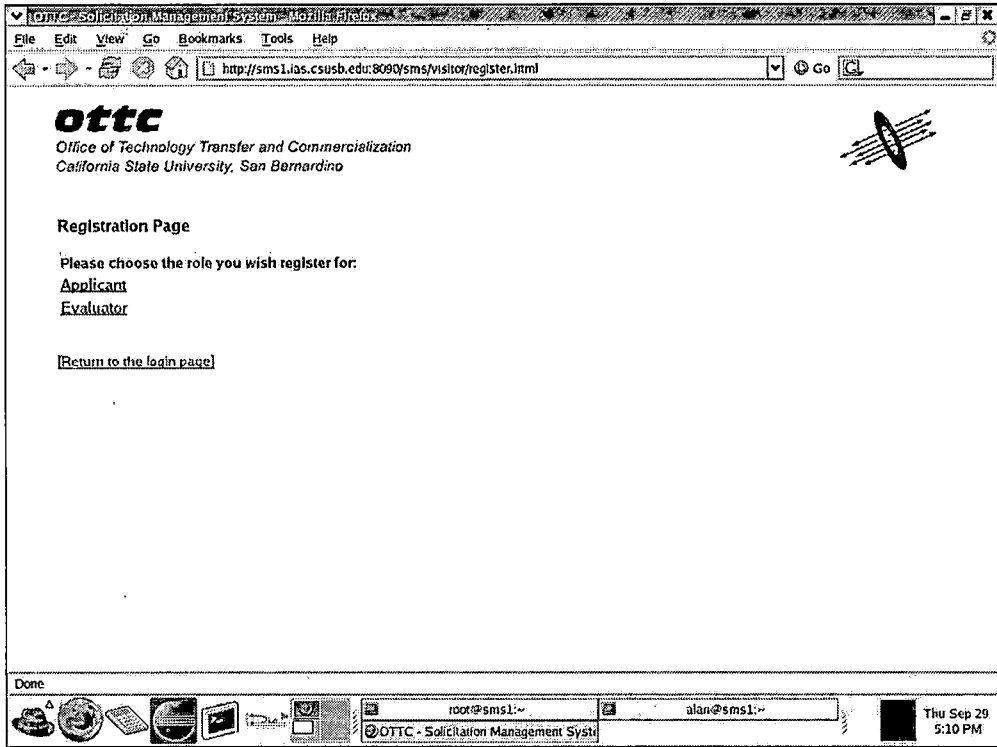


Figure 3. Registration Page

Figure 4, as shown below, is the first page of the evaluator registration sequence. On this page the user is asked to enter a username, email and password. For precautions, the user is asked to type his password twice in case a typo occurs.

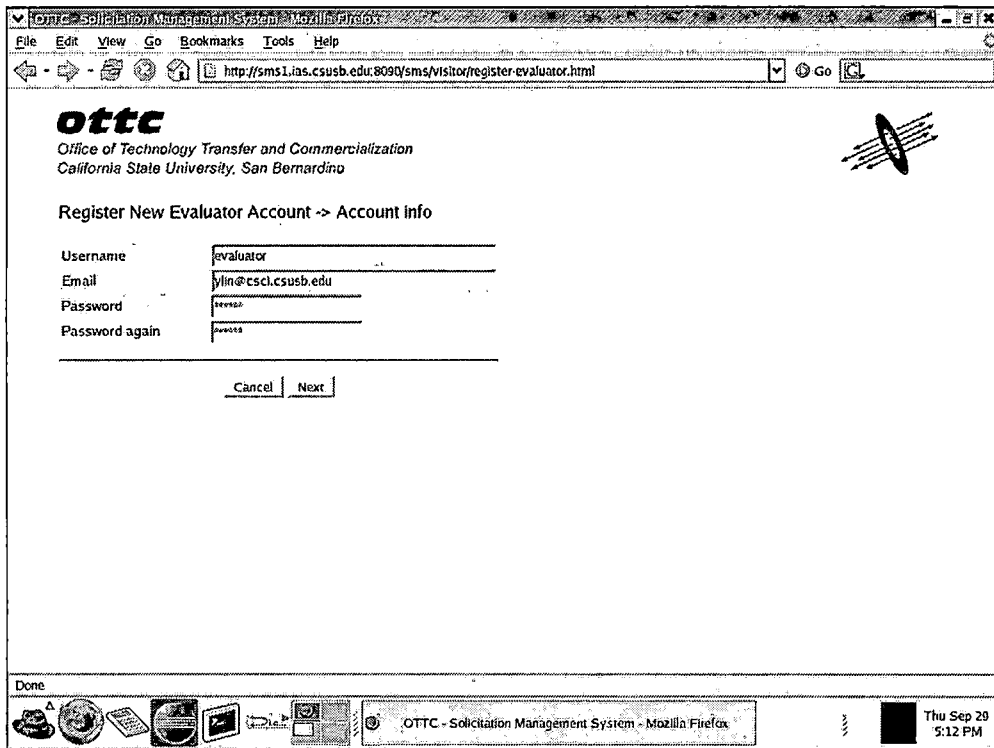


Figure 4. Evaluator Register Wizard, Page 1

Figure 5 shows the second page of the evaluator registration sequence. Here the user is asked to enter his personal information, such as name, address, phone, etc.

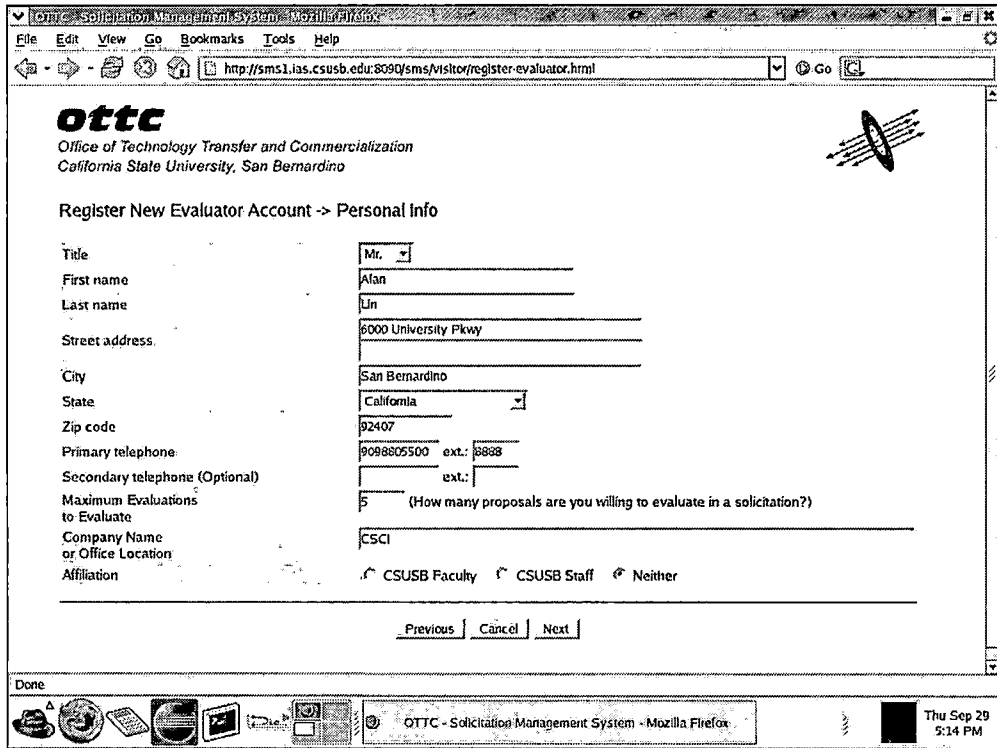


Figure 5. Evaluator Register Wizard, Page 2

Figure 6 shows the third page of the evaluator registration sequence. The user will be asked to specify whether he is a technical reviewer, a business reviewer, or both a technical and business reviewer.



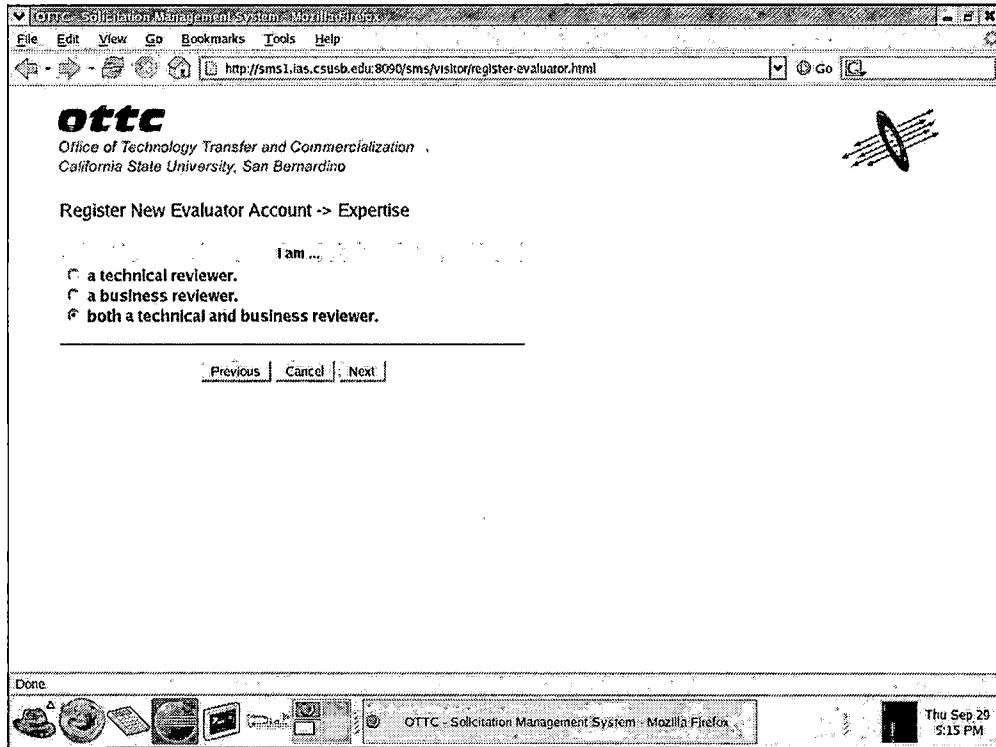


Figure 6. Evaluator Register Wizard, Page 3

Figure 7 is the fourth page of the evaluator registration sequence. The user is asked to specify his technical areas of expertise on this page.

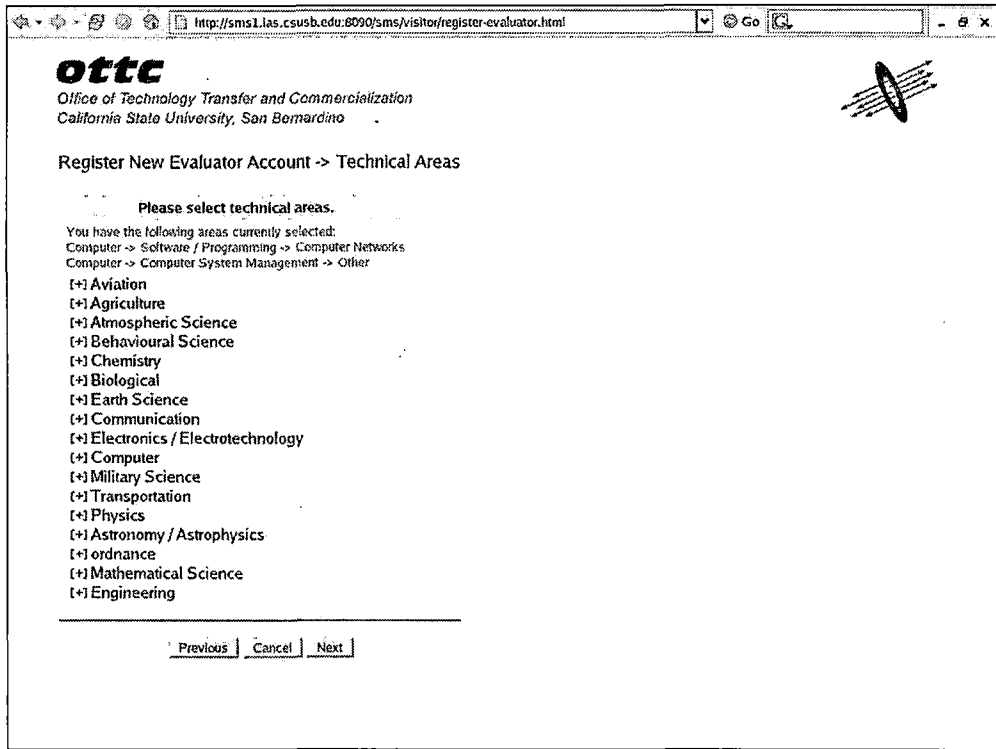


Figure 7. Evaluator Register Wizard, Page 4

Figure 8 is the last page of the evaluator registration sequence. The user is asked to specify his business areas of expertise in the case that he identified himself as either a business reviewer or both a business and technical reviewer.

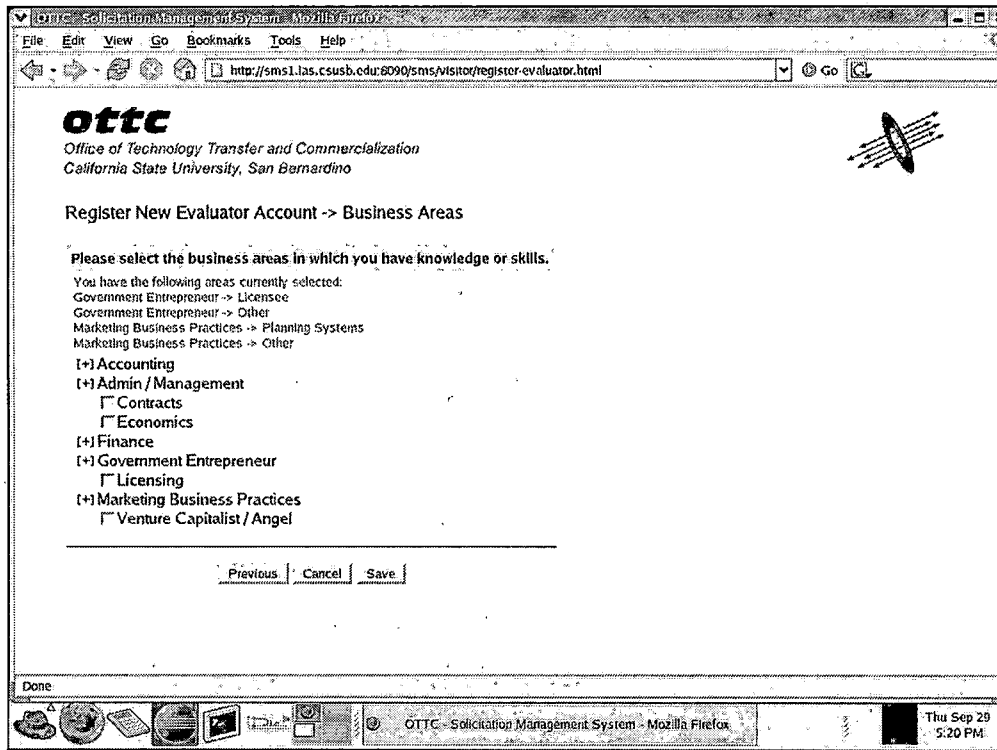


Figure 8. Evaluator Register Wizard, Page 5

2.1.2.3 User Home Page. The home page is the first page that the user sees after he logs in. From this page, the user can find links to all the functions that are provided for his user role. Figure 9 shown below is the administrator home page. From this page, the administrator can access all the functions that are available to him.

