An Online Scholarship Application System

A Manuscript

Submitted to

the Department of Computer Science

and the Faculty of the

University of Wisconsin-La Crosse

La Crosse, Wisconsin

by

Wen-Kai Shen

in Partial Fulfillment of the

Requirements for the Degree of

Master of Software Engineering

January, 2011

An Online Scholarship Application System

By Wen-Kai Shen

We recommend acceptance of this manuscript in partial fulfillment of this candidate's requirements for the degree of Master of Software Engineering in Computer Science. The candidate has completed the oral examination requirement of the capstone project for the degree.

_____ _____

Dr. Kasi Periyasamy Date
Examination Committee Chairperson

_____ _____

Dr. Thomas Gendreau Date
Examination Committee Member

_____ _____

Dr. Mark Headington Date
Examination Committee Member

ABSTRACT

Wen-Kai, Shen, “An Online Scholarship Application System”, Master of Software Engineering, January 2011, Advisors: Dr. Kasi Periyasamy.

The Foundation Office at the University of Wisconsin-La Crosse deals with a large number of scholarship applications typically in the range of five to eight thousands every academic year. Nevertheless, processing these applications is time-consuming due to their current paper-based processing. At present, applicants have to fill out their application forms and submit them manually to the office. If there is any problem with their applications while they are processed, it will also take extra time for both the reviewing committee as well as the applicant to communicate and correct the errors. As a result, additional paperwork for review may cause a delay in the entire procedure. Some basic criteria such as GPA are now examined tediously by committee members but it can be checked by a computerized system automatically. Furthermore, maintaining and keeping relevant information and updates on the status of an application for each applicant are strenuous without computerized manipulation. In order to solve these problems, a pilot project was done to automate many of the tasks performed by all the parties including the applicant and the scholarship committees.

This manuscript describes the development of a new software system for scholarship application and processing by re-engineering the pilot project. Some of the important features of this new system include efficient processing and interactivity between multiple users under a highly secure networking environment. The document

also describes some of the challenges encountered during each phase of the development life cycle, while adding more functionalities and techniques introduced for solving the issues mentioned earlier.

ACKNOWLEDGEMENTS

Above all, I am grateful to my project advisor Dr. Kasi Periyasamy, who has given me his valuable suggestions, superb guidance and essential resources. Besides, I would like to thank the project sponsor, UW-L Foundation Office, especially, to Ms. Sara Olson, who spend a lot of time on discussion of the progress during the tenure of this project, and then support my data collection by communicating with other departments.

Last but not least, special thanks are due to Information Technology Services (ITS) at UW-La Crosse for the assistance in furnishing the database access to student information system (SIS).

With all of your encouragement and commitment, the project is able to develop smoothly and be accomplished successfully.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

GLOSSARY

ACID properties

They consist of four features – Atomicity, Consistency, Isolation and Durability. By supporting these properties, it can be guaranteed that each transaction between a system and a database is processed with reliability.

Ajax

Ajax is the abbreviation of Asynchronous JavaScript and XML. This term is involved in web-based developmental technologies which provides interactive communication between client and server without impeding the current data presentation of the page.

Authentication

It is the act that refers to any mechanism that decides authenticity. In a secured system, there may be one or more identifications such ID, Username and Email that are used to perform this act.

Authorization

It is the function of granting access rights to assets, after the system has verified authenticity. Based on the access policy of the system, users of different roles would be authorized to access some specific data or devices.

FAFSA

FAFSA is the abbreviation of Free Application for Federal Student Aid. This is a form which consists of several questions in the different categories prepared annually by current and prospective college students in the United States to determine their

eligibility for student financial aid.

JasperReports

It is an open source java reporting library to extract data from any kind of data source and produce pixel-perfect documents that can be viewed, printed or exported in a variety of document formats including HTML, PDF, Excel, OpenOffice and Word.

Realm

A realm is the mechanism that identifies authorized users. This is defined within Tomcat web server to manage security.

Stress Testing

It refers to tests that determine how robust the software can be, when it approaches extreme values around the limits of normal operation. In general, stress tests are focused on robustness, availability, and error handling under an enormous load, other than correct response and processing under normal circumstances.

Tomcat Resource Factory

It is a class of Apache Tomcat API to create the resource by using the information described in /WEB-INF/web.xml.

ZK Framework

ZK is an open source Ajax Web application framework, written in Java, which enables creation of rich graphical user interfaces for Web-based applications with no Java Script. It is also a component-based framework, which is driven through events.

1 Introduction

At the University of Wisconsin-La Crosse, both freshmen and continuing students can apply to a large number of scholarships. They need to apply for scholarships explicitly illustrating their qualifications and eligibility for the scholarships they apply. There are various criteria to satisfy when a student applies for a scholarship. The majority of criteria are based on students' merits and other academic performances. Their financial information in FAFSA is another main consideration for how much scholarship they can receive. Some scholarships may also require that applicants have to take specific courses or select a particular major. The UW-L Foundation Office plays the role as a sponsor in maintaining scholarships information, processing applications and eventually either awarding or rejecting the scholarship applications.

1.1 Current Procedure of Scholarship Application at UW-L

Currently, the procedures of applying for scholarships, managing scholarship and evaluating application forms are all done manually using paper-based processing. Figure 1 shows the operations of the entire scholarship application process including the roles and responsibilities of various parties involved in the process.

The number in front of each entity represents the sequence of the operation. There are totally 12 steps in the whole process which start from creating scholarships and end in making the final decision with awarding the scholarship or rejecting an application. Besides, students can enquire about the progress of their applications only through email or phone. Each scholarship will be reviewed by a committee consisting

of mostly the faculty members of the university but may also include outside members. A member of each committee is chosen as the chair of the committee who makes the final decision on behalf of the committee to rank the applicants based on the evaluations and comments by other members of the same committee.

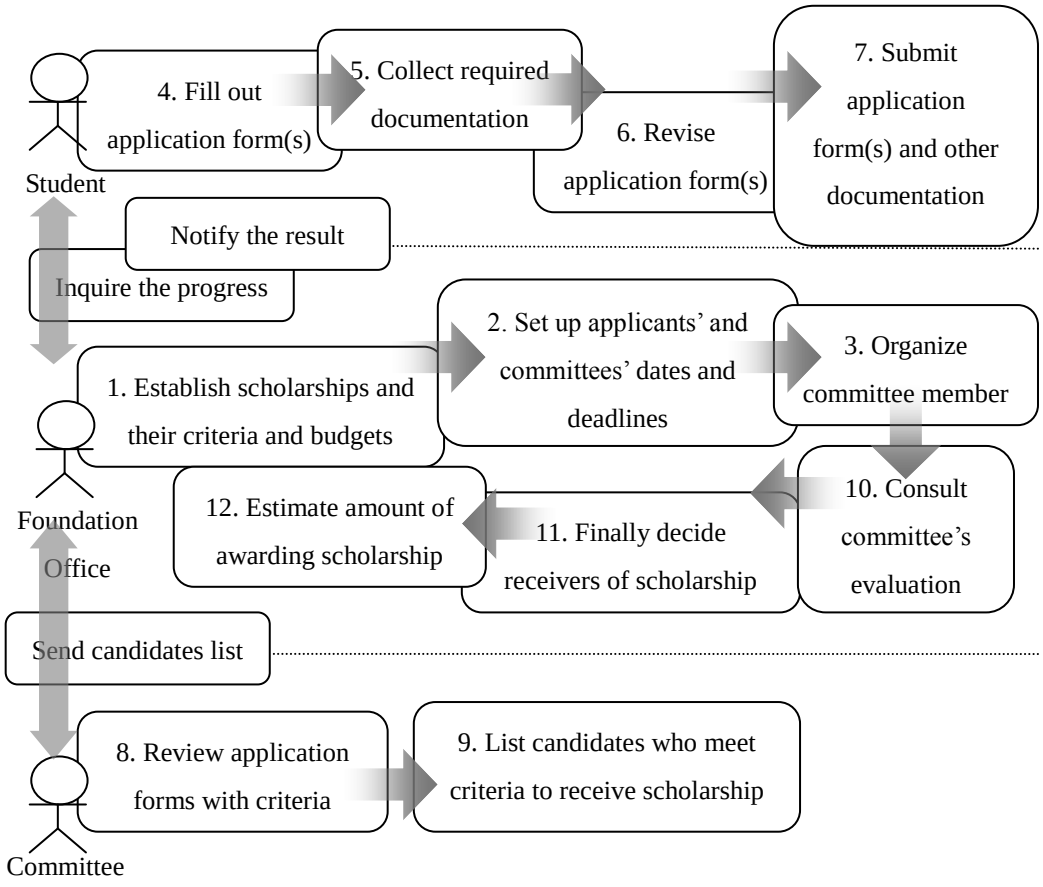


Figure 1: Current Operation of Scholarship Application

The communication between various parties (applicant, committee members, committee chairs and the Foundation office) involves a lot of paper work. Sometimes, an error in the application or the evaluation report may cause the paper to be sent back to the previous processing step which may further delay the processing of the application.

1.2 Objectives of this project

Apparently, there are some disadvantages in the current paper-based processing. The following is a list of these problems:

1. Inconvenience and time-consuming communication
2. Paper and ink consumption
3. Heavy workload
4. Difficulty in data maintenance and search
5. Deficiency of timely notification for application deadline

The main purpose of this project is to develop an online web-based system which can facilitate the processes of various scholarship applications. In addition, the system can also provide different functionalities for different kinds of users including students, guidance directors, committee members and administrators. This project emphasizes the convenience for these users who participate in the scholarship application procedure to access this web-based system. In addition, the workload for the UW-L Foundation Office can be effectively and considerably reduced.

In order to computerize each entity shown in Figure 1, a friendly user interface is required for all different kinds of users, which enables the users to perform the following:

- Login and logout mechanism for all users.
- Changing passwords and editing their accounts.
- Registration for new students.
- Management of scholarship information and their criteria.
- Creating and processing of scholarship applications by students.

- Organizing committees and their members.
- Facilitating guidance directors at various schools to provide necessary documents such as transcripts and reference letters.
- Appointing committees for reviewing scholarships.
- Automatically checking various deadlines, scheduling, and email notifications.
- Automating the checking of some of the criteria for the scholarships such as GPA and ranking.
- Uploading and downloading of various documents such as transcripts, references letters and supporting materials for scholarships.

All functionalities in the system are categorized based on users' roles. Because several types of users can access the system through the Internet, security must be assured to protect confidential information including application form, contact data, students' scores and transcripts.

2 Requirement and Analysis

An earlier version of the online system for scholarship application was developed in the previous year which made it easier to understand and extract the functionalities of the new system. Since the previous version was not just a prototype, the project sponsors decided to revamp the product and add new functionalities. In addition, the previous version was developed using the .NET technology which made it dependent on Microsoft Technologies. However, the sponsors decided to make it portable and wanted to use open source technology for the newer version.

2.1 Re-engineering Feasibility Investigation

A re-engineering process was applied to the previous version of the online system which gave rise to the following objectives:

- Enhancement of already implemented functionalities.
- Re-structuring functional units to provide a better application process.
- Improving security.
- Improving the look and feel of the GUI.
- Shortening response time between service requests.

Besides, there are some more new requirements added. For example, adding one more new role such as the guidance director in schools. Figure 2 depicts the re-engineering process applied to the previous system.

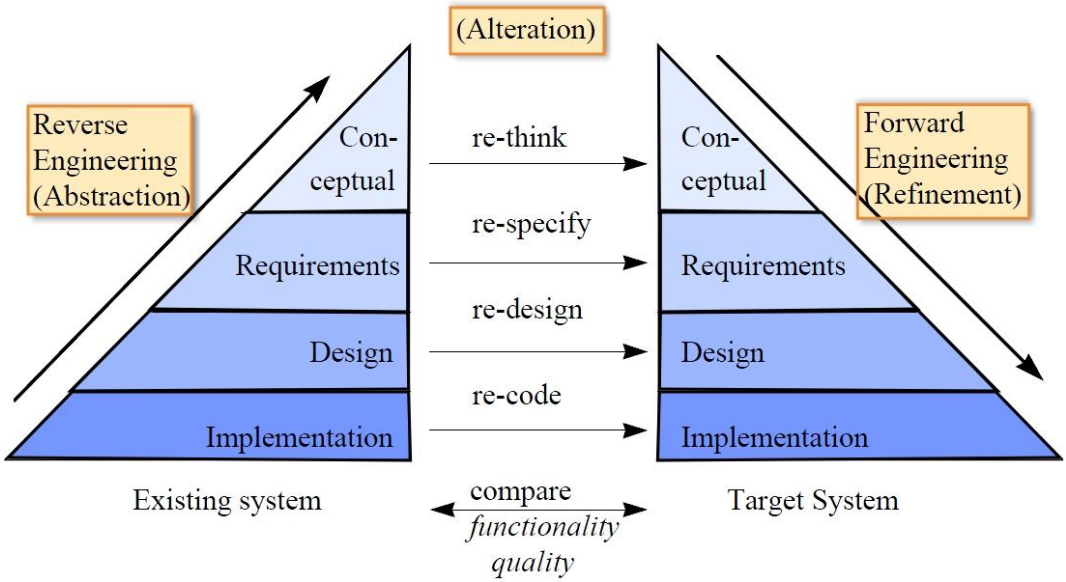


Figure 2: General Model for Software Re-engineering

Derived from each phase in the development life cycle, this model embodies a complete processing through all layers of re-engineering based on the levels of abstraction in the existing system. By means of alteration between the target (new) and existing system, each layer can be represented in the target system. Unfortunately, in the implementation phase, source code from the existing system is not accessible; that is, the alteration (re-coding) could not be done.

2.2 Re-engineering Approach

Basically, through reading the existing documentation, most valuable facts can be collected from the old system development. The original architecture and platform were not adopted for the new system development and re-coding could not be conducted. Instead, the re-engineering approach shown in Figure 3 named Evolutionary Re-engineering Approach was used to identify all critical functionalities

from the existing system. This evolution focuses on replacement in which portions of the existing system is substituted for re-engineered system portions.