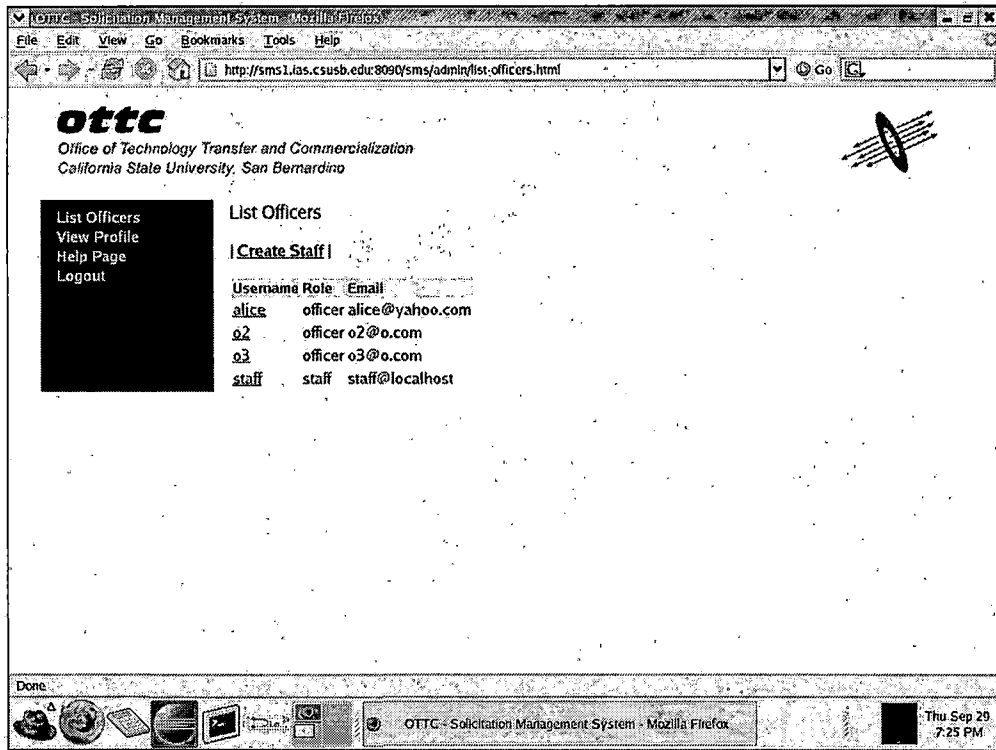


Figure 9. Administrator Home Page

Figure 10 shown below is the officer home page. From this page, officers can access functions that are specific to their user role. On this page, the officer sees a list of all solicitations in the system.

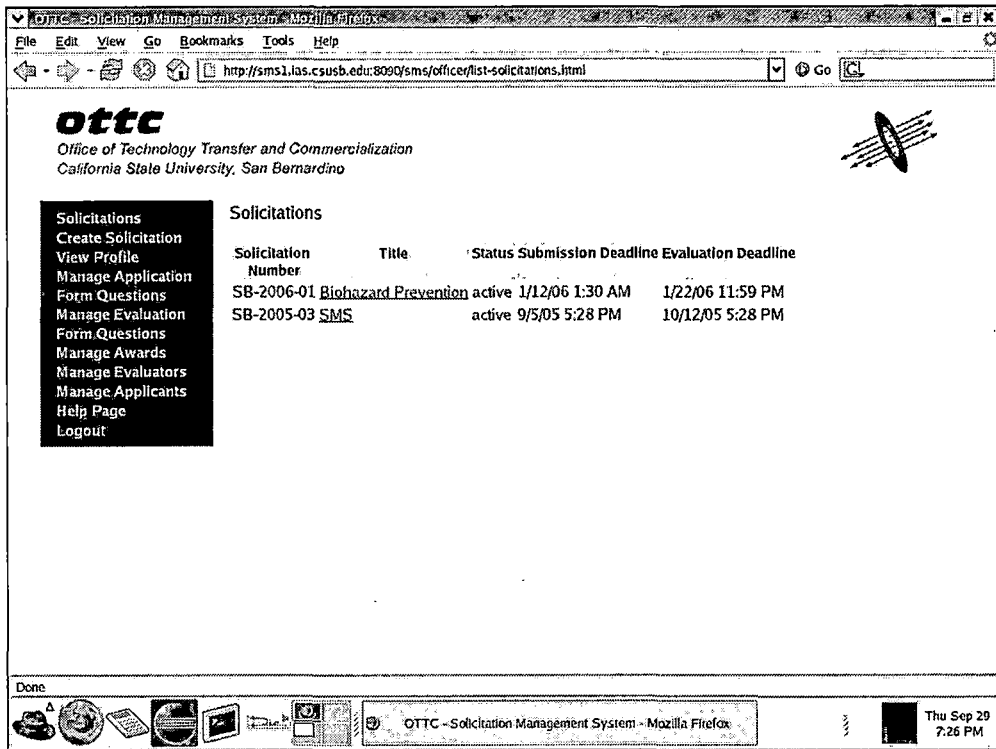


Figure 10. Officer Home Page

Figure 11 shown below is the staff home page. From this page, staff members can access functions that are specific to their user role. On this page, the staff member sees a list of all solicitations in the system.

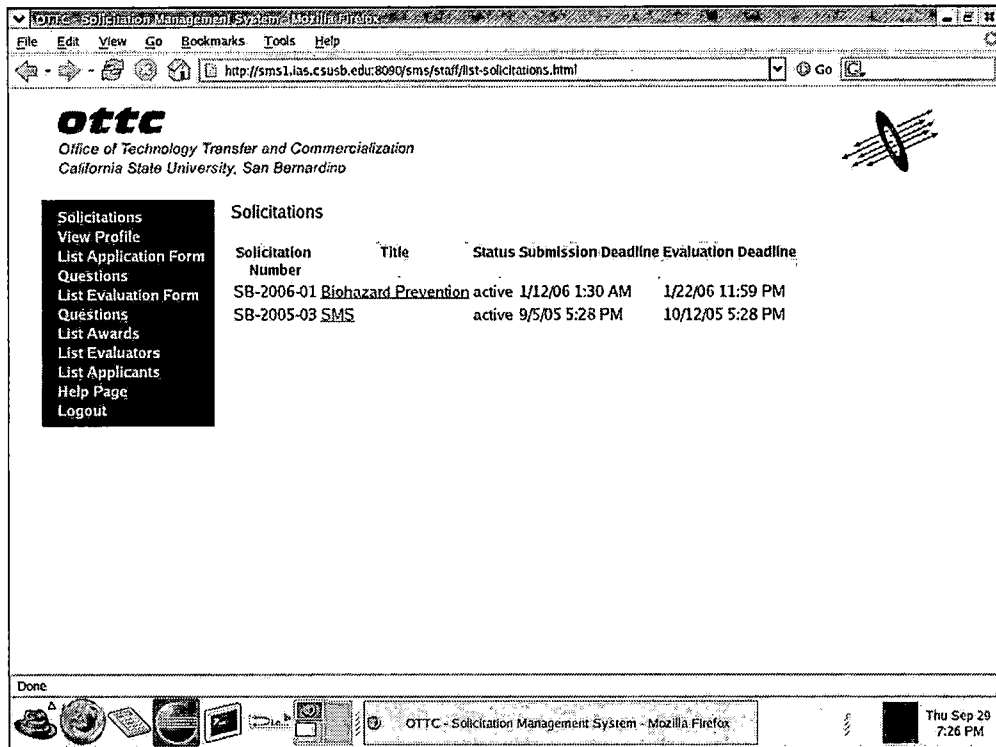


Figure 11. Staff Home Page

Figure 12 shown below is the evaluator home page. From this page, evaluators can access functions that are specific to their user role. On this page, the evaluator sees a list of work assigned to him.

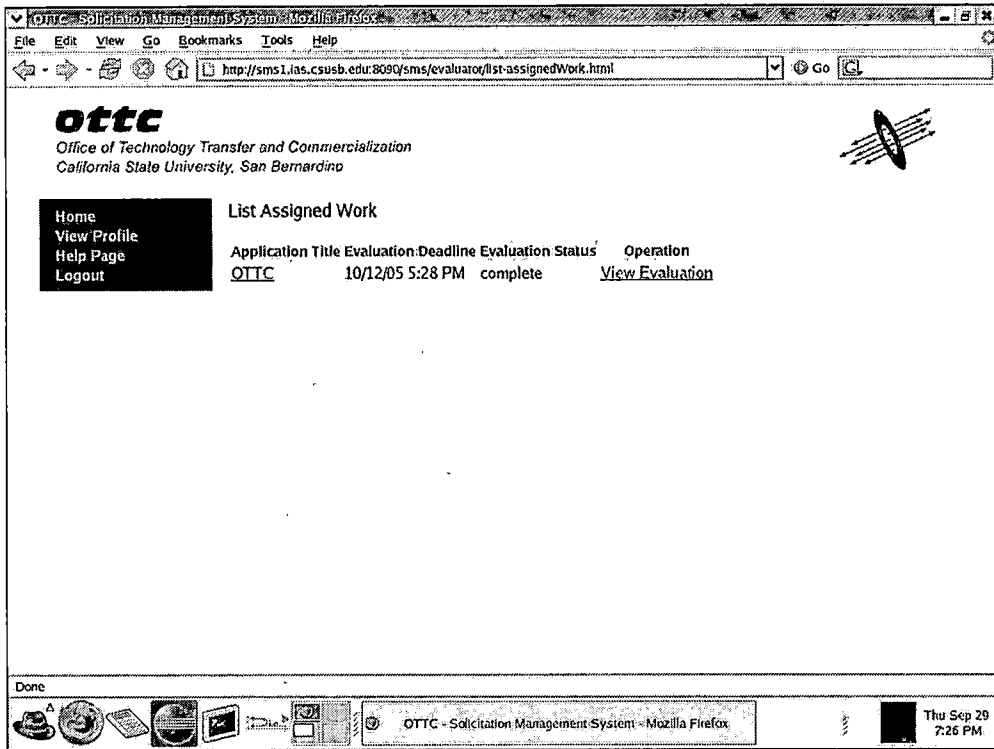


Figure 12. Evaluator Home Page

Figure 13 shown below is the applicant home page. From this page, applicants can access functions that are specific to their user role. On this page, the applicant can see all open solicitations that he can apply to and a list of applications that he has already submitted.

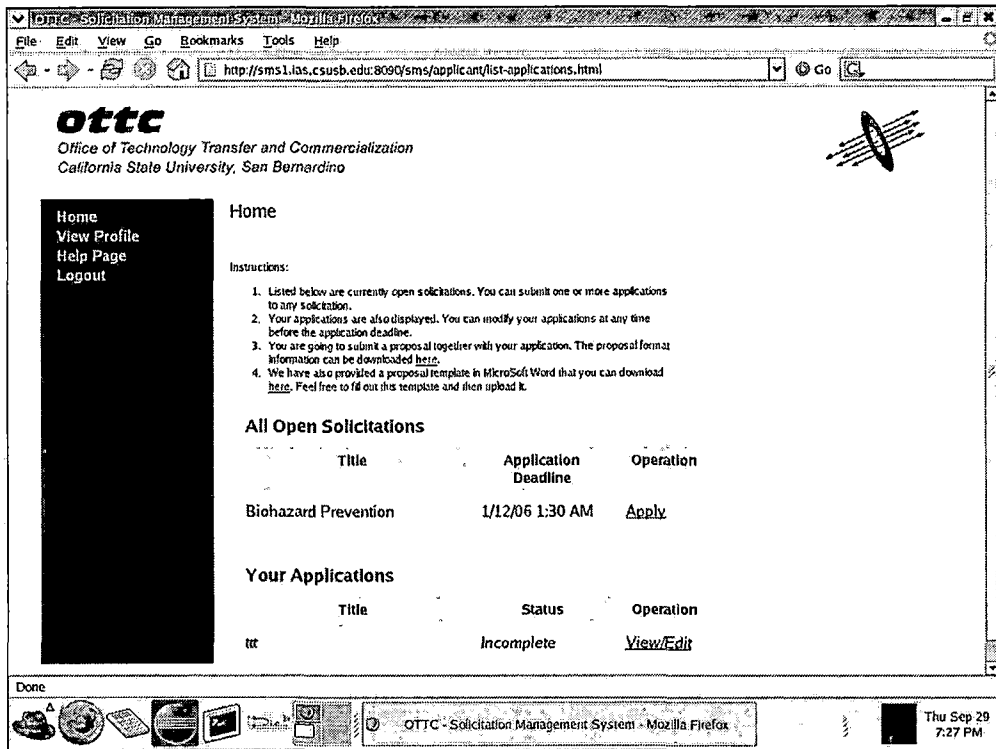


Figure 13. Applicant Home Page

### 2.1.3 Hardware Interfaces

The system does not require specific hardware interfaces. The only requirement is that the client has a computer that has Internet access. Hence, any operating system running on any hardware architecture will be available to access the system.

### 2.1.4 Software Interfaces

The system is designed to run on two different machines. The main part of the system runs on the Linux



server and the file converting part of the system runs on Windows XP Professional with Microsoft Office 2003 installed.

#### 2.1.5 Communications Interfaces

Any web browser such as Mozilla or Microsoft Internet Explorer can access the system. More recent versions of the various browsers are preferred, as the outdated version of the browser can lead to unexpected results in accessing certain functions in the system.

#### 2.1.6 Memory Constraints

The primary memory for the solicitation management system should be at least 256 MB. The size of the secondary storage depends on the software installed and the database that will hold all records. A initial estimate indicates that at least 20 GB will be needed.

#### 2.1.7 Operations

The solicitation management system is an easy-to-use, self-explanatory system. The user can click on the operation he wants and interact with the system. However, in some cases the user might have questions, for which a help page is provided for each user role.

#### 2.1.8 Site Adaptation Requirements

The solicitation management system does not have any site adaptation requirements. However, a screen resolution of at least 600 X 800 is encouraged.

### 2.2 Product Functions

This product includes several functions. From the organization's end, it allows officers to process solicitations and generate reports. From the user's end, it allows applicants to submit applications and evaluators to submit evaluations.

### 2.3 User Characteristics

The SMS system facilitates the processing of a solicitation. It serves users from five different role groups. These roles include applicant, evaluator, admin, officer, and staff. Each user role can log into his account from the login page and modify his own information, and is provided with a help page to guide the user.