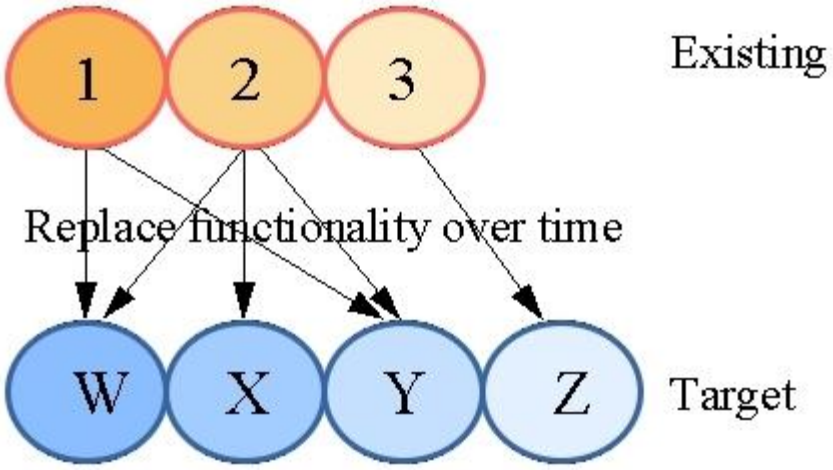


Figure 3: Evolutionary Re-engineering Approach

2.3 Potential Risks

It is worth mentioning that the previous version was designed for dealing with continuing students' applications. But the new system was required to do the processing of freshmen scholarships. Based on this underlying requirement, some functionality from the existing system could not be used individually by users, but could be integrated into another function in the target system. For processing freshmen applications, there are also additional data which must be supported by not only Information Technology Services in UW-L but also students' guidance directors from their high schools. For example, the typical data to assess students' academic performance is their high school transcripts which are not required for continuing students. As a result, after interviewing the sponsor, there are some risks identified as follows:

Risk	Probability	Impact	Reason
Misunderstanding from the current scholarship application procedure and rule	0.5	High	All functionality may not suitable for the current paper-based application processing, because of the developer's unfamiliarity in its regulation.
Interface is replaced with new look and feel and operation procedure	0.8	Low	Functional portions gathered from the previous system are re-engineered instead of architectural portions.
Users access through insecure connection via Internet	0.1	High	Personal information may leak and be tampered with.
Some data retrieved from another external system or database	0.3	Middle	When external resources are not available anytime, the system usage is limited because of its dependency.

Table 1. Evaluation of Potential Risks

The probabilities shown in the table above are rough estimates based on the situations described by the sponsor and do not actually reflect any real incidents. To alleviate these risks when they happen, the following measures were used in the current development process:

- A simple and easy-to-navigate user interface was developed and shown to the sponsor, before implementing system functionalities. By this way, the look and feel was ensured to be satisfactory and the operation procedures were accommodated by the sponsor.
- Taking advantage of the incremental software development model, the

sponsor was able to experience and review partial functionalities. Afterwards, they gave some feedback for developing additional requirements in the next increment. In consequence, the system development was able to respond to changes made by the sponsor rapidly.

- The system was able to store and maintain most important data which are used frequently, so as to abate the dependency on external resources.

3 Life Cycle Model

3.1 Selection of Life Cycle Process Model

An incremental development method shown in Figure 4 is employed to reduce the risks, explained in the last Section 2.3. As opposed to waterfall model, this one has more flexibility to fine-tune the current developmental direction and gradually satisfy the sponsor's anticipation based on changes without "a chain reaction" in all phases – Requirements, Design, Implementation and Test. Especially, it is realized that some system characteristics are uncertain and changing requirements will occur often. Through the iterative improvement in each increment, the development process can be refined accordingly.

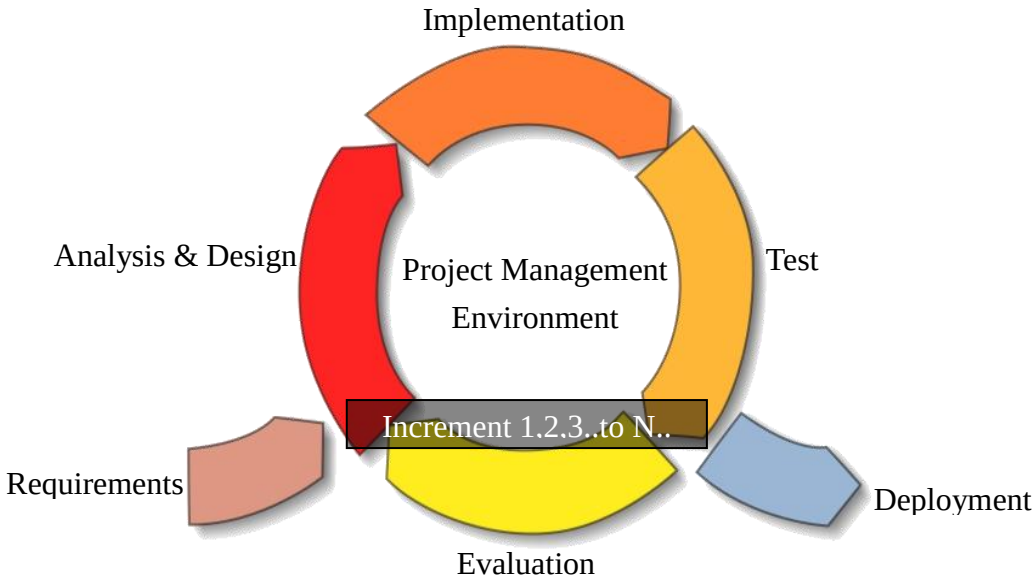


Figure 4: Iterative and Incremental Development

The major consideration to determine which partial functionalities should be arranged in the earlier increment is according to their importance and contribution rate

to the entire system development. Therefore, the functionalities concerning the scholarship application procedure should have a higher priority than others; the increments that occurred in this project are listed below:

Increment 1: Applicant's functionalities related to application portion.

Increment 2: Administrator's functionalities related to scholarship portion.

Increment 3: Administrator's functionalities related to committee portion.

Increment 4: Committee member's functionalities related to application portion.

Increment 5: Administrator's functionalities related to application portion.

Increment 6: Guidance director's functionalities related to application portion.

Increment 7: Enhancing application review and evaluation.

Increment 8: System configuration.

Increment 9: Scheduling time-based tasks.

Increment 10: Enhancing interactivity of UI.

Increment 11: Applying security constrain.

By presenting the whole application procedure (included in Increment 1~6) to the sponsor, they were able to get a clear picture of how different users can utilize the system to complete one part of the processing. The next section will explain the relationships among the distinct roles including applicants, guidance directors, committee members and administrators, and also illustrate their interaction for completing each scholarship application.

3.2 Roles Definition

The functionalities in this system mainly support the scholarship application processing which is collectively performed by different users. There are four main

types of users who can use the system, and all of them must login to the system first:

- Administrator:

  An administrator can create, modify or delete the accounts for other committee members. In addition, he/she is able to view all application forms and designate a committee group to review and evaluate application forms related to one or more kinds of scholarships. Consequently, an administrator decides which applicants are rewarded with scholarships. Moreover, he/she may establish a scholarship and define its criteria.

- Applicant:

  An applicant is a student who has registered with this system and also provides an email address of his/her guidance director. He/she can apply for scholarships by browsing the scholarship information and manage the application form before submitting to the UW-L Foundation Office. The progress of the applications can also be traced by the applicant.

- Committee member:

  A committee member can review those students' applications according to administrator's designation, write remarks and rank these applications.

- Guidance director:

  Each potential freshman has a guidance director from the high school from where he/she graduates. Therefore, a guidance director can provide a student's score and rank as a performance reference via the system.

In addition to the four main roles mentioned above, there is one role called *Committee chair* derived from Committee member. The person with role acts as the

communicator between the committee and the administrator. Apart from fulfilling his responsibilities as a committee member in terms of writing remarks and raking the applications, the committee chair is also responsible for consolidating the results of evaluation of the applications and then reporting it to the administrator via the same online system. Figure 5-8 display the activities performed by various roles through use case diagrams.

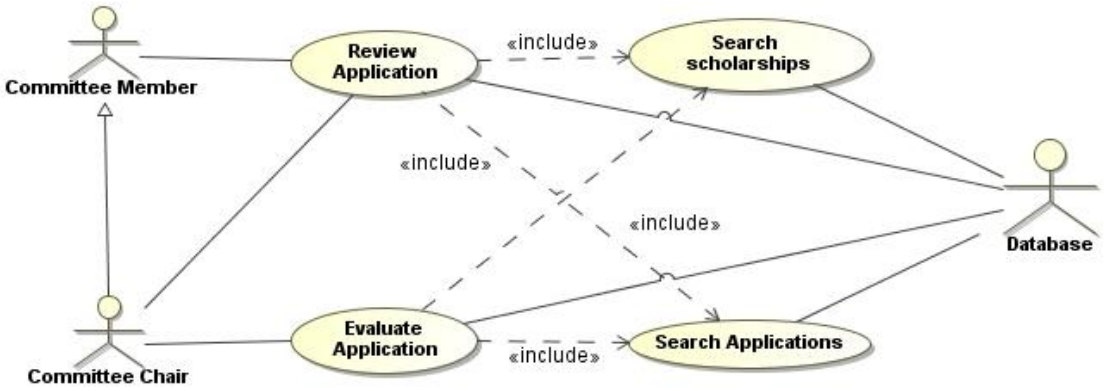


Figure 5: Use Case Diagram for Committee Member and Chair

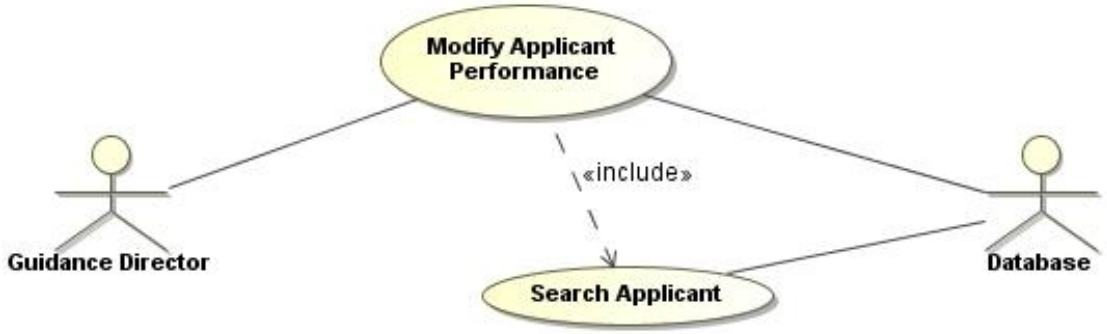


Figure 6: Use Case Diagram for Guidance Director

Figure 7: Use Case Diagram for Administrator

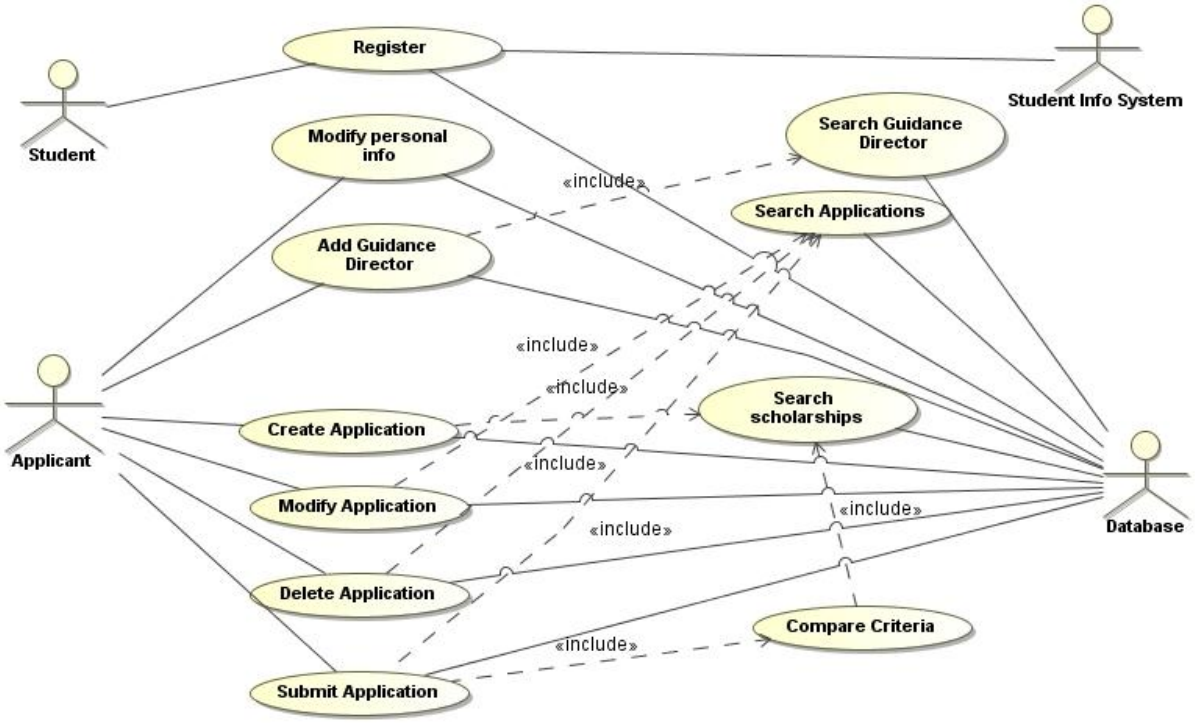


Figure 8: Use Case Diagram for Applicant

Other general functionalities, such as Change Passwords, Login and Logout are able to be used by all the roles. All guests who visit this website can browse and search all scholarship information and their criteria.

4 Design

4.1 Application Architecture

The previous version was developed under Windows development platform using C# in the business logic layer, ASP.net in the presentation layer and Oracle in the database layer. It was run under IIS application server. As mentioned in Section 2.2, the re-coding and re-designing cannot be done during the re-engineering approach. Besides, the choice of the developmental approach and architecture were not constrained by the sponsor. So, due to the flexibility given to the developer, they consent to drastically change the core architecture and technology from Windows development platform to Enterprise Java platform where the new version will be implemented and executed. For the database, Oracle is still a better choice due to the convenience and compatibility with the external database system supported by ITS. The developer anticipated that security mechanisms are easier to implement in a Java-based environment compared to the Windows environment, and the reasons for this belief are described below:

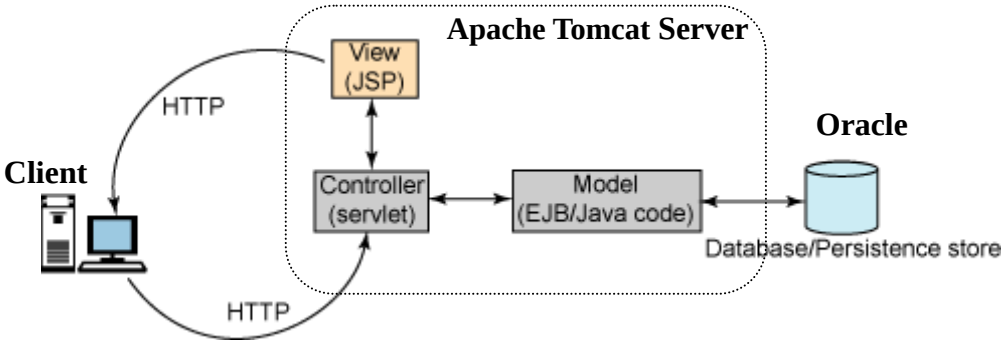
	Java EE platform	Windows .NET platform
Authentication	Under JAAS mechanism implemented by various compliant servers.	Must be bound with IIS to perform authentication.
Authorization (Access control)	Under JAAS mechanism implemented by various compliant servers.	Must be bound with IIS to perform its CAS feature.