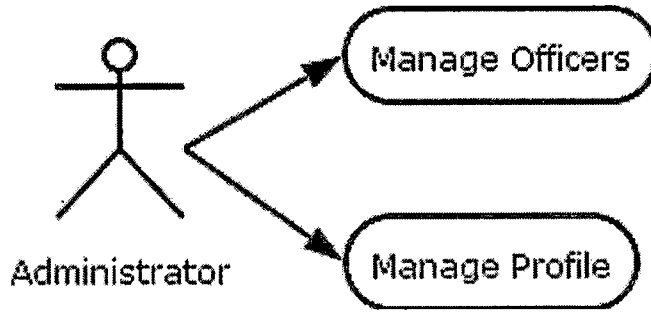


Figure 14. Use Case Diagram - Administrator

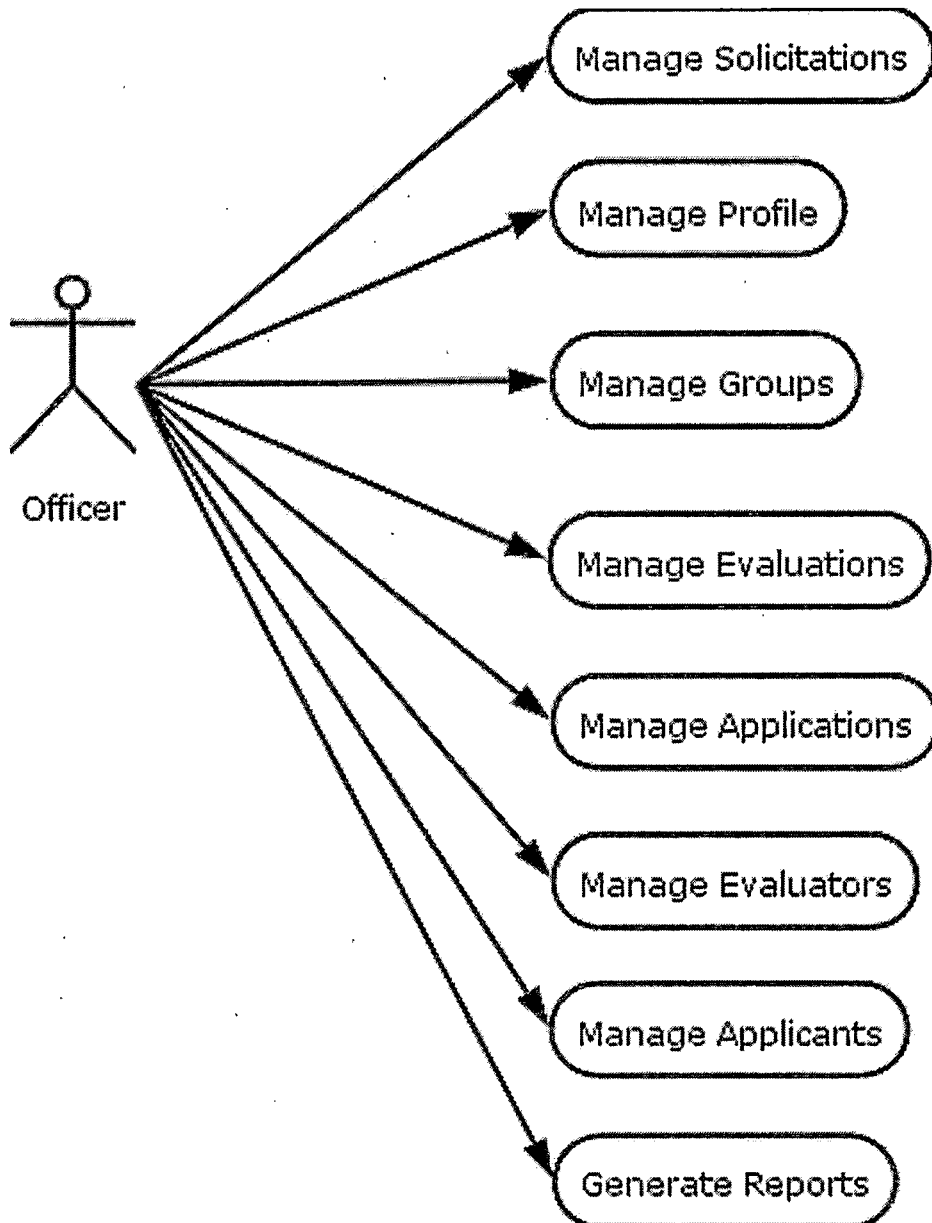


Figure 15. Use Case Diagram - Officer

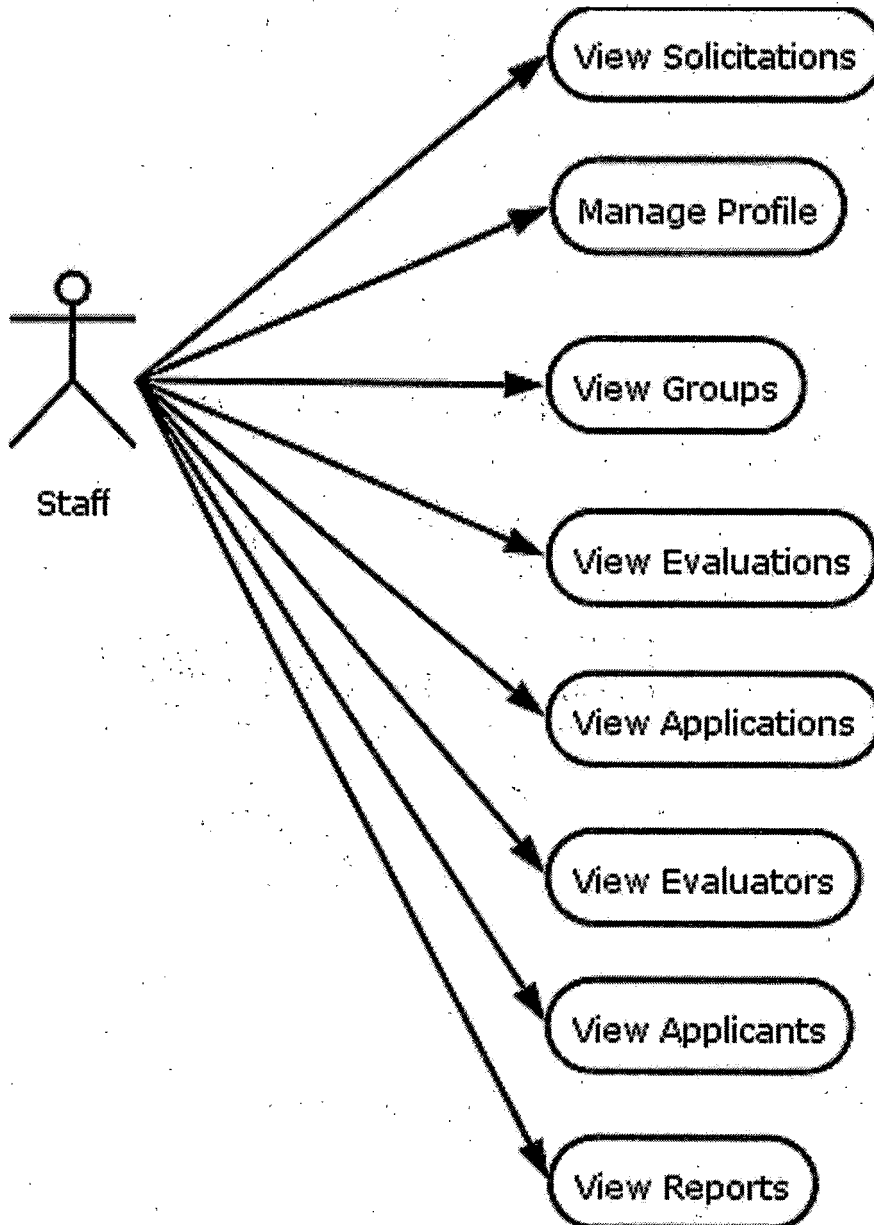


Figure 16. Use Case Diagram - Staff

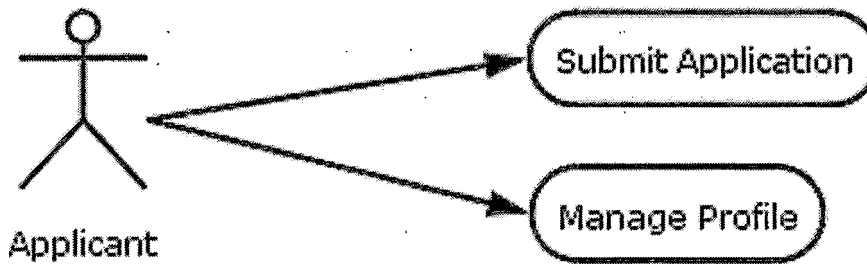


Figure 17. Use Case Diagram - Applicant

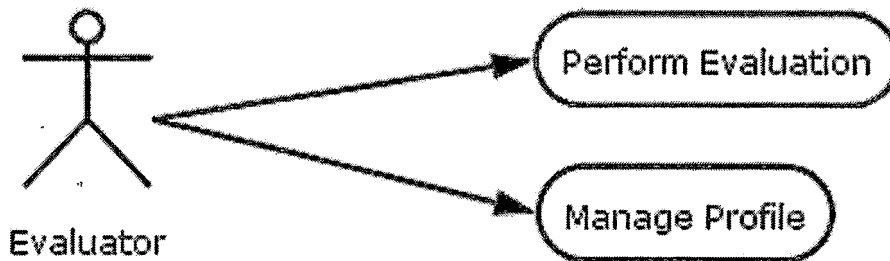


Figure 18. Use Case Diagram - Evaluator

A new user can either register himself as an applicant or an evaluator. If a user registers as an applicant (shown in Figure 17), he can use this system to apply to open solicitations and submit his application and proposal online. This system accepts proposals in either a word (doc) format or a pdf format.

If a user chooses to register as an evaluator (shown in Figure 18), he can login to the system to evaluate applications once they have been assigned to him. He can start this process by reading the applicant proposal and then answering the evaluation questions assigned to the specific solicitation.

The administrator (shown in Figure 14) is responsible for managing officers and staff members. The administrator can manage officers and staff members through various operations. These operations include viewing a list of officers and staff members, viewing the detailed information of a specific officer or staff member, creating a new account for an officer or staff member, editing information for a specific officer or staff member, and finally deleting an account of an officer or staff member.

The staff member (shown in Figure 16) can list and view many components of the system. These components include the solicitation, applications, awards, evaluators, applicants, and evaluations. The only components that the staff member can modify are his or her personal profile, which includes password and email address.

The officer (shown in Figure 15) carries out the main functionality of the system. An officer can do anything a

staff member can do. Beside that, an officer can execute create, edit and delete operations. These operations can apply to solicitations, applications, awards, evaluators, applicants, and evaluations. However, some rules apply while executing those operations. These rules are listed in the constraints area.

One of the greatest values of this system lies with the officer's ability to assign evaluators to applications. Each application is required to identify the technology areas related to the proposal. This is also true for evaluators; upon registration, they need to specify technology areas for which they have expertise. One job of this system is to match evaluators to applications. The evaluators will be displayed with matching score and organized into three categories: evaluators with both business and technical expertise, evaluators with only technical expertise, and evaluators with only business expertise. With automated matching, the solicitation process can be sped up significantly when compared to human matching.

#### 2.4 Constraints

The constraints for this project are mainly set for users.



All Roles:

1. All roles can change their own password, but it has to be at least six characters.

Admin:

1. Admin can not create an officer or staff member if a duplicate username is used.

Applicant:

1. An applicant must finish the whole sequence of pages for application in order for the application to be saved. However, this does not require the applicant to upload the proposal for his application to be saved.

2. An applicant can only upload his proposal in one of the two formats: pdf or doc.

3. An applicant can edit his application only before the submission deadline.

Evaluator:

1. An evaluator can only evaluate after the submission deadline and before the evaluation deadline.

2. After an incomplete evaluation has been submitted, the evaluator can change or edit the evaluation before the evaluation deadline.

3. After a complete evaluation has been submitted, the evaluator can no longer modify the evaluation.

Officer:

1. An officer can not create a solicitation without all of its properties defined.

2. An officer can not delete a solicitation when there has been one or more applications submitted to it.

3. An officer can not reassign an award after an application has been submitted to the solicitation.

4. An officer can not delete an evaluator who has a submitted evaluation.

5. An officer can not delete an applicant who has a submitted application.

## 2.5 Assumptions and Dependencies

The project is built using the Spring Framework and the Hibernate framework. The dependencies of this project are the dependencies of the frameworks that we are using.

## 2.6 Apportioning of Requirements

No requirements specified in this software can be delayed until a future version without prior permission of the client.

CHAPTER THREE  
SOFTWARE REQUIREMENTS SPECIFICATION

3.1 External Interface Requirements

3.1.1 User Interfaces

Each page is provided with several user friendly interface components that guides the user throughout the system. These components include a static menu, a trail of breadcrumbs, and a dynamic sub-menu.

A static menu is provided for each page and is located of the left of the page. The content of the submenu differ from role to role. The user can use the static menu to navigate to the pages or functionality that he desires.

Also on each page is a trail of breadcrumb. The pupose of the breadcrumb is to display to the user which level of submenu he is currently at and to provide the user with a mean to trace back from where he came from if necessary.

Finally, another user friendly interface component is the dynamic sub-menu. The dynamic sub-menu displays the functionality relative to this page. Since each page has different functionalities and serves different purposes, the sub-menu changes dynamically.

The screen format is a regular full-screen Web page. The screen can be minimized on the task bar, maximized to a full screen and resized to a window.

When a user input something wrong either in format or content, an error message will be displayed to notify the user.

There are instructions throughout the system that guides users to complete his task.

### 3.1.2 Hardware Interfaces

Two Dell Server computers are used as servers.

Each server has at two network connectors and four USB connectors.

The servers are connected to the Internet. Their IP addresses are: 139.182.139.150 and 139.182.139.152.

### 3.1.3 Software Interfaces

The main system is installed on a Linux RedHat Enterprise 3.0 platform. To support the system, Tomcat 5.0.28 and Oracle 9i are installed. Java Database Connectivity (JDBC) is used to connect the Web application to the Oracle 9i database system.

A Web browser is installed on the user systems, which is used to connect to the Tomcat server.

The Oracle protocol is used as a connection between tomcat and the database.

The connection between Web application and document conversion server is done using an application-specific protocol over TCP. A single-threaded server client running in the Web application makes a single connection to the document conversion server. For each document to convert, the Web application sends a 8-byte integer in network byte order representing the size of the Word document. Then it sends the bytes for the Word document. After receiving the document, the server returns a 4-byte integer status code, also in network byte order; zero indicating success, non-zero indicating failure. If success is indicated, the server sends an 8-byte integer representing the number of bytes in the postscript file. Then it sends those bytes.

#### 3.1.4 Communications Interfaces

The System uses TCP/IP (Transmission Control Protocol/Internet Protocol) to communicate over the Internet.

### 3.2 Functions

#### 3.2.1 User Class: System Administrator

A System administrator is responsible for managing the system as a whole.

3.2.1.1. The system administrator is checked for his identity with the login page before he is allowed to use this functionality.

3.2.1.2. The system administrator is able to add, remove and/or edit an officer or staff account.

3.2.1.3. The system administrator is able to create a password for any staff member or officer. He is able to assign a new password for any staff member or officer if, in a special case, the staff member forgets his password.

3.2.1.4. The system administrator is able to view all the accounts created for the staff members and officers.

3.2.1.5. The system administrator is able to edit his own personal information, including his profile and password.

### 3.2.2 User Class: Officer

An officer can operate all functions that are relevant to processing a solicitation.