Structure view	Under a heterogeneous environment providing a platform-independent security.	Rely on a homogeneous environment based on the integration of Windows products.

Table 2. Comparisons between Java EE platform and Windows .NET platform

The platform-independent characteristic in Java really provides a flexible authentication and authorization framework which are defined in Java EE interface. These interfaces can be implemented based on developers' requirements. By taking advantage of its flexibility, a more robust security model can be organized to protect the system. Hence, more importantly, the interoperability among different units can be ameliorated. Another good reason is that all techniques and tools in Java EE platform are free to use.

The application architecture is illustrated in Figure 9 as a typical Java EE model with MVC pattern.



Source: Patrick Gan, Ease the integration of Ajax and Java EE, http://www.ibm.com/developerworks/java/library/j-ajaxee/ , Copyright© IBM, 2006.

Figure 9: High Level Architecture of the system

In this principal architecture, ZK Framework is used to perform Ajax mechanism for

the improvement of application usability by diminishing inordinate web page refreshes. It can also reinforce the utility between client and server communication and the interaction in user interface by writing some brief Java code.

4.2 MVC Pattern

The concept to develop this system abides by MVC (Model-View-Controller) pattern. In the *view*, also called the presentation layer, JSP is used to display all data, such as application forms sent from the server side, also including user interface layout. The *model* consists of business objects structured in JavaBean. For instance, each role such as an applicant and a committee member in the system are treated as different business objects. The *controller* as a logic processor manages the flow of the application. It can initiate actions for dealing with clients' requests and bridge the communication between JSP pages and JavaBean. This work is usually performed by servlets.

Using MVC pattern can bring the following benefits to this system:

- Flexibility for adding and removing functional units.
- Reusability for Model, View and Controller layers.

For example, in the reusability, there can be several views to present data in different ways which share with the same controller and/or model layers. Since each layer is responsible for distinct tasks, modifying one of them has little or no impact on the other layers. In short, they are independence by separation.

4.3 Direct RIA

In order to enrich this web-based system and use it at ease, the characteristics of

desktop applications are manifested through all interactions between users and the system. The developer aims at designing a Rich Internet Application (RIA) instead of a native web application. In practice, there are many tools, models and technologies for building a RIA program. In Figure 10, it shows the main design options to develop a RIA program.

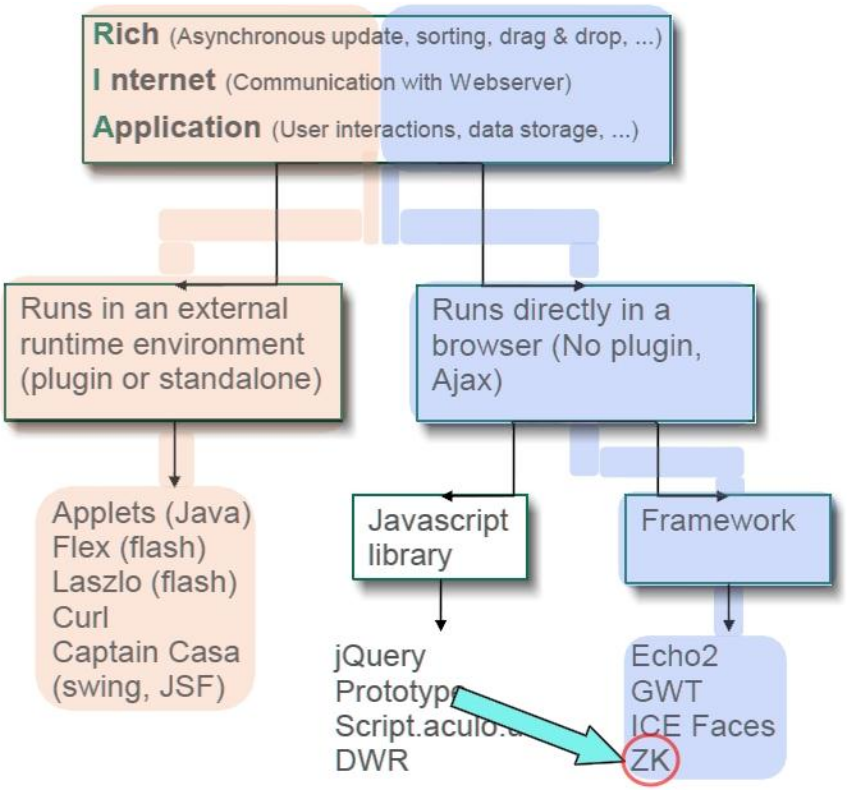


Figure 10: Hierarchy of RIA

Even though external Run-Time Environment (RTE) can provide users with more attractive and richer UI, installation of RTE like Flash plug-in and JRE is required. In addition, the communication with the server is limited inside a specific area on a web page. On the other hand, by using Ajax, the system availability is better, as long as a compatible browser is used. Choosing ZK, an open-source Ajax + Mobile web

application framework, is able to carry out full Ajax mechanism in the system by coding in Java rather than Javascript. This means that the system is dominated by the server side and all processing is done by the application server, because of the Server-Centric approach adopted in ZK.

There are many advantages listed below to use server-centric instead of client-centric approach for the system design.

- Strengthen security and stability:

  Most crucial decisions and actions can be done on the server side, and clients must follow the rules defined by the server. All logic behavior is concealed from clients, so it is more difficult to comprehend the system logic by exploiting client code.

- Prevent browser incompatibility:

  Specified code can be written on server side for solving incompatible presentation from any browsers.

- Better extensibility:

  All data and entities are managed by the server, and clients are only responsible for the data presentation. It is therefore easy to extend the system features without changing and influencing any client.

- Easy to maintain consistency:

  When one functional unit is modified by the serve, it can impact all clients; that is, users' requests for the same task can be always done under the corresponding service. Therefore, data consistency between client and server, more or less, can be maintained asynchronously.

4.4 User Interface

The online scholarship application system has been designed in such a way that the layout for each page looks alike, and the look and feel also retains consistency. In other words, most components remain in the same position. Same image icons are used for visual effects and feedback on buttons, grid and list which are operated for the same purpose. Therefore users can memorize their functionalities and locate them quickly and easily. The integrity constraint is applied for all input components such as edit boxes and application forms. When users input any invalid value for a required field, a message will pop up to guide users through data input procedure. By accomplishing Ajax mechanism, users are able to perform two or more different tasks simultaneously by triggering any UI events. Once the system completes the processing and sends all altered data back to the client, the presentation in a partial area of the page will be re-rendered accordingly.

The major look and feel is illustrated in Figure 11. Basically, the majority of pages are modularized by their usage. The operating procedure to perform each action, such as adding a scholarship, modifying a committee, submitting an application, awarding an application, saving configuration, remove a member from a committee, appointing a scholarship to a committee and so on tend to be steady. Hence, the operation can be done quickly, when users get accustomed to its procedure. For example, one typical procedure is:

> 1. Choose one operation from the main menu → 2. Search for one or more desired record(s) and then selecting them to set them as active item(s) → 3. Display its detailed content in view area → 4. Input data in

fields → 5. Click one action button to execute the function.

After clicking an action button, a request will be sent to the server. One event handler in the server will deal with this request and respond to the client eventually. During its processing, usually data access to the database occurs.

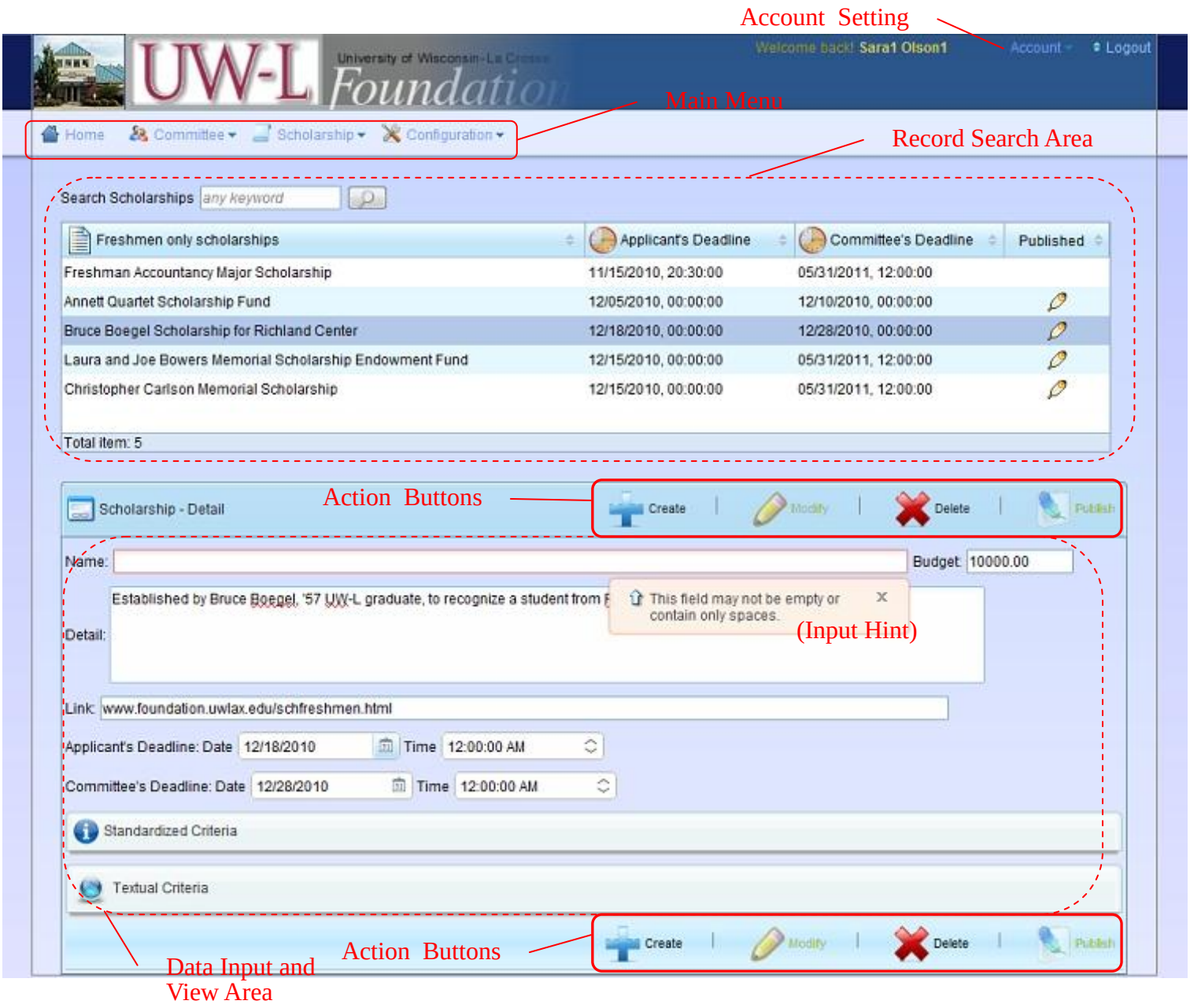


Figure 11: A Page for Administrator to Design Scholarships

4.4.1 Reusability of Components Group

Some of components in the user interface are designed to serve identical functions. When reusing them in different pages, their consistency and appearance can be preserved. For instance, in Figure 11, Record Search Area is composed of one label, one input box, one search button and a list to show all searching result. It is a *macro* to perform search request, which can be reused in another page. When comparing with Figure 12, they both have this macro to search a set of scholarships, even if the results in the lists are presented for different purposes. These two lists are actually implemented by the same component as a container to display scholarships with their necessary attributes required by different roles, Administrator and Applicant. Dynamically rendering a suitable component like this list according to the data sent from the server will be implemented by manipulating DOM structure.

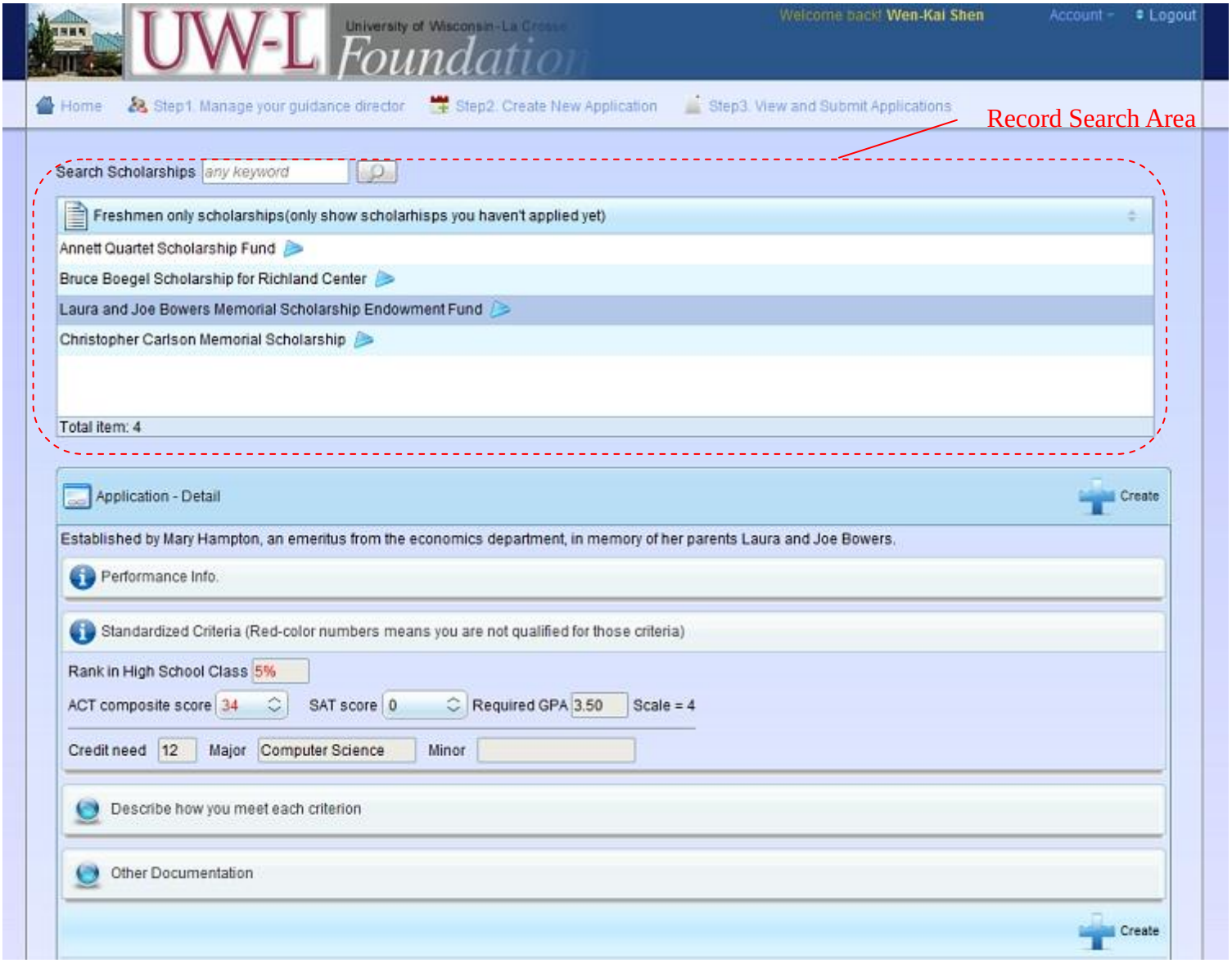


Figure 12: A Page for Applicant to Create New Application

4.5 Data Access

There are two data sources which the system can access to. One is from the database and the other one is from *.properties* files which are allocated in the server. All settings in the system such as configurable parameters will be saved in and loaded from these files. Administrators can conveniently restore default settings and modify their values directly by opening these files.