3.2.2.1. An officer is checked for his identity with the login page before being able to access this functionality.

3.2.2.2. An officer is able to list, view, create, edit, and delete a solicitation. Figure 19 is the Web page that officers use to list all solicitations in the system. To create a solicitation, an officer must enter

information through a series of Web pages. The first page (Figure 20) allows the officer to specify detailed solicitation information which includes the title, submission deadline, evaluation deadline, and status.

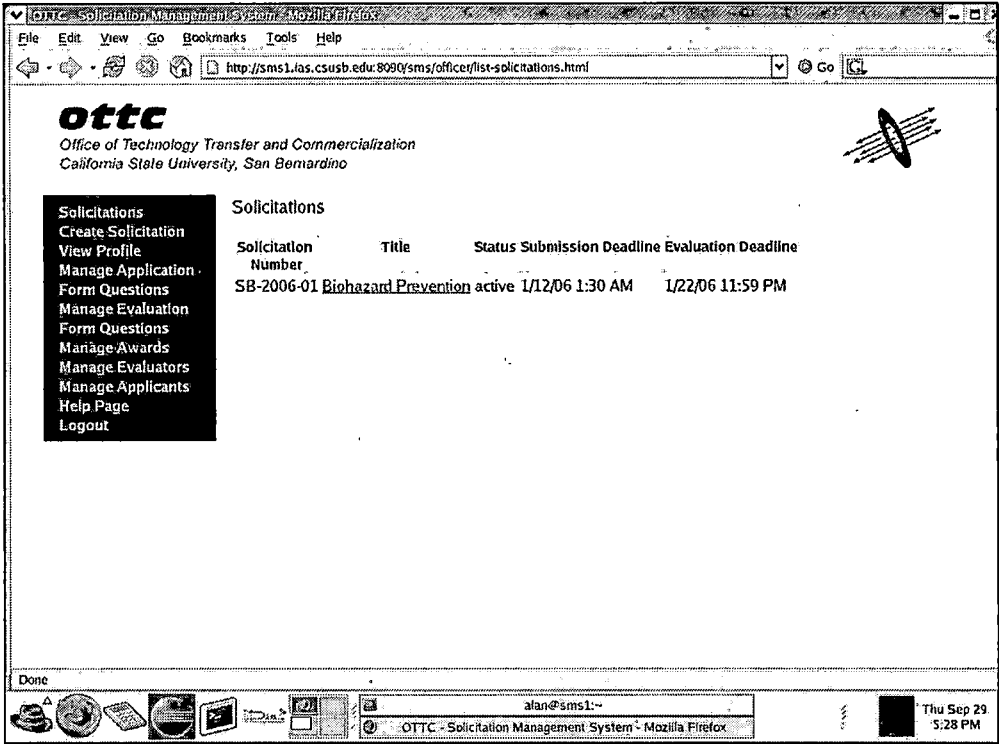


Figure 19. Officer List Solicitations

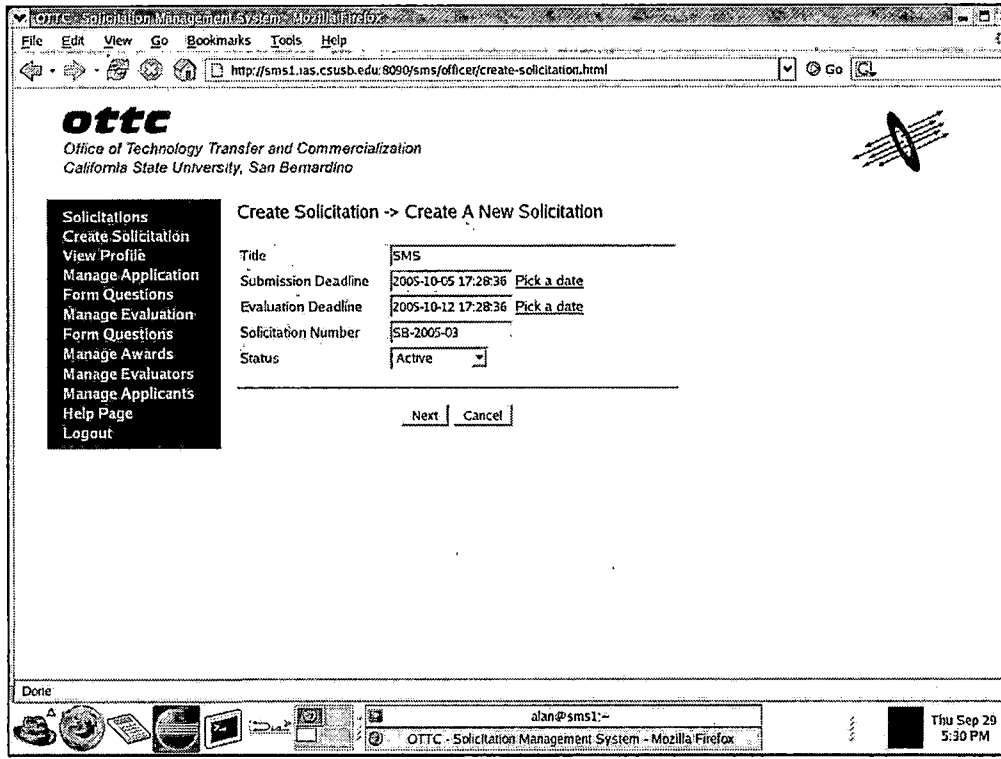


Figure 20. Officer Create Solicitation Wizard Page 1

Figure 21 shows the second page of the solicitation creation sequence. The officer is asked to select awards that will be displayed to the applicant when an applicant applies for a solicitation.



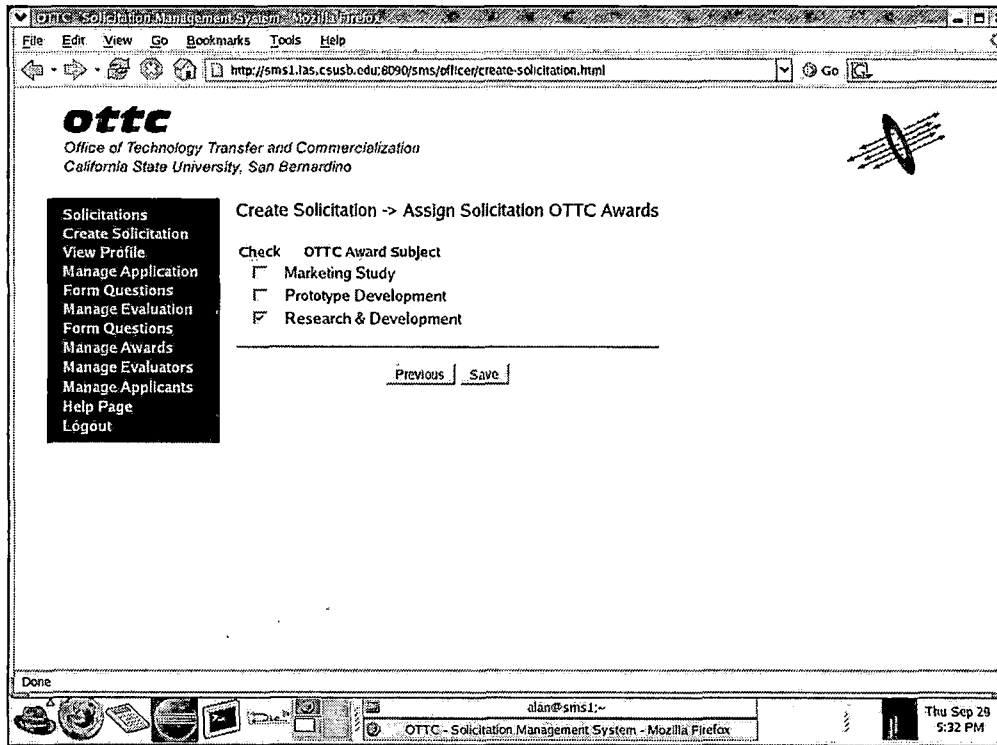


Figure 21. Officer Create Solicitation Wizard, Page 2

Figure 22 shows the last page of the solicitation creation sequence. The officer is asked to select the groups that are associated with this specific solicitation.

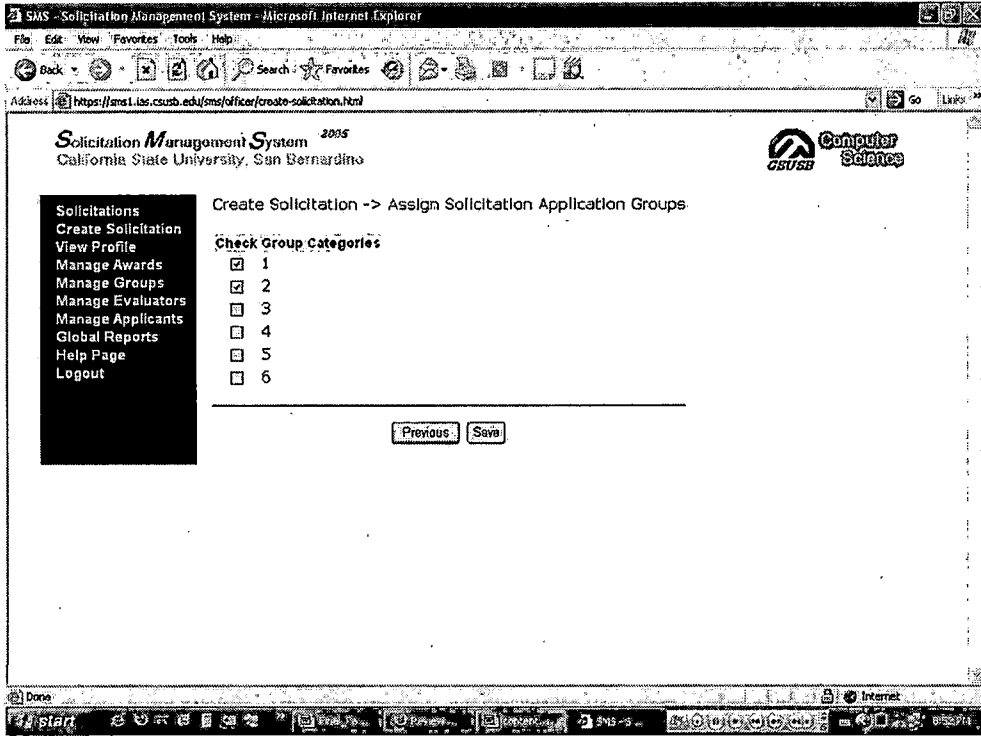


Figure 22. Officer Create Solicitation Wizard, Page 3

Figure 23 shown below is the assign evaluator to an application page. The officer can assign evaluator to applications according to the information provided on this page.

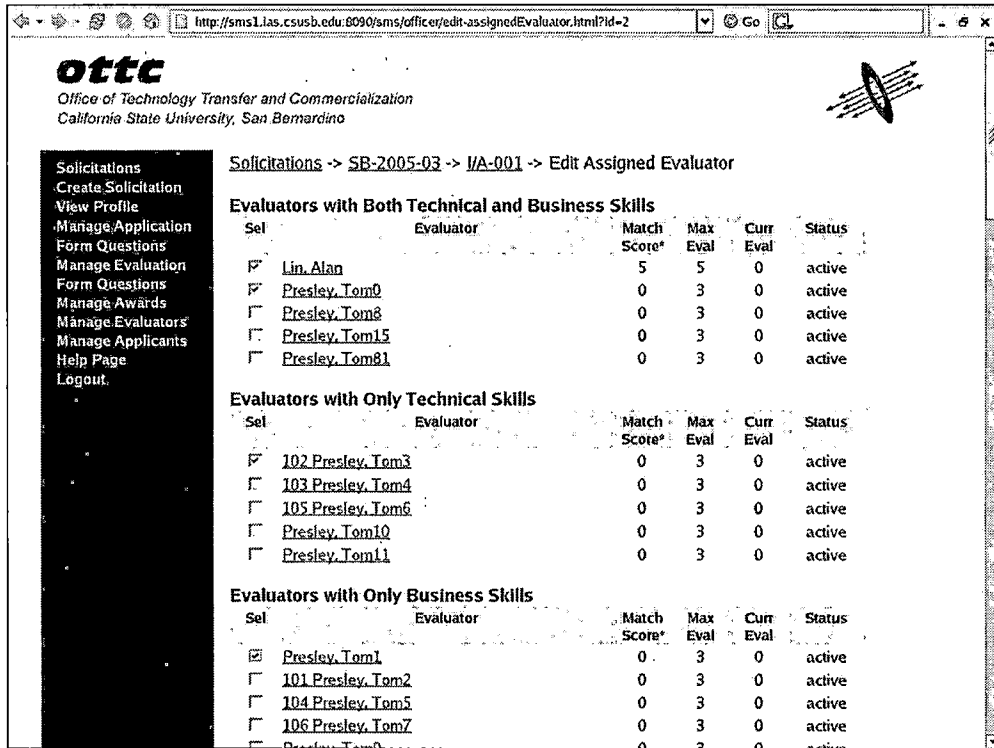


Figure 23. Officer Edit Assign Evaluators

3.2.2.3. An officer is able to list, view, create, edit, and delete awards.

3.2.2.4. An officer is able to list, view, edit, and delete applicant related information.

3.2.2.5. An officer is able to list, view, edit, and delete evaluator related information.

3.2.2.6. An officer is able to list, view (shown in Figure 24), and delete applications that an applicant has submitted.

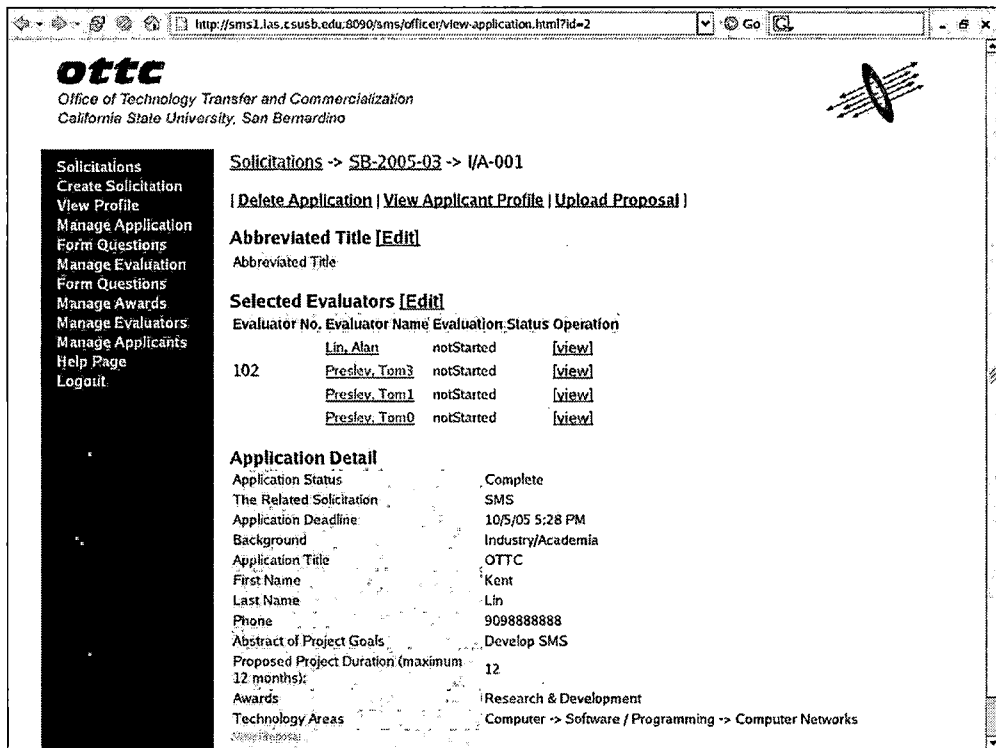


Figure 24. Officer View Application

3.2.2.7. An officer is able to list, view, and delete evaluations that an applicant has submitted.

3.2.2.8. An officer is able to edit his own personal information including his profile and password.

3.2.2.9. An officer is able to upload a proposal for an applicant.

3.2.2.10. An officer is able to generate reports for a specific solicitation.

3.2.2.11. An officer is able to list, view, create, edit, and delete application groups for a specific solicitation.

### 3.2.3 User Class: Staff

A staff member do all that an officer can do except for create, edit, and delete.

3.2.3.1. A staff member is checked for his identity, with the login page, before being able to access this functionality.

3.2.3.2. A staff member is able to list (shown in Figure 25) and view a solicitation.

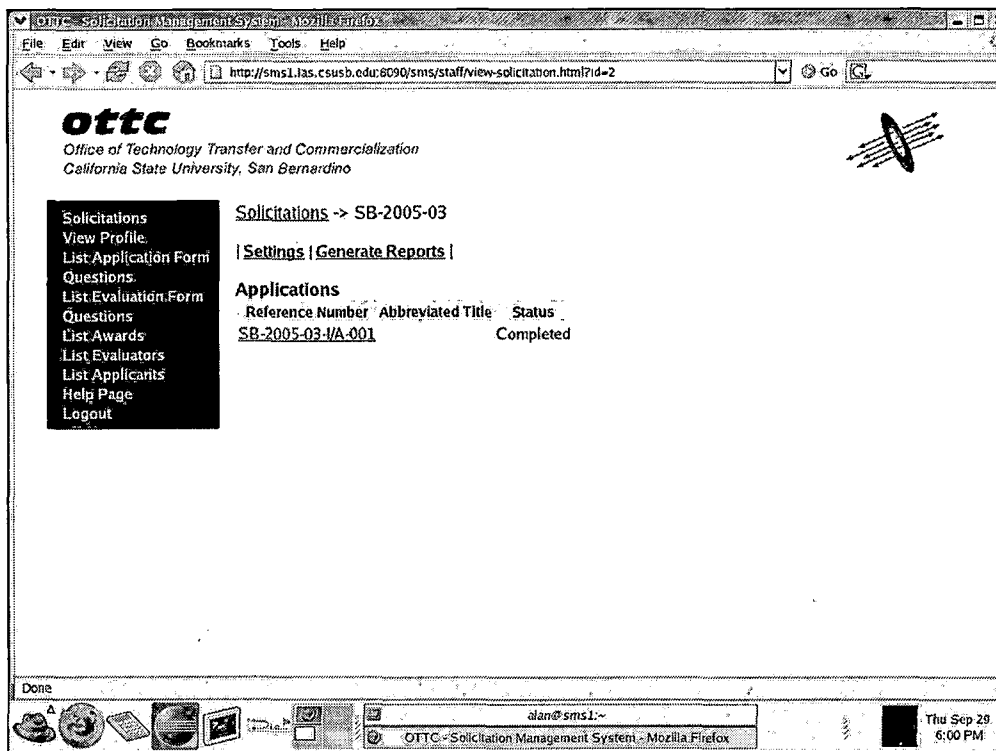


Figure 25. Staff List Solicitations

3.2.3.3. A staff member is able to list and view awards.

3.2.3.4. A staff member is able to list and view applicant related information.

3.2.3.5. A staff member is able to list and view evaluator related information.

3.2.3.6. A staff member is able to list and view applications that an applicant has submitted.

3.2.3.7. A staff member is able to list and view evaluations that an applicant has submitted.

3.2.3.8. A staff member is able to edit his own personal information including his profile and password.

3.2.3.9. A staff member is able to upload a proposal for an applicant.

3.2.3.10. A staff member is able to generate reports for a specific solicitation. This is shown below in figure 26.

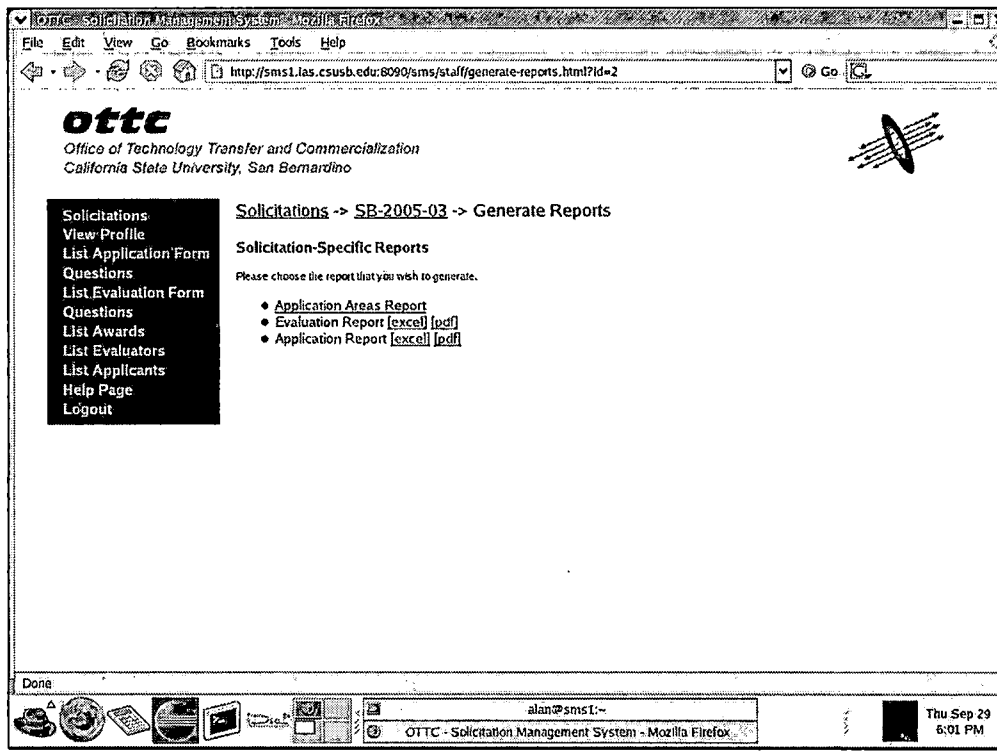


Figure 26. Staff Generate Reports

3.2.3.11. A staff member is able to list, view, create, edit, and delete application groups for a specific solicitation.

#### 3.2.4 User Class: Evaluator

An evaluator can evaluate the applications that are assigned to him.

3.2.4.1. An evaluator is checked for his identity, with the login page, before being able to access this functionality.

3.2.4.2. An evaluator is shown a list of assigned applications (shown in Figure 27).



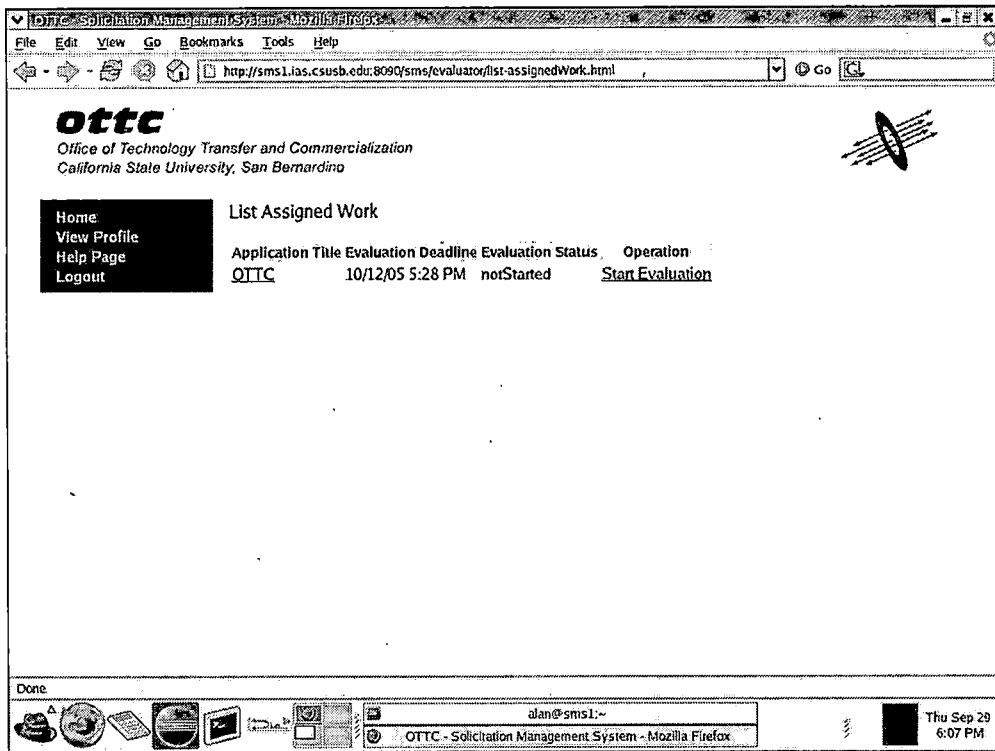


Figure 27. Evaluator List Assigned Work

3.2.4.3. An evaluator is able to read proposals of applications that are assigned to him.

3.2.4.4. An evaluator is able to evaluate applications that are assigned to him (shown in Figure 28).

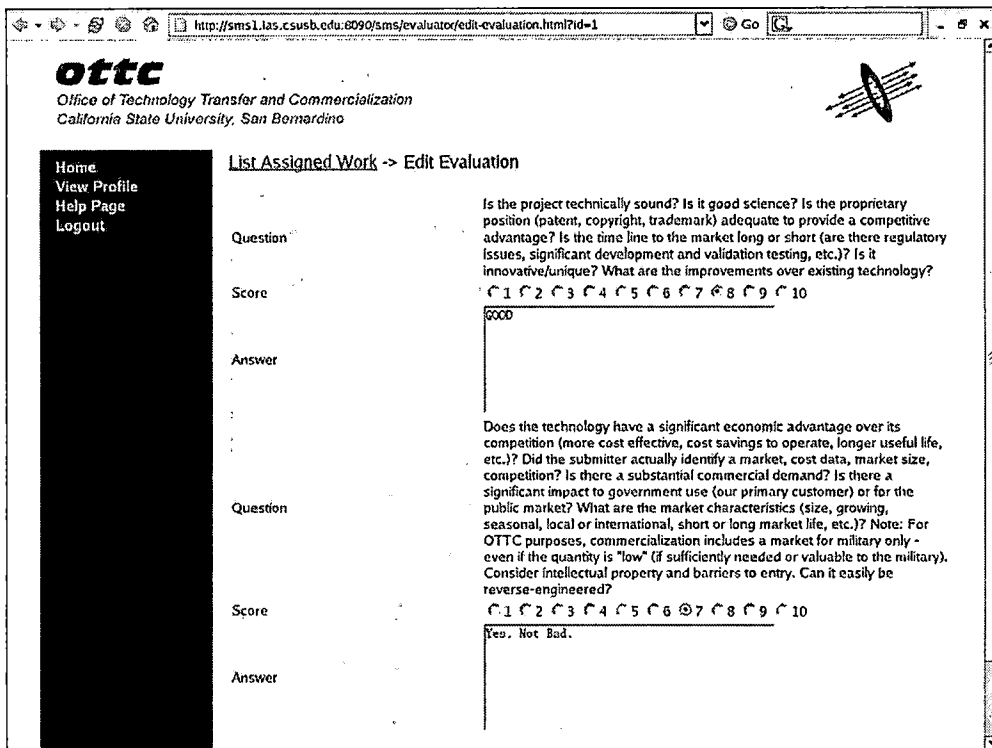


Figure 28. Evaluator Evaluates Application

3.2.4.5. An evaluator is able to view evaluations that he has written (shown in Figure 29).

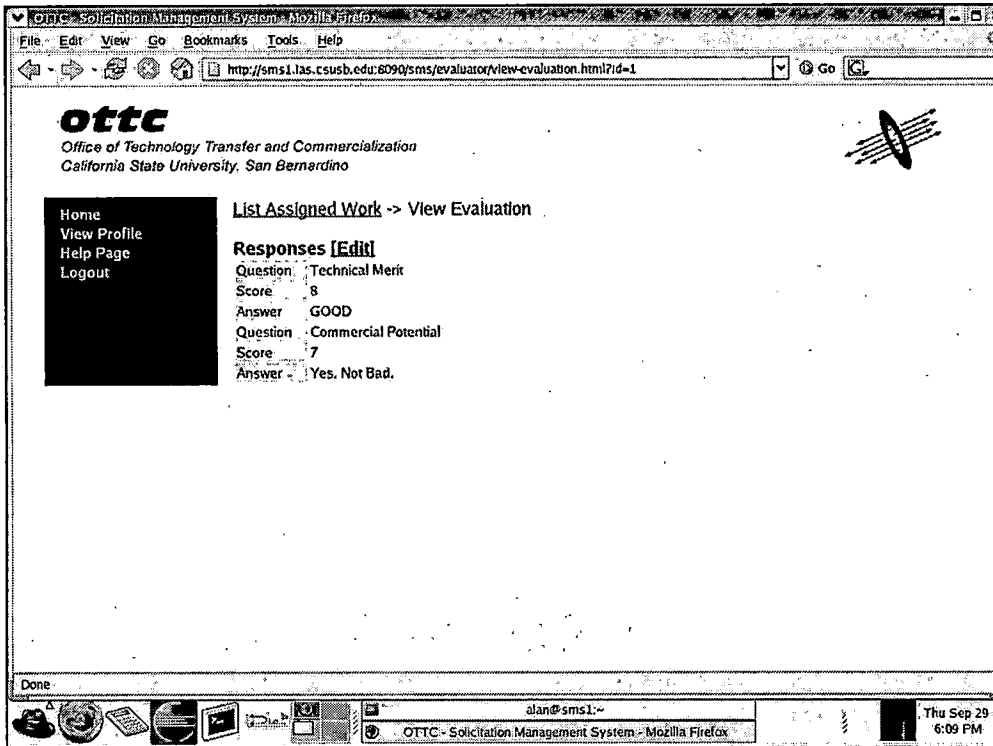


Figure 29. Evaluator Views Evaluation

3.2.4.6. An evaluator is able to edit his own personal information including his profile and password.

### 3.2.5 User Class: Applicant

An applicant can submit applications to open solicitations.

3.2.5.1. An applicant is checked for his identity, with the login page, before being able to access this functionality.

3.2.5.2. An applicant is shown a list of open solicitations as well as a list of his own applications (shown in Figure 30).