For better communication with the external data storage entity, Student Information System (SIS) in ITS, this system utilizes Oracle 10g for data access. As shown in Figure 8, students must register in the system before applying for any scholarship. First of all, their 9-digits IDs need to be verified by comparing them with their records retrieved from SIS. Actually, this is only one locality in the system for a need of data access to the external entity. The less dependency on SIS the system has, the more flexibility the system owns to maintain the functionalities.

The scheme of database design is intended for increasing the performance for searching data and decreasing the coupling in relations. For example, the status of a scholarship application undergoes several changes through several transitions. Besides, each role's authorization and authentication information are also managed inside the database for supporting JAAS mechanism.

4.5.1 Relationship between Scholarship and Application

The relationship between a scholarship and an application is described in Figure 13. In fact, the application itself is complex which consists of four internal relations depending on the relation, APPLICATION. When applicants start to compose their applications, they are free to upload their documentation files such as resumes and

recommendation letters. Their documentation will be stored in APP_DESC and APP_RECOM_LETTER relations, when they upload them. Similarly, before applicants submit their applications and any of committee members reviews one application, APP_EVALUATED_INFO relation is empty for the application. The purpose of this design is to reduce the redundant records used in each relation, because the number of applications grows up gradually and the complexity of an application is escalated based on how much it has been done in the processing.

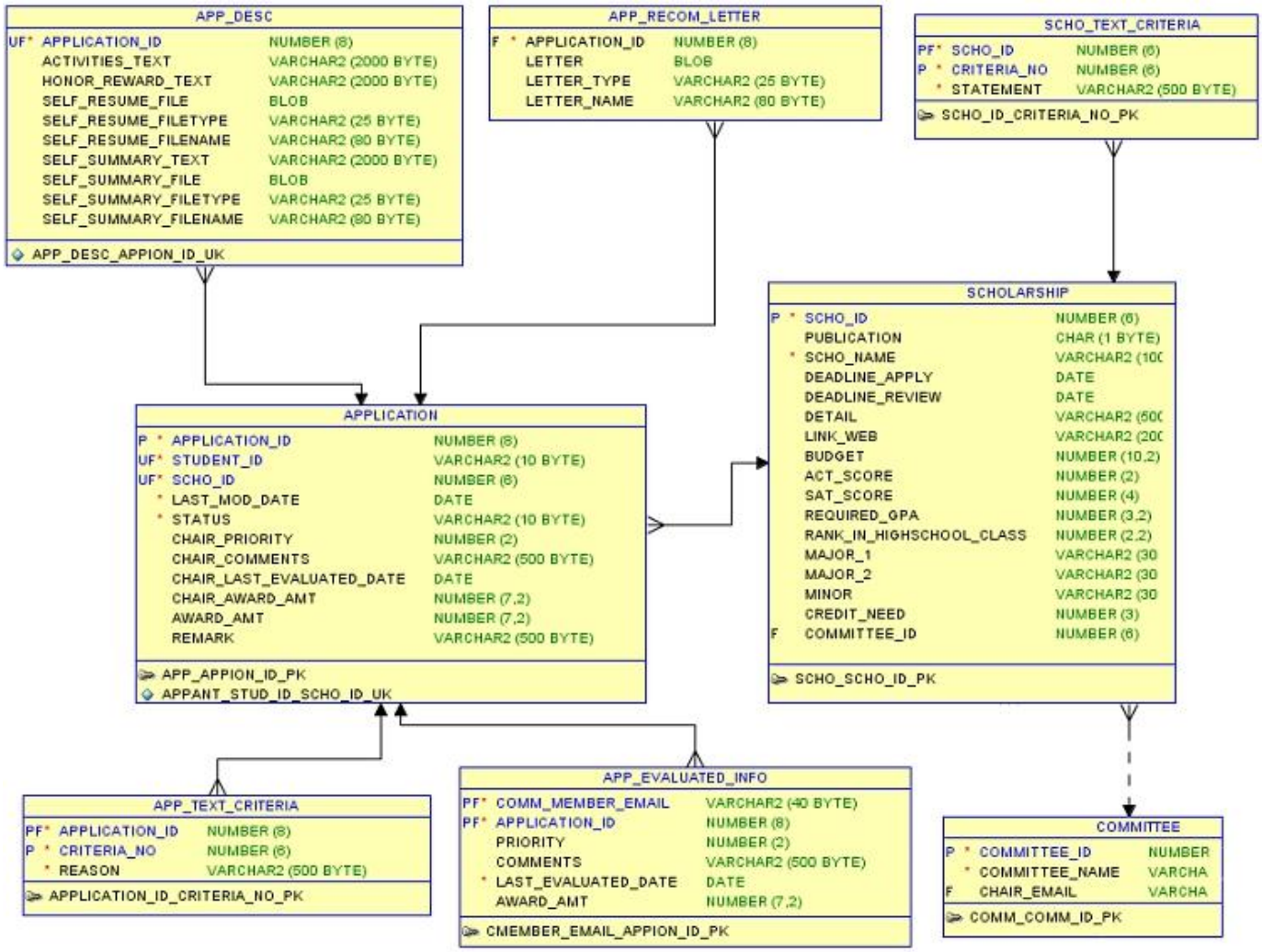


Figure 13: Scholarship and Application ER Diagram

4.5.2 Criteria Relation

The criteria for each scholarship is comprised of many textual statements stored in the relation, SCHO_TEXT_CRITERIA, and their corresponding reasons which applicants certify in order to meet each criterion are stored in the relation, APP_TEXT_CRITERIA. There is an implicit relationship between these two relations maintained by some stored procedures. The criteria number is generated sequentially for each scholarship. When an applicant fills in a reason for a criterion, both of them will be added in the relation. In short, the reasons which applicants state in their applications refer to the corresponding criteria in the scholarships which they apply. Under this relation design, the mapping between criteria statements and their reasons can be located and retrieved more efficiently.

4.5.3 Committee Relation

Based on the sponsor's requirement, each committee member can review the whole applications of one scholarship which is appointed to one committee they belong to. Figure 14 shows another subset of database relation among committees in COMMITTEE relation, committee members in COMMITTEE_COMM_MEMBER relation and their evaluations in APP_EVALUATED_INFO relation.

COMMITTEE_COMM_MEMBER relation is used to solve a many-to-many situation between committees and committee members; that is, one member cannot exist repeatedly in the same committee, but distinct members can be associated with the same committee. Besides, it is acceptable to establish a committee without containing any members or assigning a chair from one of the committee members, because committees and members are regarded as two individual entities. Once a

chair is assigned, its reference will be added in COMMITTEE relation.

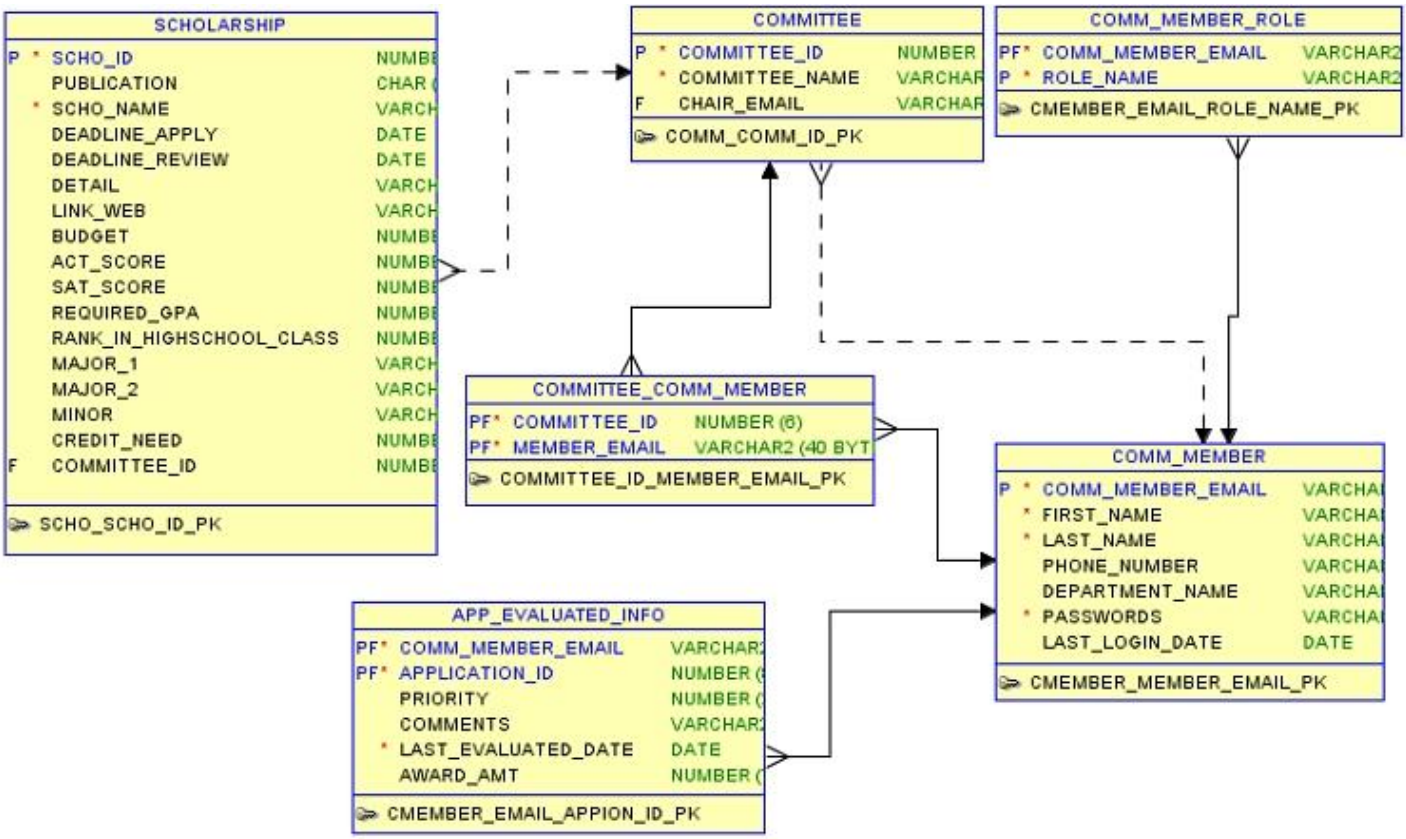


Figure 14: Committee and Committee Member ER Diagram

4.5.4 Singleton Pattern for Connection Pool

Because of a time-consuming process to initiate a connection to a database, the system performance can deteriorate. Connection pooling is one of approaches which can be used to improve this situation. To set up a connection pool on the server side, one data source connecting to Oracle database is configured in Tomcat's Resource Factory. Figure 15 shows how connection pooling works within Oracle database structure.

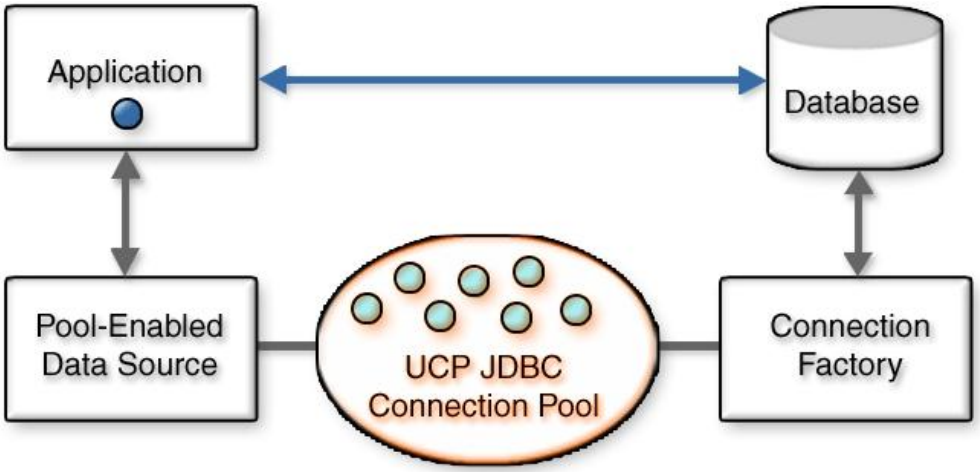


Figure 15: Conceptual View of interaction between the system and connection pool

Consequently, a collection of connection objects is stored and maintained in the pool. When one client requests data access to Oracle, the application server will borrow one connection object to process relevant data transactions. Once all tasks associated with this connection have been done, the connection is closed and returned to the pool, and then it becomes available for next usage.

When developing a connection pool, singleton pattern can be applied to simplify the procedure for borrowing a connection object from the pool. If only one connection pool instance can be created to manage all connections, it is relatively easy to oversee and handle their variations, and finally prevent a waste of database resources. Figure 16 simply illustrates how to combine singleton pattern with connection pool design.

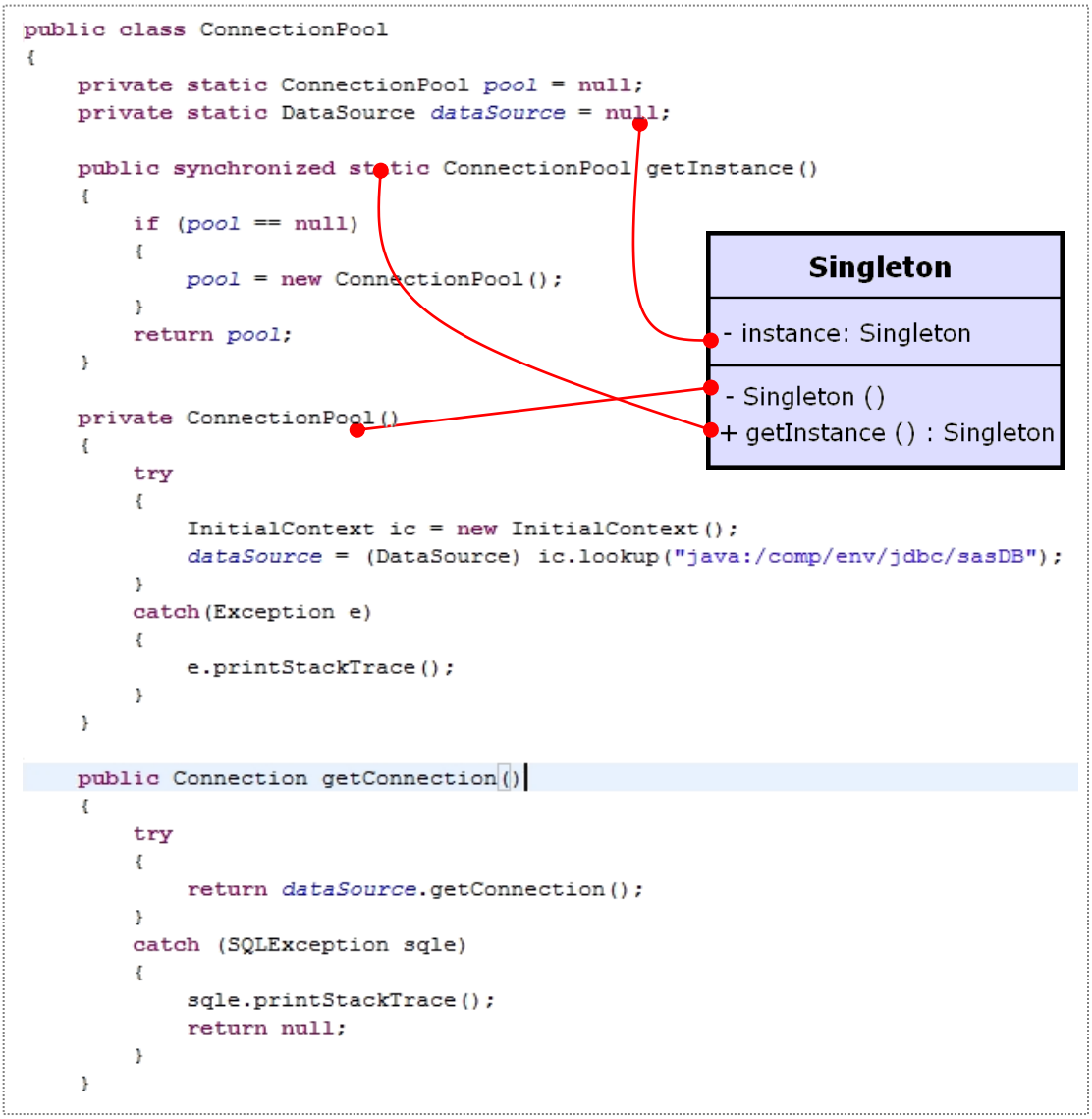


Figure 16: Coordination of Singleton Pattern and Connection Pool

5 Implementation

As stated earlier, incremental prototyping approach was used in the implementation phase. To begin with, an initial prototype was built up for demonstrating user interface to the sponsor. In this prototype, all components had to be arranged in each page for supporting friendly interaction according to the functionalities of different roles. Afterwards, the sponsor experienced how to interact with the system and sent some feedback to the developer. Then, the next phase was started to implement the applicant's functionalities based on the sequence of processing scholarship applications.

This chapter describes some of the important features and their implementations in the online scholarship system.

5.1 Security

Security is a key consideration for assuring data confidentiality and integrity. As long as users have been identified and authorized to interact with the system through Internet, security becomes a mandatory issue. During every communication between clients and the server, a secure connection must be offered. When data access occurs between the server and database, a safe transaction should be applied. When multiple users request the same service or one user requests multiple services simultaneously, a race condition must be prevented.

5.1.1 Secure Connection via Internet

Most data used in the system transmitted between clients and the server are private

and sensitive. A secure channel is implemented in Secure Sockets Layer (SSL). When this channel is enabled, HTTP protocol will be changed into HTTPS protocol. With SSL, outgoing data sent to clients (browsers) or server are encrypted and incoming data received from client or server are decrypted. To make this work, a digital secure certificate is required from a trusted source like www.verisign.com. During the implementation phase, a self-signed certificate named *.keystore file* generated by using the keytool program in JDK can replace a real one without purchase. To take advantage of a secure connection, some settings have to be configured in the application server. For example, the following XML code is written to start a secure connection when and after users login by inputting their identification and passwords.

```
<transport-guarantee>CONFIDENTIAL</transport-guarantee>
```

Figure 17 shows a screenshot when clients and server are communicating under a secure channel. The browser will also show a caution to indicate that this certificate is not granted by a trusted source.

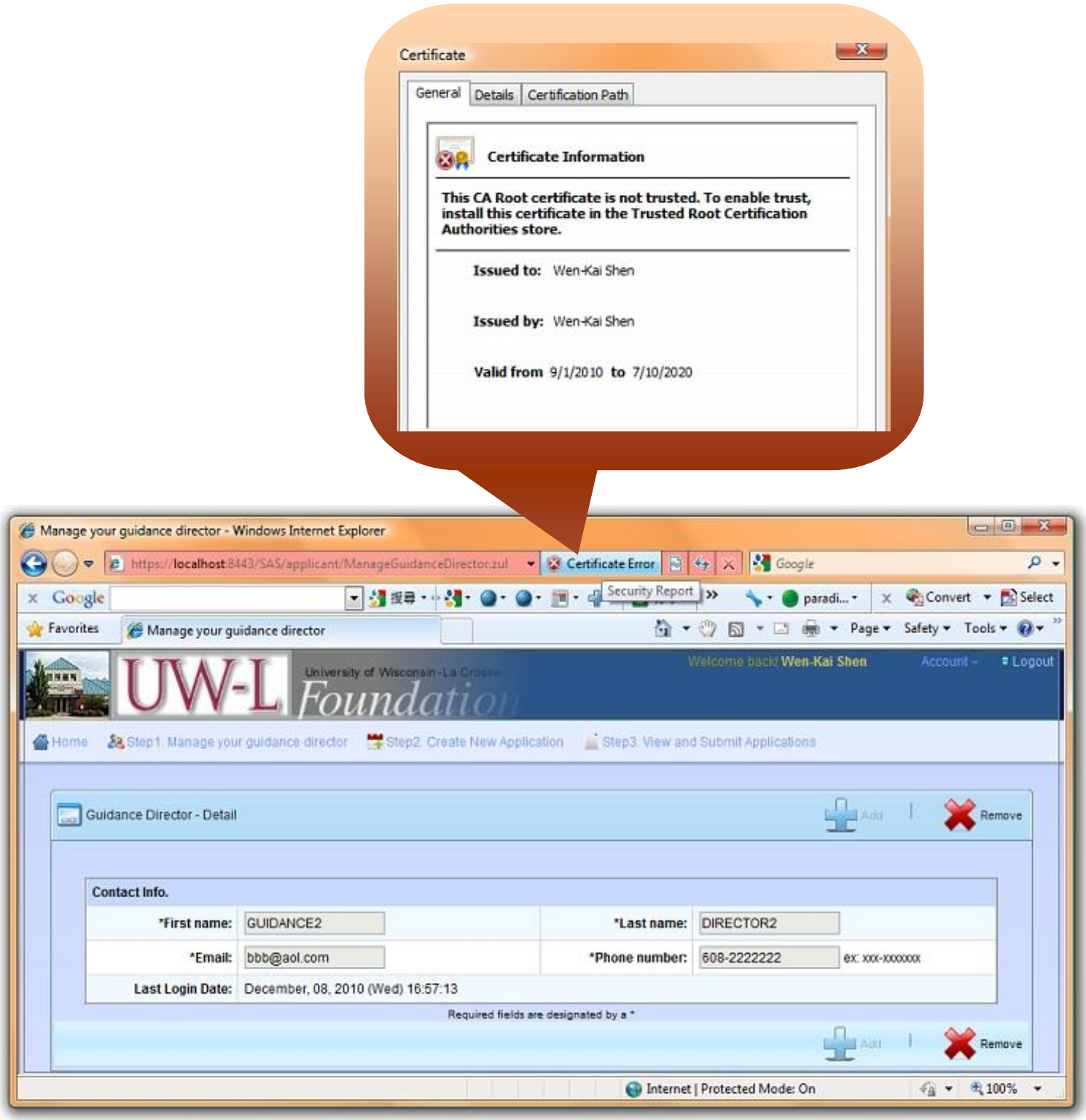


Figure 17: Self-Signed Certificate is configured for a testing environment

5.1.2 Secure Database Access for Data Integrity

For maintaining the integrity and completeness of relational data stored in the database tables, every time when a user requests an action for accessing the database, it must be fulfilled within *one transaction*. In particular, users from different clients may be able to access the same section of data concurrently via Internet connection. For example, while an applicant is submitting his application data including a summary and some documentation, the administrator is deleting the scholarship which he is applying to. In this example, there are two transactions – submission of application and deletion of scholarship. No matter which one is processed first in the system, it will end up in a safe and reasonable circumstance; that is, his application data will not exist in the database because the corresponding scholarship has been deleted.

This system implements the ACID properties: Atomic, Consistent, Isolated and Durable features, by defining referential constraints in the database. Besides, the committing transition is imposed to ensure that all SQL statements to perform an action are executed completely or not at all. For example, in Figure 13, an application is associated with several entities. One of them is recommendation letters because one application can refer to many letters or none. Adding an application as a transaction has to be divided into some tasks which will be executed sequentially in the program. Two main tasks of them are described as follows:

Task 1 – Add application basic information into the main table.

Task 2 – If applicants upload any recommendation letter, add them into the relevant table in the database.

If one of these tasks cannot be complete somehow, the whole transaction is regarded as a failure. Then, the system rolls the data back to the previous status; that is, it must restore values to what they were before these two tasks were executed.

5.1.3 Secure Authentication and Authorization

In Section 3.2, all functionalities are classified based on the role definition. The system will grant a user some privileges to manipulate these functions, as long as his or her identity is authenticated successfully. As shown in Figure 18, each role is identified by different parameters with specified formats.

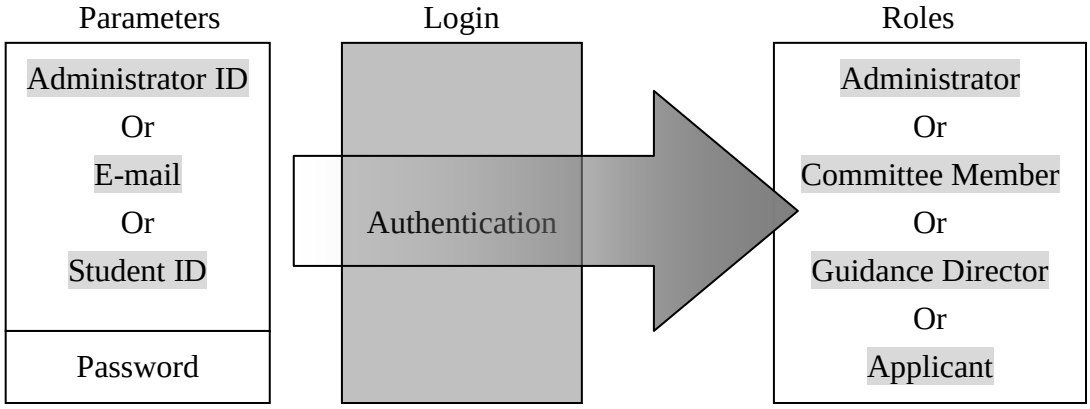


Figure 18: Mapping between Parameters and Roles

If a user inputs 9-digit number as student ID and password to login to the system, the applicant's identity data will be retrieved from the database for completing authentication. If successful, this user will be granted the privileges of a student/applicant. Likewise, E-mail is used for authentication of committee member as well as guidance director. Administrator ID is formatted with prefix "admin" plus a 4-digit number.

For implementing the above authentication and authorization processing, four *Realms* have to be configured in the application server, Apache Tomcat. Because the system takes advantage of connection pooling for optimizing the access to Oracle database, *DataSource Database Realm* is chosen to manage the processing. The application server can connect to the database and collect these actual parameters, passwords and associated roles. Besides, a *Combined Realm* is used to combine four of DataSource Database Realms, due to the different parameters from different sources. Under this authentication and authorization mechanism, applicants cannot navigate through other pages which do not belong to their privileges. The other roles are also subject to this discipline. For instance, applicants cannot reach an irrelevant page such as one of the administrator's pages by copying its link and paste it on another browser. On the other hand, it is possible to grant one user more than one privilege. In this case, this user can navigate the pages where the authorities are encompassed.

5.1.4 Default Password Generator

After the administrator creates a committee member and applicants enter their guidance directors successfully, the system will send an email including an auto-generated password to the newly created committee members and the guidance directors for their login. However, they are able to change their passwords after initial login. An ideal password generator should be able to generate an unpredictable and greatly random combination of the alphabet and numbers. This generator is implemented by using java.security.SecureRandom class with a deterministic algorithm, SHA1-PRNG (Pseudo-Random Number Generator).

5.2 Ajax Mechanism and Event Processing

The implementation of Ajax mechanism is ascribable to the support from ZK Framework. It provides a systematic underlying infrastructure where the developer can build web pages and functional unit by inheriting its classes and/or implementing its interfaces. Figure 19 illustrates how client and server communicate under ZK architecture with the implementation of Ajax mechanism. Basically, most tasks are processed and done by the server, so the developer only has to organize the business logic and data module by writing Java code. As a server-centric framework, in most instances, clients are only responsible for presenting the changes including data and UI components made by the server. When a user clicks a button, one Ajax Request will be generated and sent to the server. The developer has implemented several event listeners and registered them to handle the client requests.