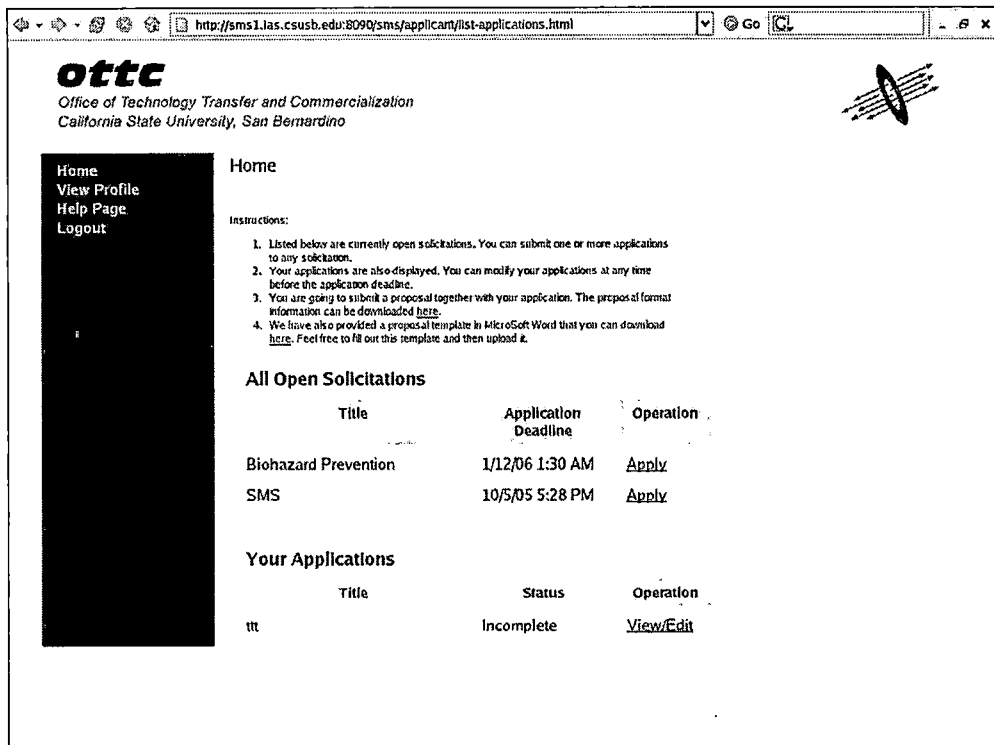


Figure 30. Applicant List Open Solicitations

3.2.5.3. An applicant is able to submit applications to open solicitations. After the applicant has submitted his application, he can also view (shown in Figure 36), edit, and delete his application through a link on his

home page. Figure 31 shows the first page of the application wizard sequence. The applicant is asked to specify his organization on this page.

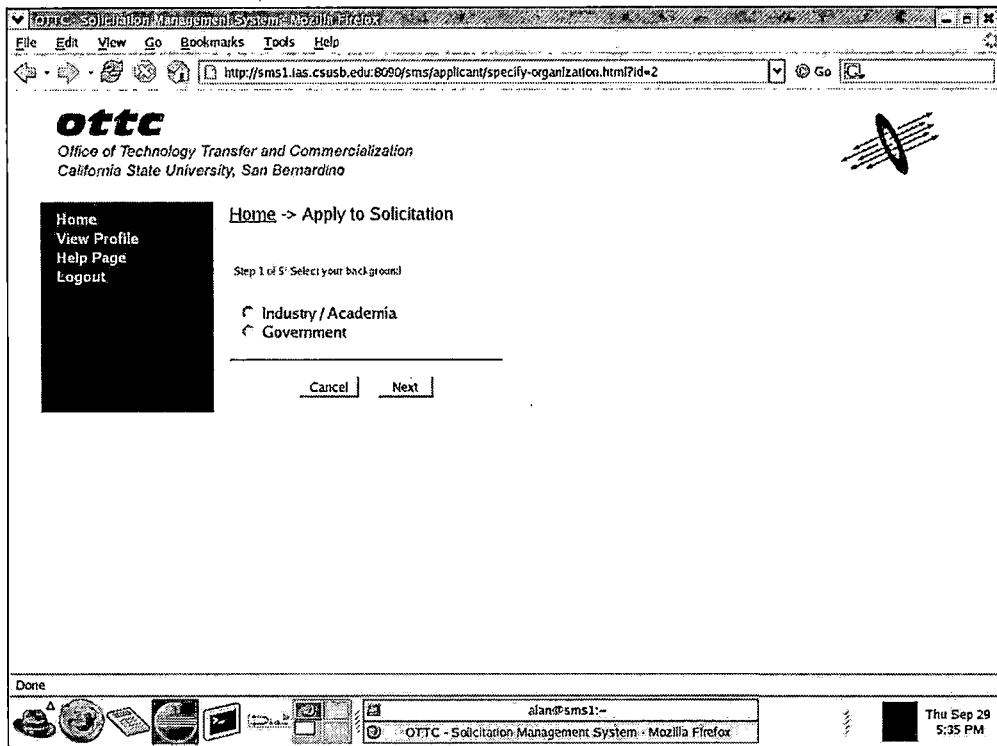


Figure 31. Applicant Create Application Wizard Page 1

Figure 32 shows the second page of the application wizard sequence. The applicant is asked to answer application questions that were assigned to this solicitation.

OTTC  
Office of Technology Transfer and Commercialization  
California State University, San Bernardino

Home -> Apply to Solicitation

Step 2 of 5: Answer the application questions

Application Title	OTTC
First Name	Kent
Last Name	Lin
Phone	9098888888
Abstract of Project Goals	Develop SMS
Proposed Project Duration (maximum 12 months):	12

Previous Cancel Next

Figure 32. Applicant Create Application Wizard, Page 2

Figure 33 shows the third page of the application wizard sequence. The applicant is asked to indicate which award he wishes to apply.

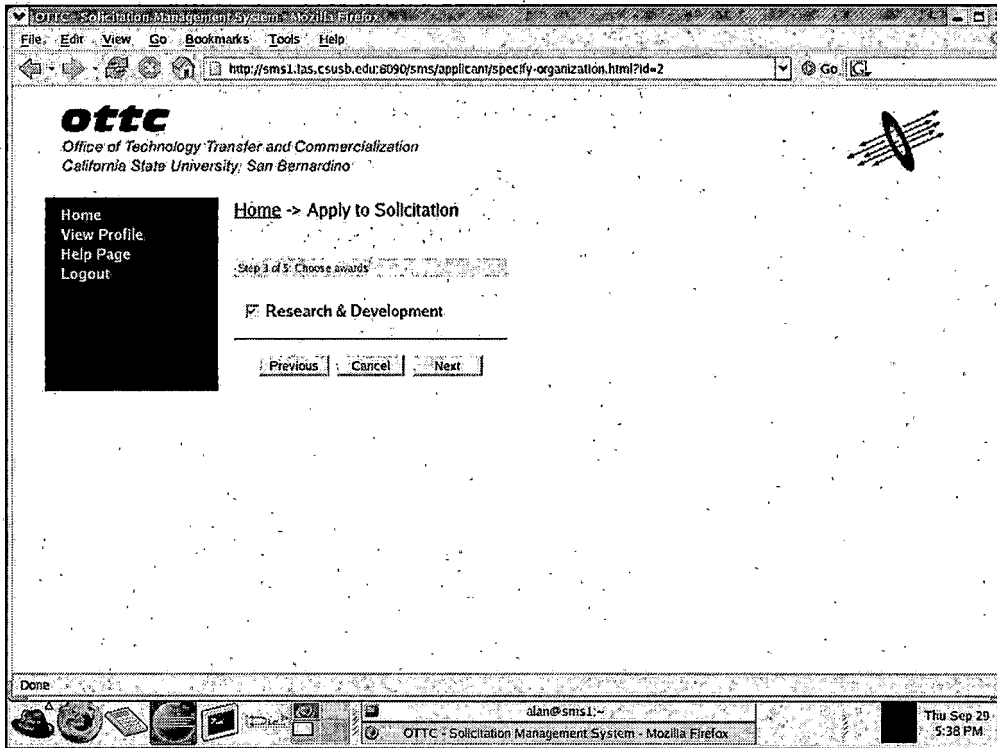


Figure 33. Applicant Create Application Wizard, Page 3

Figure 34 shows the fourth page of the application wizard sequence. The applicant is asked to specify technical areas that are related to his application.

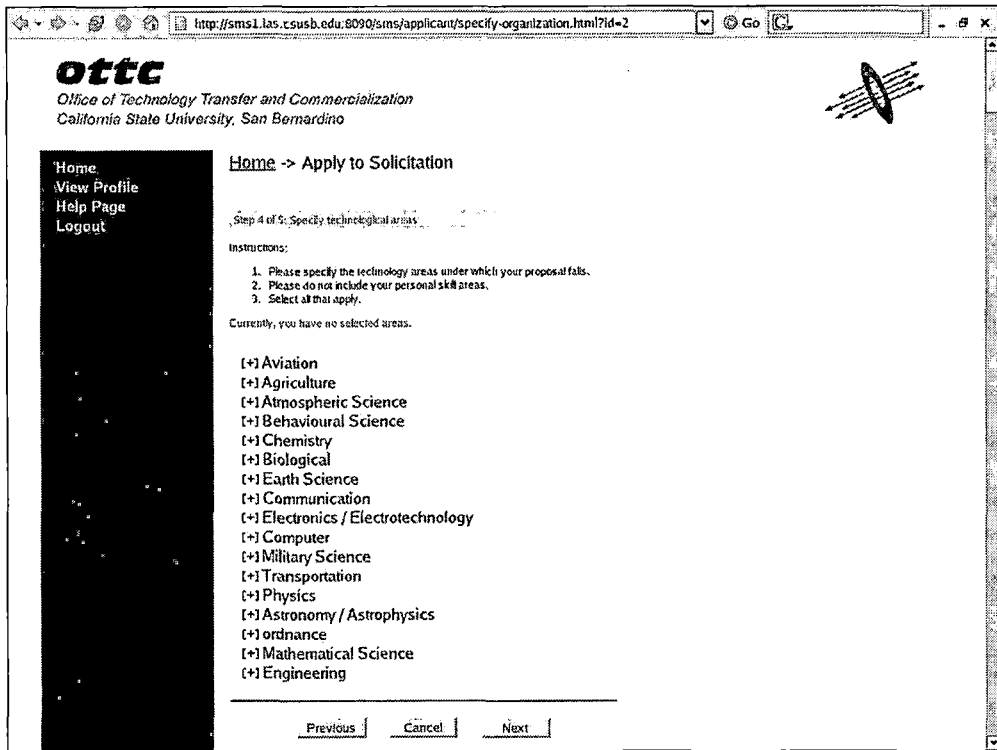


Figure 34. Applicant Create Application Wizard, Page 4

Figure 35 shows the last page of the application wizard sequence. The applicant is notified with a list of instructions of how to finish the application process.



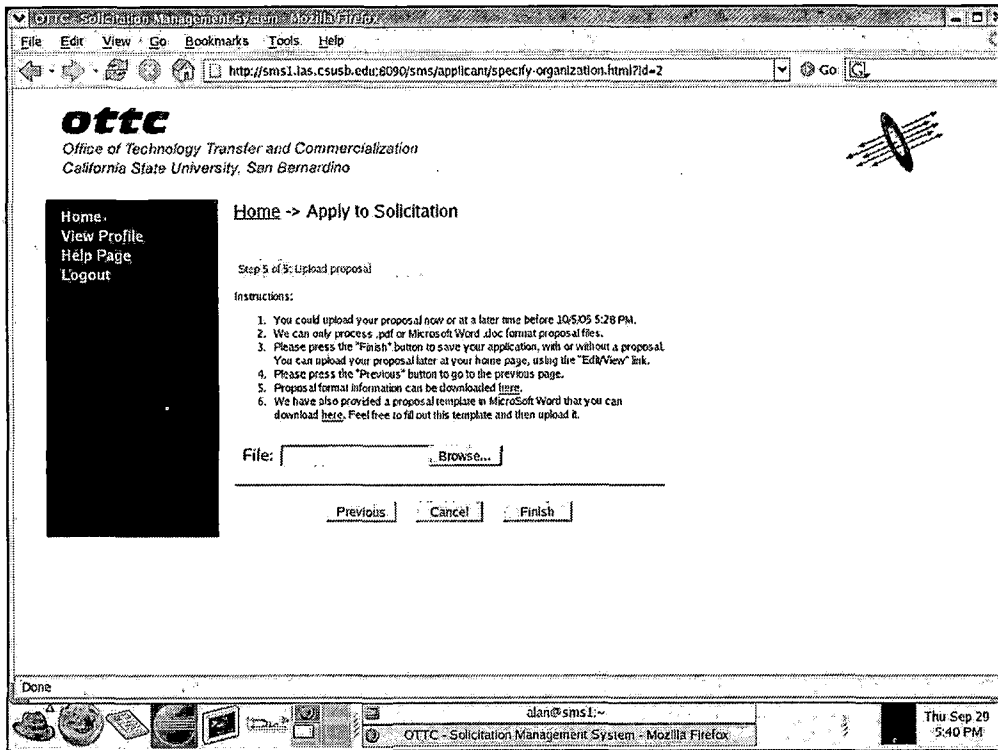


Figure 35. Applicant Create Application, Final Page

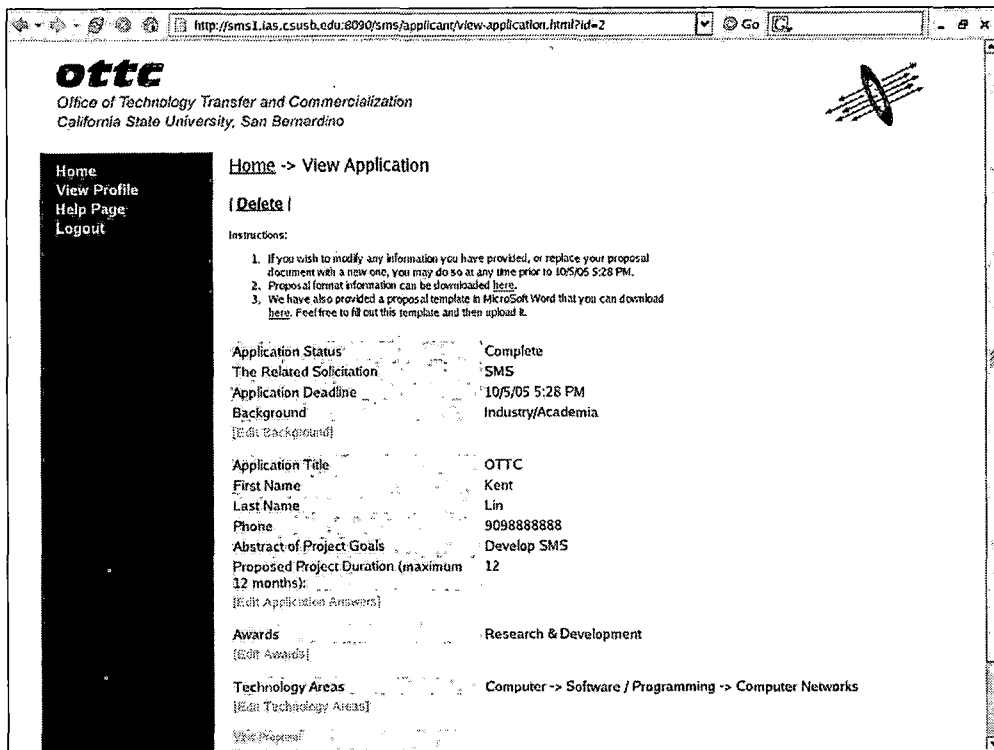


Figure 36. Applicant View Application

3.2.5.4. An applicant is able to edit his own personal information including his profile and password.

### 3.3 Performance Requirements

The loading of most Web pages should take less than 5 seconds. The exceptions to this include generating reports, document conversion, and file uploading.