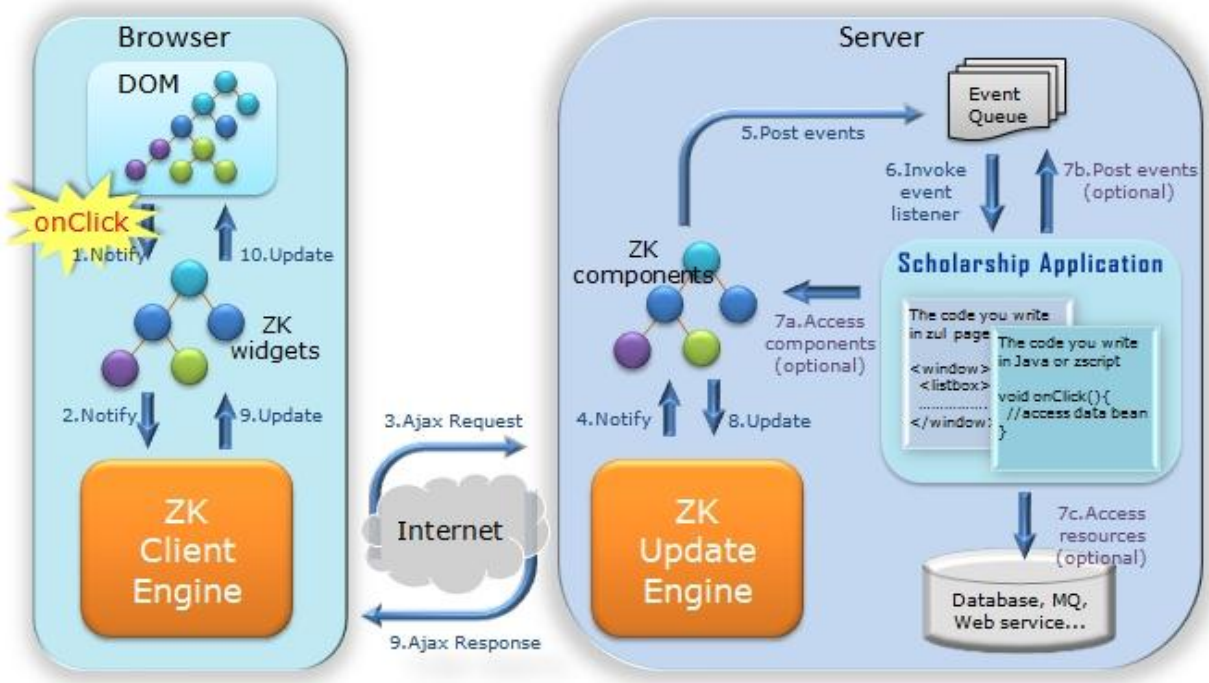


Figure 19: ZK's Server-Client Fusion Architecture

Besides, it is also possible to add event handler on a dynamically generated component for more advanced interaction. For example, in Figure 20, guidance directors are able to upload transcripts for those applicants who choose them as their guidance director. By clicking one upload button of a row, the transcript can be uploaded for its corresponding applicant. When it is successful, a cross icon followed by a label will be dynamically created. In addition, they can also download these transcripts by clicking the labels behind the cross icons. Contrarily, by clicking this button, they can remove what they just uploaded.

Figure 20: Uploading transcripts and managing them

5.2.1 Threading Issues and Solutions

As mentioned in the previous section, the asynchronous communication promotes the interaction through user interface and thereby reduces response time from refreshing (or reloading) pages. After an Ajax request is sent to the server, the corresponding events will be posted on the waiting queue. There is another thread to process each event waiting in the queue. Normally, all of these events are processed in sequence. However, it is possible to change their order by an interruption or

deferment. As a result, race conditions might eventually occur. The technique described in Section 5.1.2 can be the last line of defense to guarantee data consistency and integrity. Besides, when a user presses any action button, all action buttons will be disabled to prevent consecutive clicks, shown in Figure 21. This protection is implemented by writing JavaScript on the client side. Once the request has been processed by the server and the client has received the response, these action buttons turn enabled again.

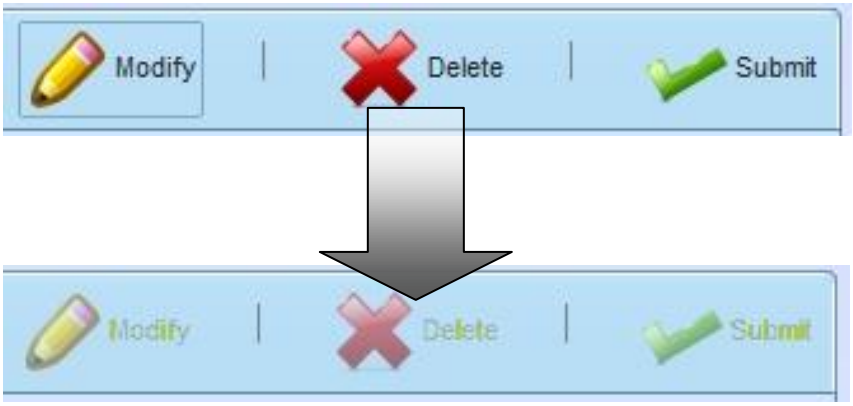


Figure 21: The Switch between Disabling and Enabling actions

When the server is processing a client's request, a processing message will be shown on the user's screen as demonstrated in Figure 22. During this period, it is not allowed to receive another request from this user.

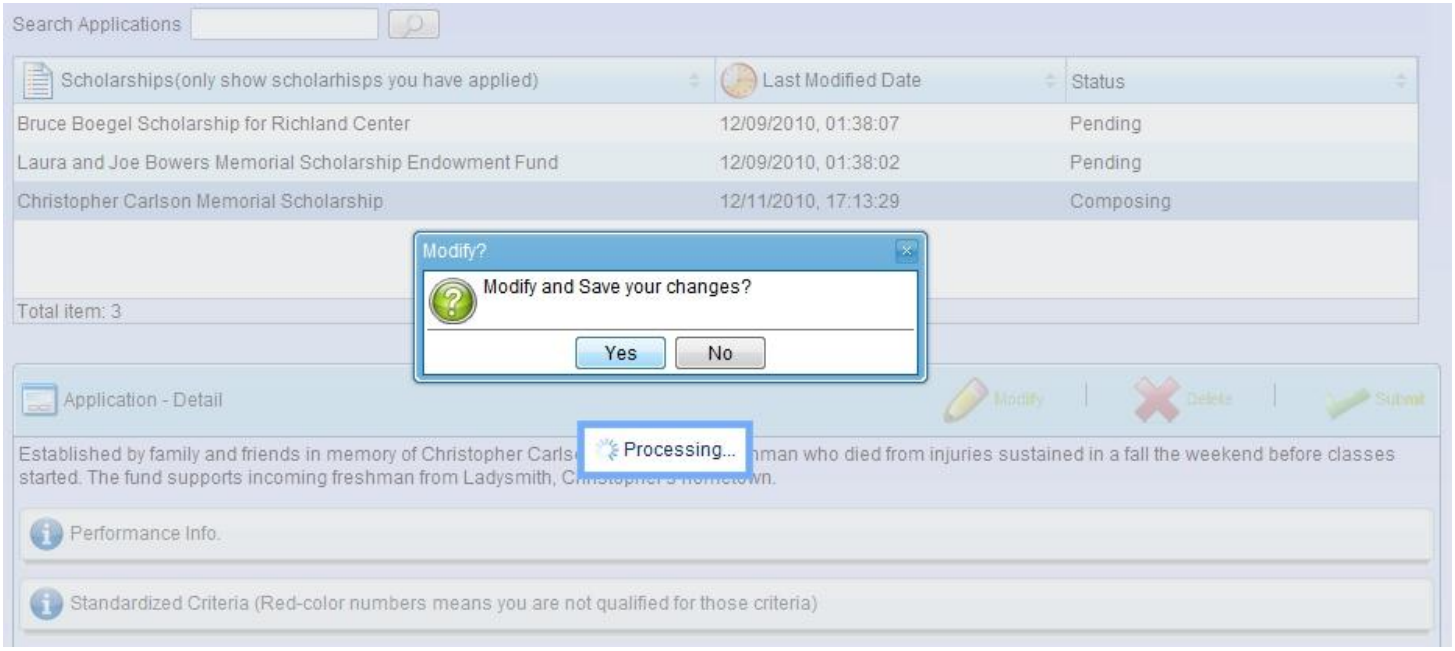


Figure 22: A Processing Message display

5.3 Scheduling Tasks

In order to process some tasks which must be executed at specific time every day, a scheduler is implemented in the system to arrange and perform these tasks. For example, all scholarships have their own deadlines. Applicants cannot apply for the scholarships after their deadlines. In addition, on the day when a scholarship application is due, the system will send an email automatically to all committee members who are associated with that scholarship as a friendly reminder to inform that they can start reviewing the applications.

Figure 23 describes a basic idea of how a scheduler handles time-related tasks based on daily intervals. Each task added into the scheduler will be processed daily in

another thread when the time arrives at their scheduling time. The scheduler also deals with daylight-saving time issue for accurate processing.

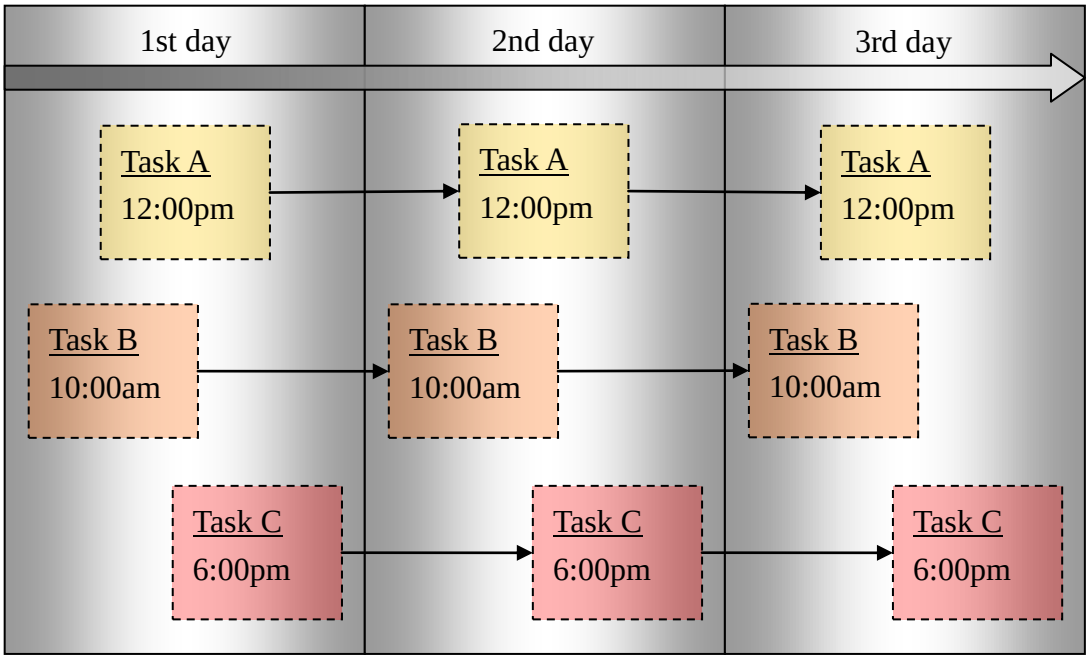


Figure 23: A Scheduler to Manage Temporal Tasks

5.4 Coding in Oracle PL/SQL

Making use of some features in Oracle database can accomplish more complex utilities in data query and access. In this project, Oracle 10g Express Edition, a free version of 10g Release 2, is utilized as the back-end data storage. There are 11 triggers, two procedures and at least three sequences to be written for the following implementation:

- Generate auto serial number for the primary keys in Administrator, Scholarship, Application and Committee tables.
- Update role tables retrieved by the security realms for maintaining system

authentication and authorization.

- Create and manage the composite primary key for each set of criteria of scholarships. Unlike MySQL which automatically supports the maintenance of composite primary keys, its mechanism has to be implemented by the developer in Oracle.

Besides, all stored procedures are categorized and stored in five packages based on their characteristics. They are intended for integrating multiple DML statements which perform the same task. Their deployment is shown in Figure 24.

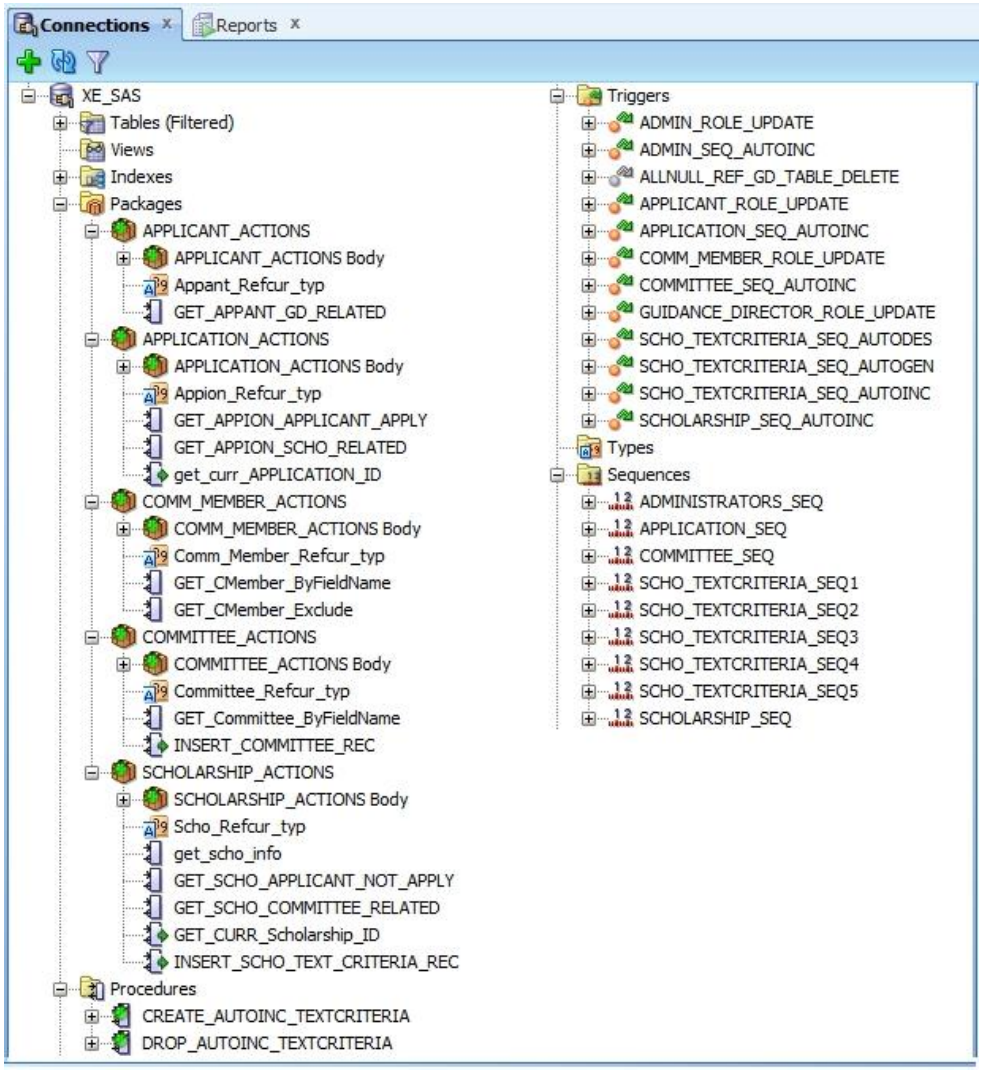


Figure 24: The Structure of deploying all Oracle Components

5.4.1 Dynamic SQL

Because some tasks cannot be implemented by using static SQL whose full statements can be known at compilation, in these cases, dynamic SQL enables the developer to write more flexible DDL statement as well as execute a complex query with user-selectable fields. For example, after the administrator creates a new scholarship, a new sequence component will be also created accordingly with a name related to this scholarship. This task needs to be done dynamically because the values of this scholarship can only be given at runtime by the administrator. The following code inside the procedure called CREATE_AUTOINC_TEXTCRITERIA is responsible for this task.

```
PROCEDURE CREATE_AUTOINC_TEXTCRITERIA(
      NAME_SEQ_TRIGGER IN VARCHAR2,
      New_ID IN NUMBER) AUTHID CURRENT_USER
AS
   PRAGMA AUTONOMOUS_TRANSACTION;
   SEQ_SQL VARCHAR2 (200) ;
BEGIN
   SEQ_SQL :=
      'CREATE SEQUENCE ' || NAME_SEQ_TRIGGER || '_SEQ' || TO_CHAR (New_ID) || '
      MINVALUE 1 MAXVALUE 999999 INCREMENT BY 1 START WITH 1 NOCACHE  NOORDER
      NOCYCLE';
   EXECUTE IMMEDIATE SEQ_SQL;
END CREATE_AUTOINC_TEXTCRITERIA;
```

The parameter, New_ID refers to one scholarship ID given by another trigger when this new scholarship is created successfully.

6 Testing

In iterative and incremental development, testing is conducted during each increment after its implementation phase. Figure 25 shows what kinds of serial testing are performed in one iteration under the limited developmental time. The system is constructed gradually by achieving more and more increments. Regression testing is used to make sure every time when new functionalities are implemented, all other functional modules implemented and tested in the previous increment can still be executed correctly in the current increment. Then, in progression testing, these new functionalities can work correctly in every developing increment.

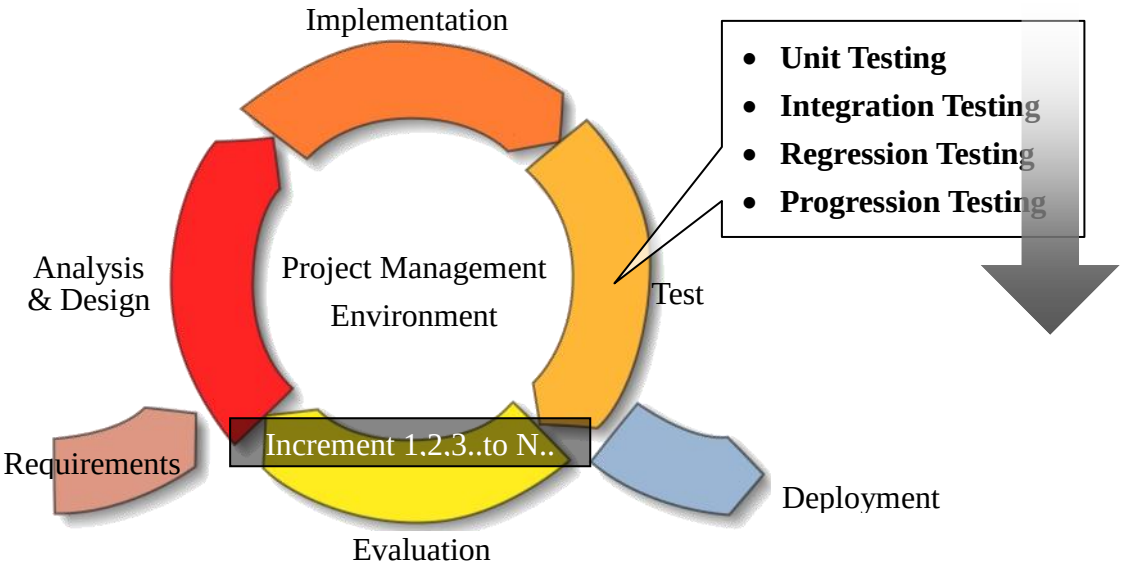


Figure 25: Testing for each increment

Testing data are generated according to the inputs of the requirement specification. The distribution of testing data is getting more and more complete and extensive when more iterations are passed in the developmental cycle. Similarly, the complexity of

association among all entities also grows up. For instance, in the first increment mentioned in Section 3.1, only partial data in the application can be attained from applicants after they compose their applications. When more functionalities are implemented in forth increment, the reviews and evaluation reports from committees refer to the applications. At that time, an application is compounded with the data provided from applicants as well as committee members. All testing data are prepared for ensuring that the outputs can be presented on GUI components correctly and each functional unit can work properly.

6.1 The Variation in States of Scholarship Application

There are five distinct states throughout the scholarship application procedure where an application can belong to. Because all main functionalities originate from this procedure, to understand its variation is very helpful in the testing phases. Figure 26 shows the variation in each state of a scholarship application with a state diagram.

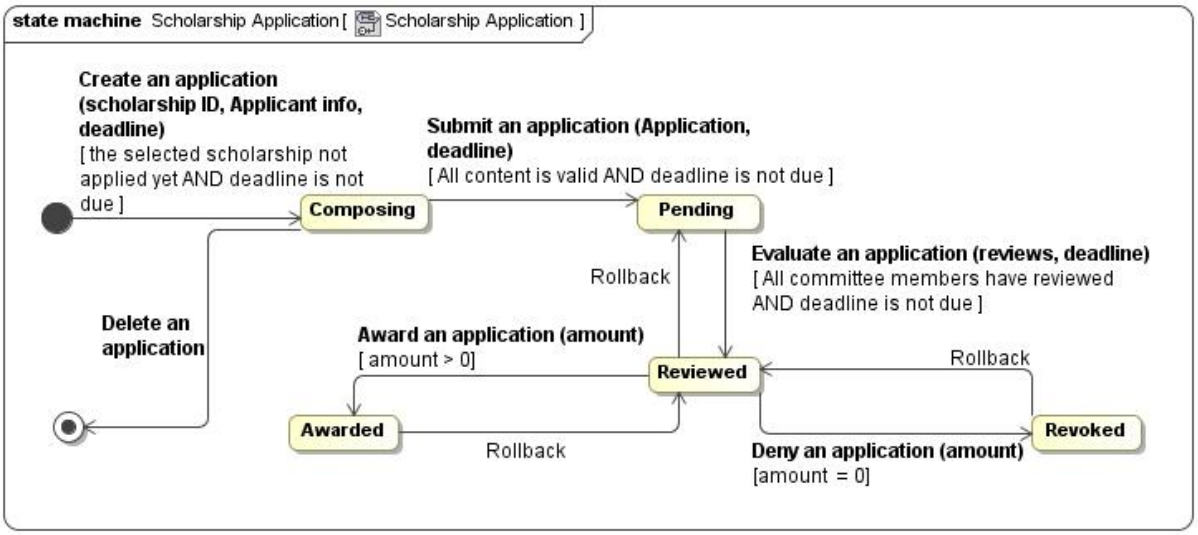


Figure 26: State Diagram for Scholarship Application

By inspecting each state and event regarding different parameters from testing data, every process in the scholarship application procedure can work correctly. The states of Reviewed, Awarded and Revoked are reversible, so as to provide more flexibility for the administrator and committee member and its chair.

7 Challenges

The most challenges and problems including their solutions have been described in the previous chapters based on each phase of the systems development life cycle. Apart from those issues including security, thread, file uploading, inputs validation, UI interaction and so on, there are some additional challenges encountered by the developer which are specified as follows:

- Add and Remove guidance director:

These two of applicants' functionalities required more actions for dealing with different scenarios. First, only one of these two functions can be enabled for the operation, depending on whether or not the applicant has been associated with one guidance director.

Before applicants *add* a guidance director, they have to search this guidance director by inputting the email address. If this person is found, the contact information related to this person will be displayed. They can then check whether or not this person is their guidance director. If not, they can either do search again or input a new email address to create a new guidance director. When creating a new one, the record of this guidance director is stored in the database, and the system will send this guidance director a confirmation email including the initial password. When they add a guidance director who already exists in the database, only the relationship between this person and this applicant is created, and the system will send a note to this person's email box.

When applicants *remove* their guidance directors, their relationship will be removed. If there is no applicant associated to a guidance director, the system will delete the record of this person automatically and also send a note to this person's email box.

These two scenarios have more complicated operative procedure, and it is possible that an applicant may able to add his/her guidance director while another applicant is removing them at the same time. Hence, the developer spent more time on its implementation than other functionalities in order to maintain data consistency.

8 Limitation and Continuing Work

This online scholarship application system has been developed to handle the freshmen application processing only. This is one of the more complicated and difficult applications as implied by the sponsor, because of more roles involved in the system activity. All functionalities are derived from the previous version and were re-implemented by a new developmental environment. In fact, there were more work which could be done for the improvement of user experience in the system if time was sufficient. The following list shows these continuing work based on their urgency.

- Incomplete student authentication:

  Due to the absence of student 9-digits ID, the system cannot identify whether or not the registering students have been qualified as potential freshmen at UW-L. ITS has not yet provided this field as a primary key in the view for being retrieved by the system. Furthermore, when student ID can be found in student information system, all scores and most personal information related to this ID can be saved in the system.

- A printable file to export for application forms:

  Users should be able to download and print an application form. This can be used by applicants, reviewed by committee members and evaluated by the administrator. This utility was planned to be implemented by using JasperReports package but not implemented yet.

- Archive support:

  For every academic year, all applications and their referential data including

scholarships, applicant information and evaluation reports should get backed-up. Currently, this work relies on Oracle tools but it should be more convenient to provide this function in the system itself.

- More flexible data query and statistics:

  The system should allow the administrator to gather a set of desired data and analyze their characteristics. For example, the number of total valid applications or applicants would be helpful to infer the size of data to be manipulated. An alternative solution is to execute SQL command directly in Oracle database.

- Unknown performance bottleneck:

  The acceptable maximum of the connections between clients and server has to be figured out by stress testing. This value can influence the setting on how many free connection objects should be prepared for satisfying concurrent data access. It has a need of substantial numbers of testing data to find out where a breakpoint is in the system. Besides, stress testing can also be used to detect race conditions and vulnerability against DoS attacks by sending considerable requests concurrently to the server, due to asynchronous communication employed in the system.

Bibliography

[1] Henri Chen, Robbie Cheng, *ZK™ Ajax Without JavaScript™ Framework*, Apress, Inc., 2007.

[2] Tulika Das, Joseph Ruzzi, *Oracle Universal Connection Pool for JDBC Developer's Guide, 11g Release 2*, Oracle, 2009.

[3] Patrick Gan, *Ease the integration of Ajax and Java EE – Successfully mix asynchronous and synchronous communication models*, IBM, 2006.

[4] Michael Kifer, Arthur Bernstein, Philip M. Lewis, *Database Systems: an application-oriented approach* (2nd Edition), Pearson Education, Inc., 2005.

[5] Joel Murach, Andrea Steelman, *murach's Java servlets and JSP* (2nd Edition), Mike Murach & Associates, Inc., 2008.

[6] Denis Piliptchouk, *Java vs. .NET Security, Part 4 – User Authentication and Authorization*, O'Reilly Media, Inc., 2004.

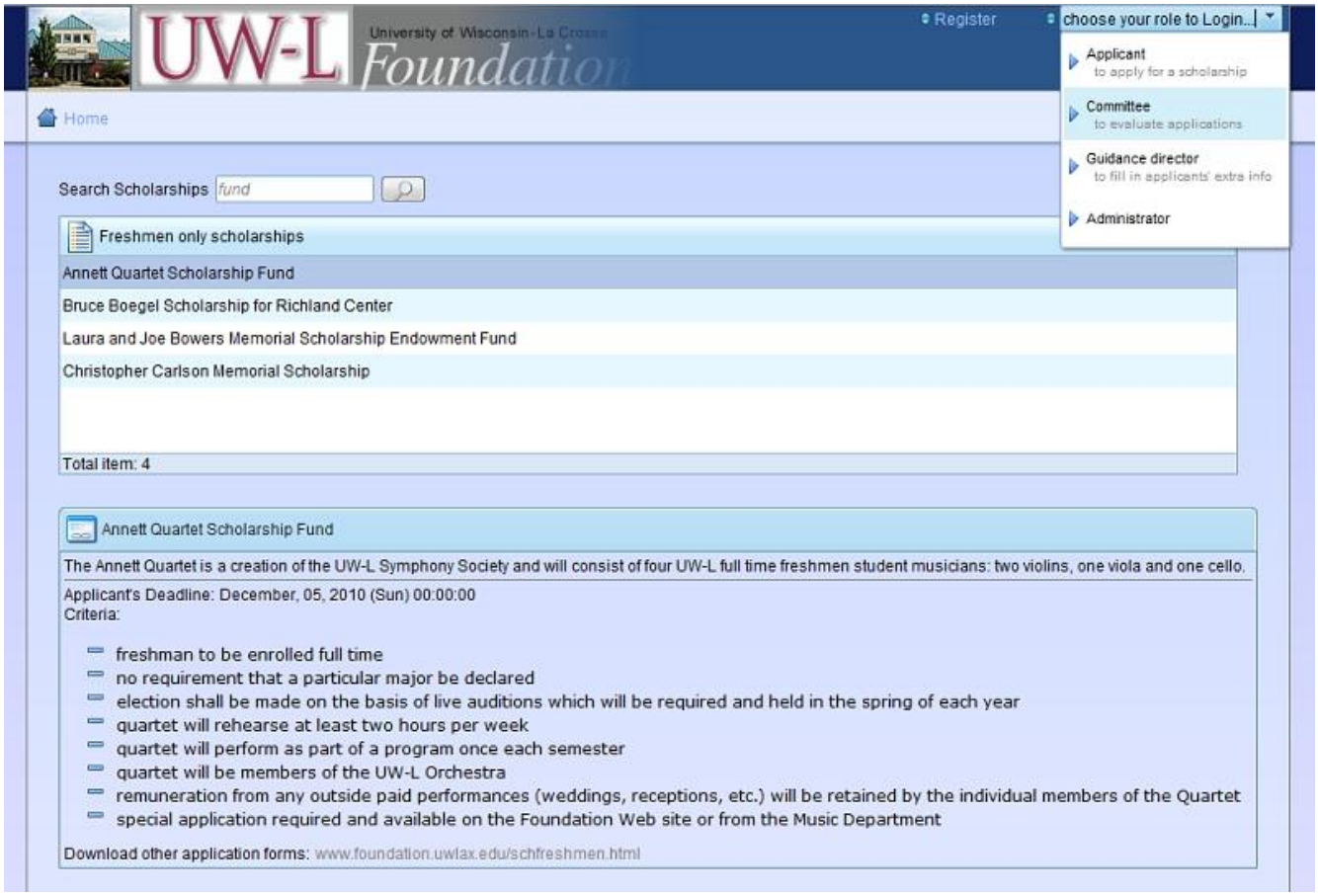
[7] Linda H. Rosenberg, *Software Re-engineering*, Goddard Space Flight Center, NASA, 1996.

[8] Markus Stäuble, Hans-Jürgen Schumacher, *ZK Developer's Guide*, Packt Publishing Ltd., 2008.