### 3.4 Local Database Requirement

The local database requirements for this system are the requirements for installing and running the Oracle database. Oracle 9i database requires at least one gigabyte memory.

### 3.5 Design Constraints

The programmer, during its design, should use only currently available computers, as specified in Section 3.1.2, as a server.

The developer should use the latest and/or most mature technology to design the system.

The developer should use JSP (Java Server Pages) to write the view component of the web application.

### 3.6 Software System Attributes

#### 3.6.1 Reliability

The system should be reliable and crashes should not occur.

#### 3.6.2 Availability

The system should be available to agency staff and other potential users who wish to register themselves into the system.

### 3.6.3 Security

This system is designed with security in mind such that a user cannot access information of another user

### 3.6.4 Maintainability

This system is built with most configurations allocated in one place so that it will be easy to make changes and maintain when needed.

### 3.6.5 Portability

This system is built with as less dependencies as possible to gain portability.

CHAPTER FOUR  
DATABASE DESIGN

4.1 Data Analysis

To effectively store all necessary data for each user role, a database with eighteen tables were constructed. Of the eighteen tables, seven are association (join) tables.

The tables used in this project mainly hold text except the proposal table, which holds uploaded files. In the previous version of the Solicitation Management System, the files were uploaded into a file system, whereas in this version, the files are uploaded directly into the database itself.

The ER diagram for this system is shown below in Figure 37.

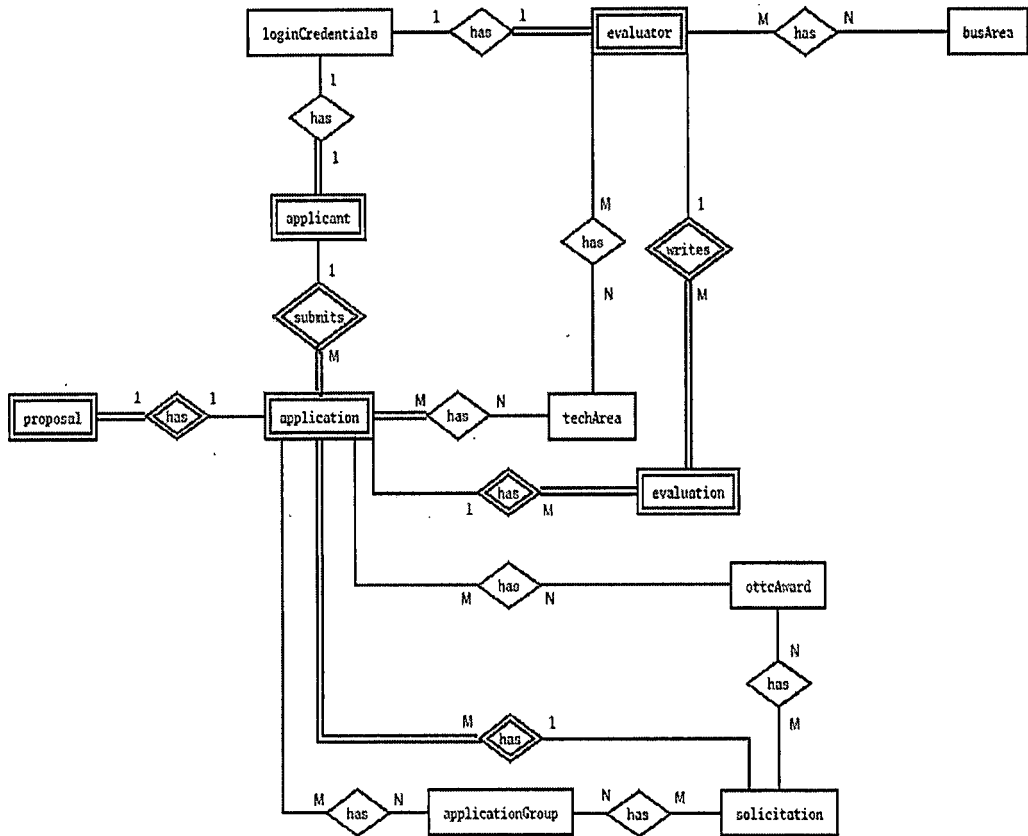


Figure 37. Entity Relationship Diagram

Please note that even though not shown in the ER diagram above, the primary key (id) for the proposal table is the same as the primary key (id) for the application table. Also, the primary key (id) for the user table is the same as the primary key (id) for the applicant and evaluator table depending on whether the user's role is an applicant or evaluator.

As a result, when an application is deleted, the corresponding proposal will be removed as well. This is the same for the users and their corresponding roles. If an evaluator or applicant is deleted, the corresponding user in the user table will be deleted as well.

#### 4.2 Database Schema Logical Model - Relational Schema

The database relational table design is shown below in Figure 38. The primary key for each table is underlined.

Field Names of Table applicant

<u>id</u>	prefix	last_name	first_name
phone1	phone1_ext	phone2	phone2_ext
address1	address2	city	state
zip			

Field Names of Table application

<u>id</u>	title	solicitation_id
applicant_id	submitted	status
reference_name	reference_no	title_abbreviation
the_abstract	qanswer1	qanswer2
qanswer3	qanswer4	qanswer5
qanswer6	qanswer7	qanswer8
qanswer9	qanswer10	qanswer11
qanswer12	qanswer13	qanswer14

Field Names of Table applicationGroup

<u>id</u>	name
-----------	------

Field Names of Table busArea

<u>id</u>	code	title
-----------	------	-------



Field Names of Table evaluation

<u>id</u>	evaluator_id	application_id	status
qanswer1	qscore1	qanswer2	qscore2
qanswer3	qscore3	qanswer4	qscore4
qanswer5	qscore5	qanswer6	qscore6
qanswer7	qscore7	qanswer8	qscore8

Field Names of Table ottcAward

<u>id</u>	award		
-----------	-------	--	--

Field Names of Table proposal

<u>id</u>	mime_type	pdf_file	
-----------	-----------	----------	--

Field Names of Table solicitation

<u>id</u>	title
submission_deadline	evaluation_deadline
solicitation_number	next_application_ref_no
status	

Field Names of Table techArea

<u>id</u>	code	title	
-----------	------	-------	--

Field Names of Table users

<u>id</u>	username	password	password_key
<u>email</u>	role		

Figure 38. Database Relational Schema

RELATION

TABLE 1	TABLE2	TYPE	FOREIGN KEY	REL TB#
evaluator	busArea	M:N		1
evaluator	techArea	M:N		2
solicitation	application	1:M	application	
solicitation	ottcAward	M:N		3
applicant	user	1:1		
application	applicationGroup	M:N		4
application	techArea	M:N		5
evaluator	user	1:1		
application	proposal	1:1		
applicant	application	1:M	application	
evaluator	evaluation	1:M	evaluation	
application	evaluation	1:M	evaluation	

RELATION TABLE

REL TB#	TABLES	FOREIGN KEY	PRIMARY KEY
1	evaluator	evaluator_id	evaluator_id
	busArea	busArea_id	busArea_id
2	evaluator	evaluator_id	evaluator_id
	techArea	techArea_id	techArea_id
3	solicitation	solicitation_id	solicitation_id
	ottcAward	ottc_award_id	ottc_award_id

4	application	application_id	application_id
	applicationGroup	app_group_id	app_group_id
5	application	application_id	application_id
	techArea	techArea_id	techArea_id

Figure 39. Relations Between Tables

### 4.3 Data Type and Details

The tables below shows the name of the field, type, whether it is null or not, what kind of key type (primary or foreign), the default value, and the extra attributes of tables used in this project.

Table 1. Structure of Table App\_AppGroup

Field	Type	Null	Key	Default	Extra
application_id	int(11)		MUL	0	
app_group_id	int(11)		MUL	0	

Table 2. Structure of Table Applicant

Field	Type	Null	Key	Default	Extra
Id	int (11)		PRI	0	
prefix	varchar (255)	YES		NULL	
last_name	varchar (255)	YES		NULL	
First_name	varchar (255)	YES		NULL	
Phone1	varchar (255)	YES		NULL	
Phone1_ext	varchar (255)	YES		NULL	
Phone2	varchar (255)	YES		NULL	
Phone2_ext	varchar (255)	YES		NULL	
address1	varchar (255)	YES		NULL	
address2	varchar (255)	YES		NULL	
City	varchar (255)	YES		NULL	
State	varchar (255)	YES		NULL	
Zip	varchar (255)	YES		NULL	

Table 3. Structure of Table Application

Field	Type	Null	Key	Default	Extra
Id	int(11)		PRI	0	
Title	varchar(255)				
solicitation_id	int(11)		MUL	0	
applicant_id	int(11)		MUL	0	
submitted	int(11)	YES		0	
status	int(11)	YES		NULL	
reference_name	varchar(255)				
reference_no	int(11)	YES		0	
Title_abbreviation	varchar(255)	YES		NULL	
the_abstract	varchar(255)	YES		NULL	
qanswer1	varchar(255)	YES		NULL	
qanswer2	varchar(255)	YES		NULL	
qanswer3	varchar(255)	YES		NULL	
qanswer4	varchar(255)	YES		NULL	
qanswer5	varchar(255)	YES		NULL	
qanswer6	varchar(255)	YES		NULL	
qanswer7	varchar(255)	YES		NULL	
qanswer8	varchar(255)	YES		NULL	
qanswer9	varchar(255)	YES		NULL	
qanswer10	varchar(255)	YES		NULL	
qanswer11	varchar(255)	YES		NULL	
qanswer12	varchar(255)	YES		NULL	

qanswer13	varchar(255)	YES		NULL	
qanswer14	varchar(255)	YES		NULL	

Table 4. Structure of Table ApplicationGroup

Field	Type	Null	Key	Default	Extra
Id	int(11)		PRI	0	
Name	varchar(255)				

Table 5. Structure of Table Application\_OttcAward

Field	Type	Null	Key	Default	Extra
application_id	int(11)		MUL	0	
ottc_award_id	int(11)		MUL	0	



Table 6. Structure of Table Application\_TechArea

Field	Type	Null	Key	Default	Extra
application_id	int(11)		MUL	0	
techArea_id	int(11)		MUL	0	

Table 7. Structure of Table BusArea

Field	Type	Null	Key	Default	Extra
Id	int(11)		PRI	0	
Code	varchar(255)				
Title	varchar(255)				

Table 8. Structure of Table Evaluation

Field	Type	Null	Key	Default	Extra
Id	int(11)		PRI	0	
evaluator_id	int(11)		MUL	0	
application_id	int(11)		MUL	0	
status	varchar(255)				
qanswer1	Text	YES		NULL	
qscore1	int(11)	YES		NULL	
qanswer2	Text	YES		NULL	
qscore2	int(11)	YES		NULL	
qanswer3	Text	YES		NULL	
qscore3	int(11)	YES		NULL	
qanswer4	Text	YES		NULL	
qscore4	int(11)	YES		NULL	
qanswer5	Text	YES		NULL	
qscore5	int(11)	YES		NULL	
qanswer6	Text	YES		NULL	
qscore6	int(11)	YES		NULL	
qanswer7	Text	YES		NULL	
qscore7	int(11)	YES		NULL	
qanswer8	Text	YES		NULL	
qscore8	int(11)	YES		NULL	

Table 9. Structure of Table Evaluator

Field	Type	Null	Key	Default	Extra
Id	int (11)		PRI	0	
prefix	varchar (255)	YES		NULL	
last_name	varchar (255)	YES		NULL	
First_name	varchar (255)	YES		NULL	
Phone1	varchar (255)	YES		NULL	
Phone1_ext	varchar (255)	YES		NULL	
Phone2	varchar (255)	YES		NULL	
Phone2_ext	varchar (255)	YES		NULL	
faculty	int (11)			0	
Staff	int (11)			0	
office_name	varchar (255)				
Attn	varchar (255)			NULL	
address1	varchar (255)	YES		NULL	
address2	varchar (255)	YES		NULL	
City	varchar (255)	YES		NULL	
State	varchar (255)	YES		NULL	
Zip	varchar (255)	YES		NULL	
evaluator_number	varchar (255)			NULL	
Memo	Text			NULL	
status	varchar (255)			active	
max_assigned_evaluations	int (11)			NULL	

Score	varchar(255)			NULL	
isTechnical	int(11)			0	
isBusiness	int(11)			0	

Table 10. Structure of Table Evaluator\_BusArea

Field	Type	Null	Key	Default	Extra
evaluator_id	int(11)		MUL	0	
busArea_id	int(11)		MUL	0	

Table 11. Structure of Table Evaluator\_TechArea

Field	Type	Null	Key	Default	Extra
evaluator_id	int(11)		MUL	0	
techArea_id	int(11)		MUL	0	

Table 12. Structure of Table OttcAward

Field	Type	Null	Key	Default	Extra
Id	int(11)		PRI	0	
Award	varchar(255)				

Table 13. Structure of Table Proposal

Field	Type	Null	Key	Default	Extra
Id	int(11)		PRI	0	
mime_type	varchar(255)	YES			
pdf_file	Longblob	YES		NULL	

Table 14. Structure of Table Sol\_AppGroup

Field	Type	Null	Key	Default	Extra
solicitation_id	int(11)		MUL	0	
app_group_id	int(11)		MUL	0	

Table 15. Structure of Table Solicitation

Field	Type	Null	Key	Default	Extra
Id	int(11)		PRI	0	
Title	varchar(255)				
status	varchar(255)				
submission_deadline	Datetime			0000-00-00 00:00:00	
evaluation_deadline	Datetime			0000-00-00 00:00:00	
solicitation_number	varchar(255)				
next_application_ref_no	int(11)			0	

Table 16. Structure of Table Solicitation\_OttcAward

Field	Type	Null	Key	Default	Extra
solicitation_id	int(11)		MUL	0	
ottc_award_id	int(11)		MUL	0	

Table 17. Structure of Table TechArea

Field	Type	Null	Key	Default	Extra
Id	int(11)		PRI	0	
Code	varchar(255)				
Title	varchar(255)				

Table 18. Structure of Table Users

Field	Type	Null	Key	Default	Extra
Id	int(11)		PRI	0	
username	int(11)		UNI		
password	varchar(255)	YES		NUL	
password_key	varchar(255)	YES		NUL	
Email	varchar(255)				
Role	varchar(255)				



## CHAPTER FIVE

### SYSTEM ARCHITECTURE

The Solicitation Management System has a system architecture based on Spring MVC framework and the Hibernate object-relational database mapping framework.

#### 5.1 Database Mappings

The database mappings used in this project are use to represent the following relationships between objects: one-to-one, many-to-many, one-to-many, and many-to-one.

##### 5.1.1 One-to-One Mapping

There are three places in this system that uses this kind of mapping. The first place is when the applicant is mapped to the user table. The second place is when the evaluator is mapped to the user. The mapping is necessary in these two places because the id field in the applicant table and the evaluator table has to be identical to their users respectively. The third place that uses one-to-one mapping is when the proposal is mapped to the application. In this scenario, the prosal id had to match the application id.

For each of the tables that are involved in a one-to-one mapping, the field in the corresponding .java file has

been set as well as the setter and getter methods for that field.

### 5.1.2 Many-to-Many Mapping

Many-to-many mapping are used in several places in this system. They are:

1. Application to techArea, ottcAward, and appGroup tables.
2. Evaluator to techArea and busArea tables.
3. Solicitation to ottcAward and appGroup tables.

For each of the tables that are involved in a many-to-many mapping, the field in the corresponding .java file has been set as well as the setter and getter methods for that field.

### 5.1.3 Many-to-One and One-to-Many Mapping

Many-to-one and one-to-many mappings are also used in this system. Their usage is listed below:

1. Application to evaluation table.
2. Evaluator to evaluation table.
3. Solicitation to application table.
4. Applicant to application table.

For each of the tables that are involved in a many-to-one and one-to-many mapping, the field in the corresponding .java file has been set as well as the setter and getter methods for that field.

## 5.2 SMS Controllers

There are four kinds of controllers from the Spring framework used in this project: Controller, ParameterizableViewController, SimpleFormController, and AbstractWizardFormController.

### 5.2.1 The Controller Class

Controller is a class under the Spring Framework. In SMS, the Controller interface is directly implemented to support user requests to generate reports. It serves as a mean to control and render the view for the excel dumps created by the Solicitation Management System.

### 5.2.2 The ParameterizableViewController Class

The ParameterizableViewController is a class under Spring Framework. In this system, this controller is mostly used for the view and list function components.

### 5.2.3 AbstractWizardFormController

The AbstractWizardFormController is a class under Spring Framework. In this system, this controller is used in two places. They are the create solicitation for the officer role and the create application for the applicant role. Using the AbstractWizardFormController lets the system create complete application and solicitations. If the user does not follow through the sequence to the end

where they press the finish button, the operation they have just done will not be saved.

#### 5.2.4 SimpleFormController

The SimpleFormController is a class under Spring Framework. In this system, this controller is the most widely used controller.

## CHAPTER SIX

### SECURITY RELATED ISSUES

As with other online applications available nowadays, security is also an important issue for the Solicitation Management System. This chapter will describe related issues of how the security of the SMS is done and how it is enforced.

#### 6.1 Security Focus

The main focus for security of the SMS is to keep the information related to the user account intact from uninvited access. This would include protecting the user password and blocking information access from unnecessary personnel.

##### 6.1.1 Protecting User Password

The mechanism for protecting the user password for the SMS is to use a Password Hasher to hash the password that the user inputs. The SMS then saves the hashed password into the database. Thus even if a person can access the database directly, he cannot retrieve the original user password, all he can get from the database is the hashed password.

### 6.1.2 Blocking Unnecessary Access

The unnecessary access mentioned here includes:

1. An evaluator cannot access(read) proposals that are not assigned to him.
2. An evaluator cannot edit, read, or write another evaluator's evaluation.
3. An applicant cannot edit, delete, read, or write another applicant's application.
4. An applicant can only apply to an open solicitation.
5. An applicant cannot edit, delet, read, or write an evaluation.

The first four restrictions are enforced by the "AuthorizationInterceptor." To bar an evaluator from reading proposals that are not assigned to him, the system compares the requested proposal to all of the assigned proposals. When a match is not found, the system will redirect the user back to the evaluator homepage.

To restrict an evalutator from editing, reading, or writing an evaluation for a proposal that is not assigned to him, the system compares the requested evaluation to a pool of assigned evaluations. When no match occurs, the system will deny the user and redirect him to the evaluator homepage.

To guarantee that an applicant cannot edit, delete, read, or write another applicant's application, the system compares the requested application to collection of his own applications. If an exact match cannot be found, the system will refuse to render the requested page and redirects the applicant back to the applicant homepage.

To ensure that an applicant only applies to open solicitations, the system compares the requested solicitation to a group of open solicitations. The applicant can only gain access to the application page when a match is found. If not, the applicant will be redirected back to the applicant homepage.

The fifth restriction is enforced by the "RoleAuthorizationInterceptor." To impose the regulation that the applicant cannot access the evaluator resource, the system compares the servlet path with the user role. If an applicant tries to access a page that does not belong to his user role, he will be redirected back to the applicant home page.

### 6.1.3 Login

The "LoginController" controls the login procedure. The logging controller first checks the session to see if an user role has already been set. If so, then when the user tries to access the login page, he will be redirected

back to his user homepage. If not, the system will set the user name, user id, and user role to the session and redirect the user to the corresponding homepage.

During this process, the "LoginCredentials" along with its "LoginCredentialsValidator" is called to respond with proper behavior to various kinds of login possibilities. This includes granting access to the correct set of user name/ password and throws the corresponding error message when an error occurs.

The sequence diagram of the login function is shown below. document.

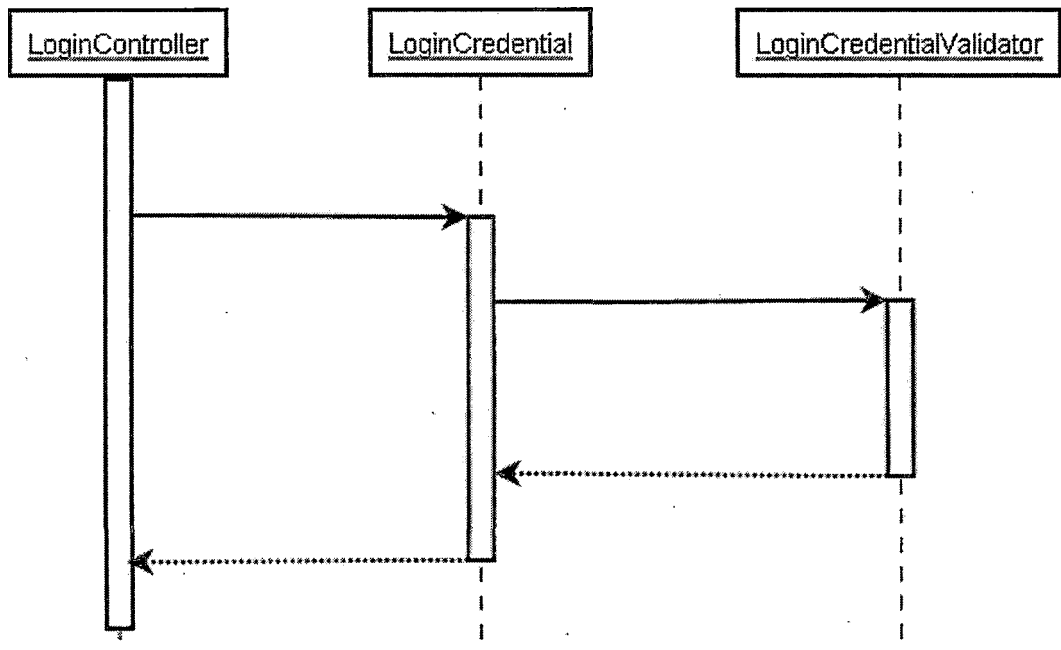


Figure 40. Login Sequence Diagram



## CHAPTER SEVEN

### MAINTENANCE MANUAL

Software products developed nowadays go through several phases of development. One of the most important phases a software product must go through after the software is in production is the maintenance phase. The maintenance phase of the software life cycle ensures that the software operates as expected. In such cases that errors might occur, it is in the maintenance phase that developers respond to the correct problem.

In this chapter, the system requirements along with the installation of this system are explained. Also in this chapter, the guide of how to change the environment for this system is also shown.

#### 7.1 Prerequisite Software Installation

In order to install the Solicitation Management System, several prerequisites are needed. The installation of these required software will ensure that the system will be able to operate as expected. The prerequisite softwares for the main program body include Linux operating system, Java Runtime Environment, Oracle database, Tomcat. For the document conversion server, the prerequisites needed are Microsoft Windows XP, Microsoft

Office 2003, and the Adobe Universal PostScript Windows Driver.

### 7.1.1 Linux Operating System

The preferred Linux Operating System is the RedHat Enterprise Linux. The manual for installing this system is different for different processor types. A list of manuals for an assortment of processors can be found on the RedHat website at <http://www.redhat.com/docs/manuals/enterprise/>.

### 7.1.2 Java Runtime Environment

The Java Runtime Environment (JRE) provides the Java Environment in which this system needs to run. The JRE can be downloaded from Sun's website. The installation instructions for the JRE also vary among different machines. A list of installation instructions can be found on Sun's website at <http://java.sun.com/j2se/1.5.0/jre/install.html>.

### 7.1.3 Oracle Database

The version of Oracle Database that this system is currently using is 9.2.0.4. The installation process involves several steps. A documentation is available on how to install Oracle 9i at <http://www.tldp.org/HOWTO/Oracle-9i-RH8-and-RH9-HOWTO-3.html>.

#### 7.1.4 Tomcat

The version of Tomcat that this system is currently using is Tomcat 5.0.28. This version can be downloaded from the apache website at <http://tomcat.apache.org/download-55.cgi#5.0.28>. The instructions for installing and setting up Tomcat can also be found on the apache website at <http://tomcat.apache.org/tomcat-5.0-doc/index.html>.

#### 7.1.5 Microsoft Windows XP Professional

The document conversion server is run on top of the Microsoft Windows XP Professional Platform. The Microsoft Windows XP Professional is a commercial software that can be purchased from various suppliers. A list of how to purchase a licensed copy of Microsoft Windows XP Professional is shown on Microsoft's website at <http://www.microsoft.com/windowsxp/pro/howtobuy/default.mspx>.

Installation of Microsoft Windows XP is fairly easy due to the user friendly interface. The user should put the disc into the optical drive upon boot and a display window will appear with instructions. The instructions are sufficient to guide the user through the installation process. Please note that at one point the product key will be prompted so have that ready upon installation.

### 7.1.6 Microsoft Office 2003

The Microsoft Office 2003 is an essential part of the document conversion server. A licensed copy can be purchased from various suppliers as well. A list of ways to purchase a licensed copy of Microsoft Office 2003 is lists on Microsoft's website at <http://www.microsoft.com/products/info/render.aspx?view=22&type=mnp&content=22/reseller>.

The installation of Microsoft Office 2003 is fairly easy too due to the user friendly interfaces. Just follow the instructions on the dialogue box and press "next" until you see the finish button. During the installation process, the user will be prompted to enter the product key that was provided with the software.

### 7.1.7 Adobe Universal PostScript Driver

The Adobe Universal PostScript Windows Driver can be downloaded from Adobe's website at <http://www.adobe.com/support/downloads/product.jsp?product=44&platform=Windows>. After a version of the Adobe Universal PostScript Windows Driver is obtained, just execute the exe file and follow the instructions in the dialogue. When the user is prompted for the filename, enter "output.ps" and assign "sms" as the printer name.

## 7.2 Installation of This System

The installation of the Solicitation Management System consists of several steps. First create the folders "file" and "temp" with the permission 777 under your chosen directory. Then copy sms.war along with log4j.properties to the same directory. Finally, copy sms.xml to the localhost directory under tomcat/conf/Catalina/localhost and make modifications to accommodate the changes to the paths.

## 7.3 Changing the System

From the user's end, the Solicitation Management System cannot be changed. However, changes can be made easily by the developer by modifying the configuration files. The configuration files are stored in the resource folder of the project. They are files that end with extensions .xml and .properties. Please take note that modification of the system configuration files might result in breaking the system.

## CHAPTER EIGHT

### CONCLUSION AND FUTURE DIRECTIONS

#### 8.1 Conclusion

The Solicitation Management System facilitates the processing of solicitations for the Office of Technology Transfer and Commercialization (OTTC). Based upon the feedback received from the OTTC, the new version of the system represents an improvement over the previous version. By using the newer technologies of Spring Framework and Hibernate, the system runs more efficiently and is easier to maintain.

With the Solicitation Management System, the whole solicitation process from the creation of a solicitation to the evaluation of applications can be done online. This can speed up the solicitation process from several aspects.

First, because everything is submitted online, there is no need to wait for the paperwork to be physically sent through the post office. This is a benefit because mail can be delayed during the process or even lost.

Second, all the information required for the solicitation is typed into the system through various interfaces. As a result, OTTC staff do not need to transcribe into the computer the information that the

users write by hand, which is a significant time saving feature.

Finally, the system matches evaluators to specific applications based on technology areas. Thus, the officers can more easily assign evaluators to applications, so the areas of expertise of the evaluators match those of the application. This is a feature that saves the most time in the solicitation process.

With the time saving and user friendly features that this system provides, the officers can process their solicitation on time with lesser effort.

## 8.2 Future Directions

The future direction for the Solicitation Management System is to expand it to a more general purpose system. A more general purpose system will be more flexible and can be easily adapted to other possible usage scenarios.

## REFERENCES

- [1] Apache HTTP Server Version 2.0 Documentation  
The Apache Software Foundation 1996  
(<http://httpd.apache.org/docs-2.0/>)
- [2] F. Ortin, B. Lopez and J. Baltasar Garcia Perez-Schofield, "Separating Adaptable Persistence Attributes through Computational Reflection," *IEEE Software*, November 2004, pp. 41-49.
- [3] Hibernate Framework  
Hibernate 2005  
(<http://www.hibernate.org/>)
- [4] IEEE Std 830-1998 (Revision of IEEE Std 830-1993),  
IEEE Recommended Practice for Software Requirements Specifications.
- [5] J. Arthur and S. Azadegan, "Spring Framework for Rapid Open Source J2EE Web Application Development: A Case Study," *snpd-sawn*, pp. 90-95, Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD/SAWN'05), 2005.
- [6] Java Servlet Technology  
Sun Microsystems, Inc 2006  
(<http://java.sun.com/products/servlet/>)



- [7] JDBC Data Access API  
Sun Microsystems, Inc 2006  
(<http://java.sun.com/products/jdbc/>)
- [8] JDBC Overview  
Sun Microsystems, Inc 2006  
(<http://java.sun.com/products/jdbc/overview.html>)
- [9] K. Laufer, "A Hike through Post-EJB J2EE Web Application Architecture," Computing in Science and Engineering, September 2005, pp. 80-88.
- [10] MySQL Homepage  
MySQL AB 2006  
(<http://www.mysql.com/>)
- [11] R. Johnson, "J2EE Development Frameworks," Computer, vol. 38, no. 1, pp. 107-110, January, 2005.
- [12] Spring Framework  
Spring Framework 2005  
(<http://www.springframework.org/>)
- [13] The Apache Jakarta Tomcat 5.0 Servlet/JSP Container  
The Apache Software Foundation 2006  
(<http://jakarta.apache.org/tomcat/>)
- [14] Webopedia: Online Computer Dictionary for Computer and Internet Terms and Definitions.  
Jupitermedia Corporation 2006  
(<http://www.webopedia.com/>)