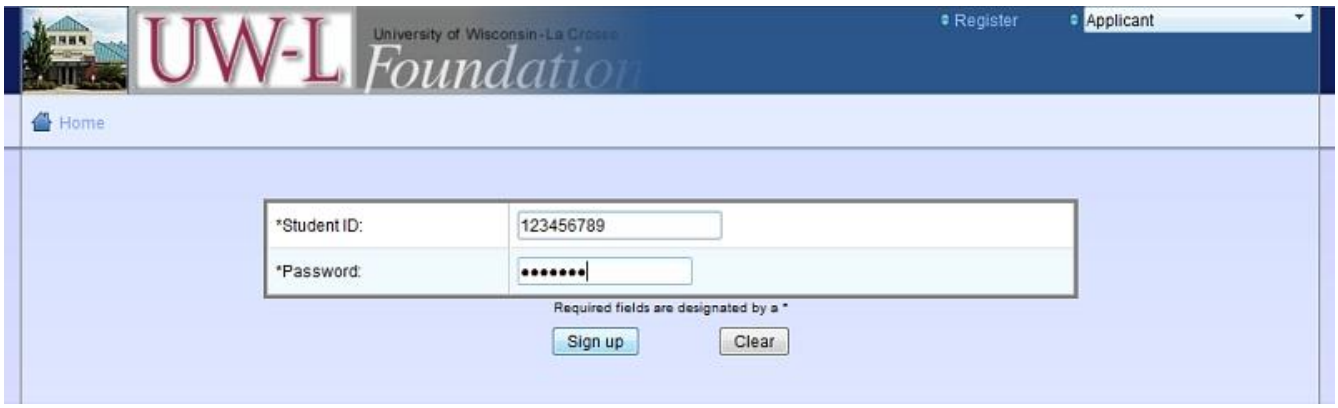
[9] *Oracle Database Express Edition 2 Day Developer Guide, 10g Release 2 (10.2)*, Oracle, 2006.

Appendix

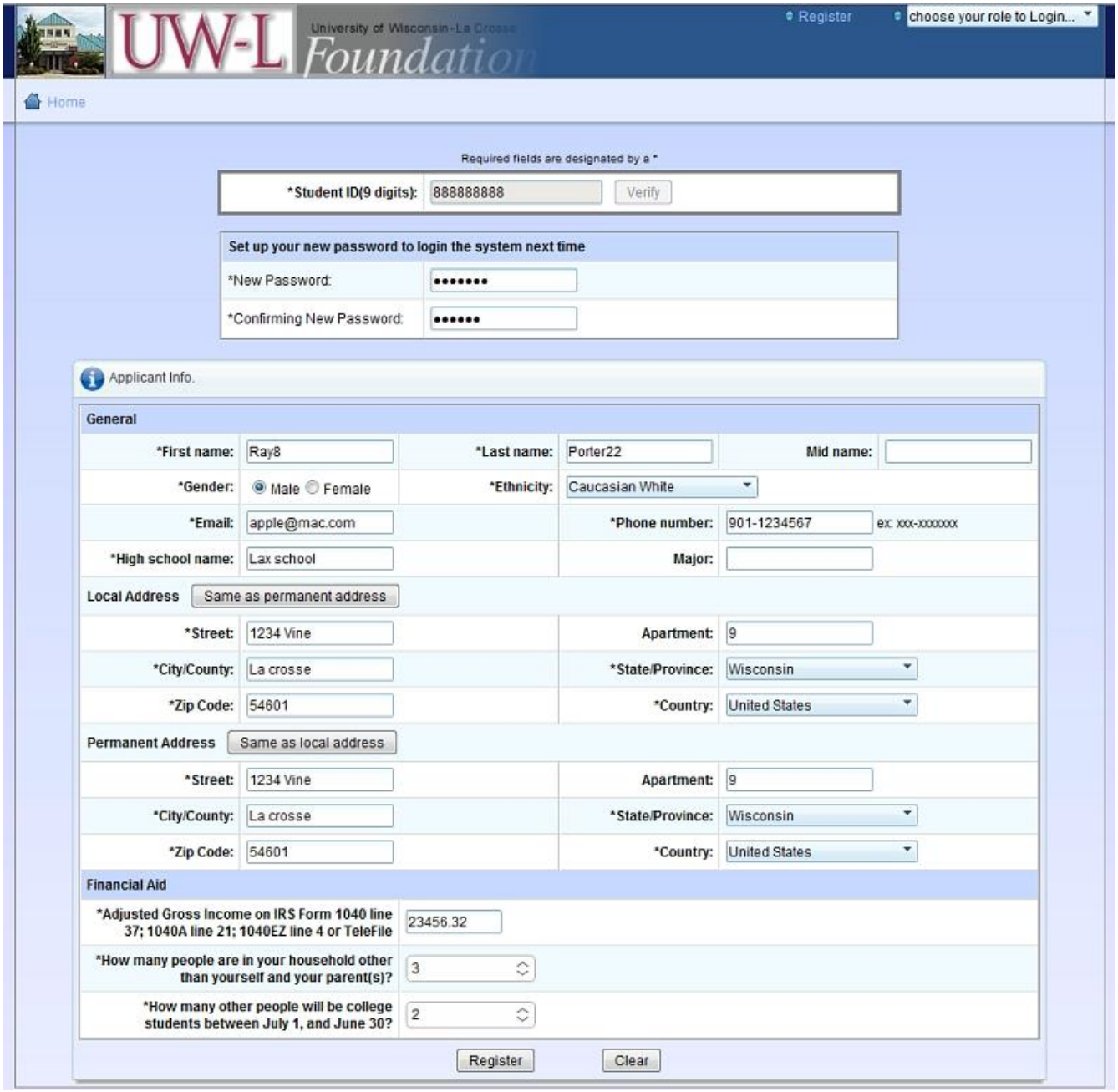
A.1 Screen Shots for sample of Guest functionalities



Home page for every visitor before login to system

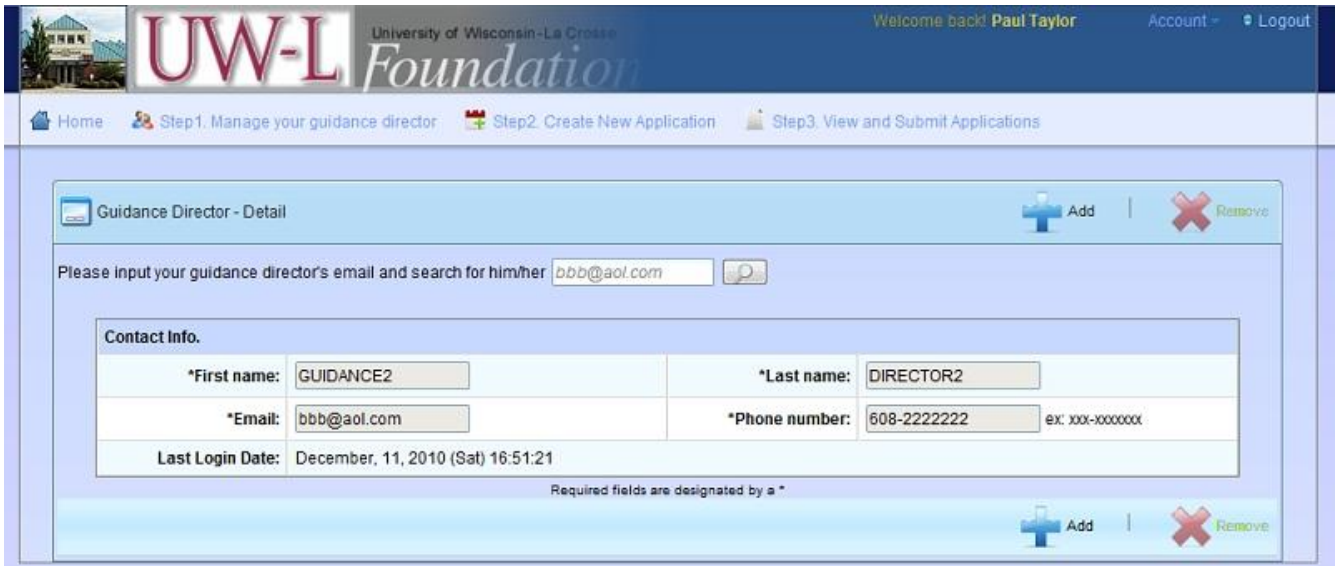


Login page for applicant by Student ID and Password
(For different roles, the identification may be changed)

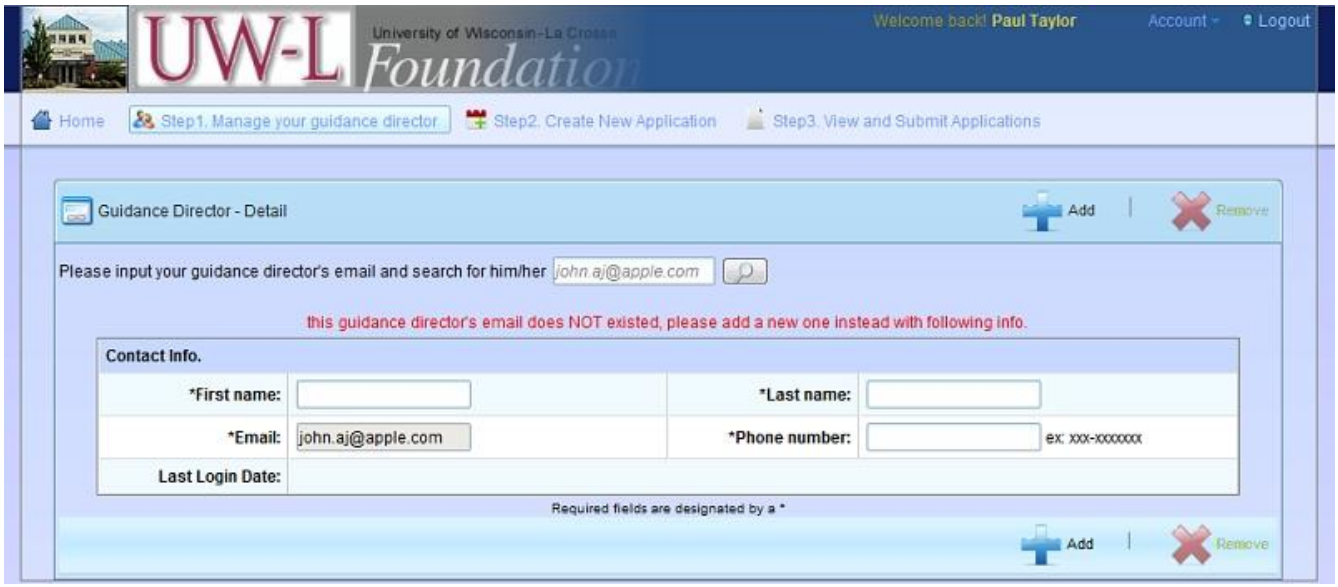


Registration page for students

A.2 Screen Shots for sample of Applicant functionalities

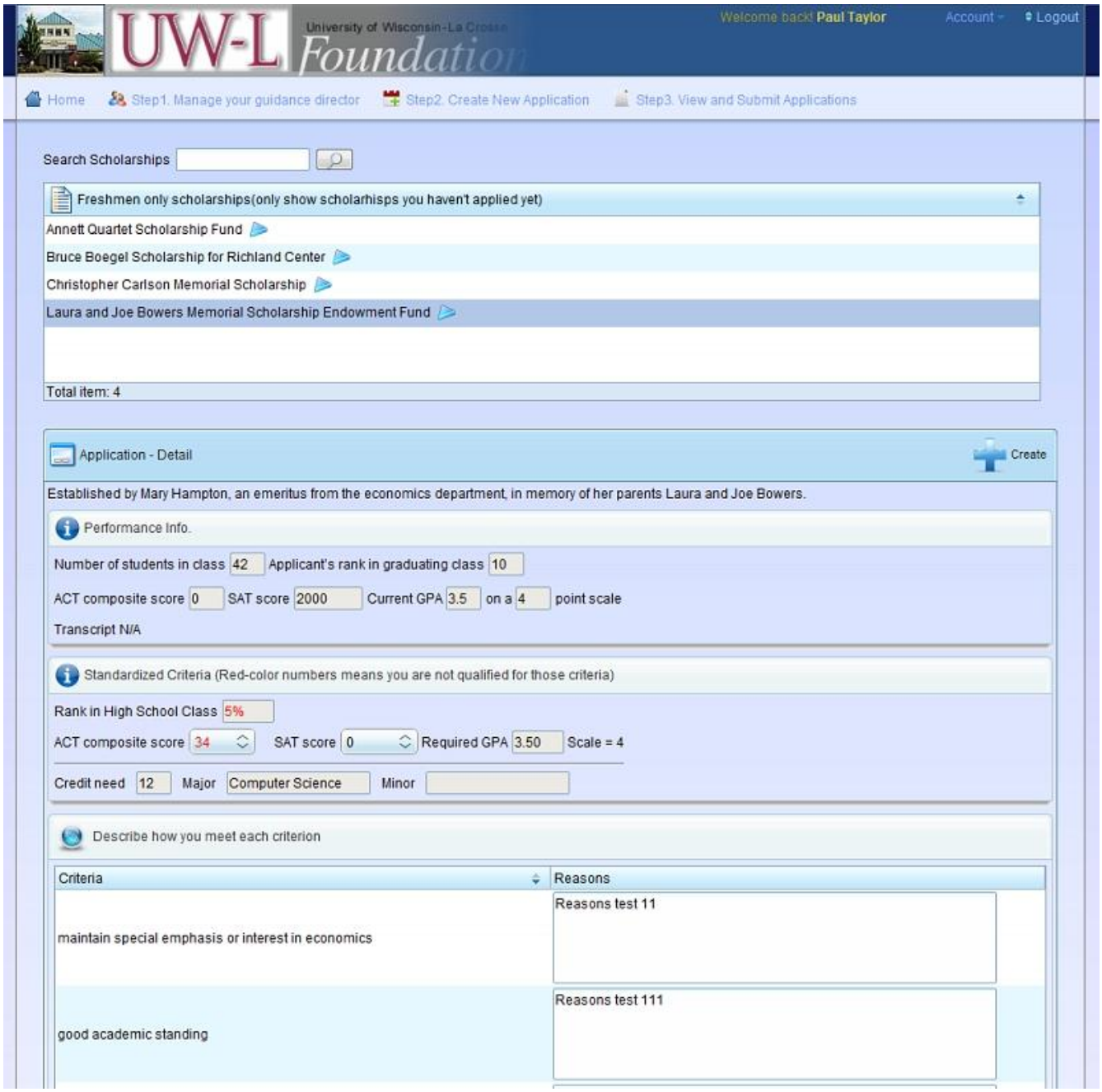


Add an existing guidance director with the applicant

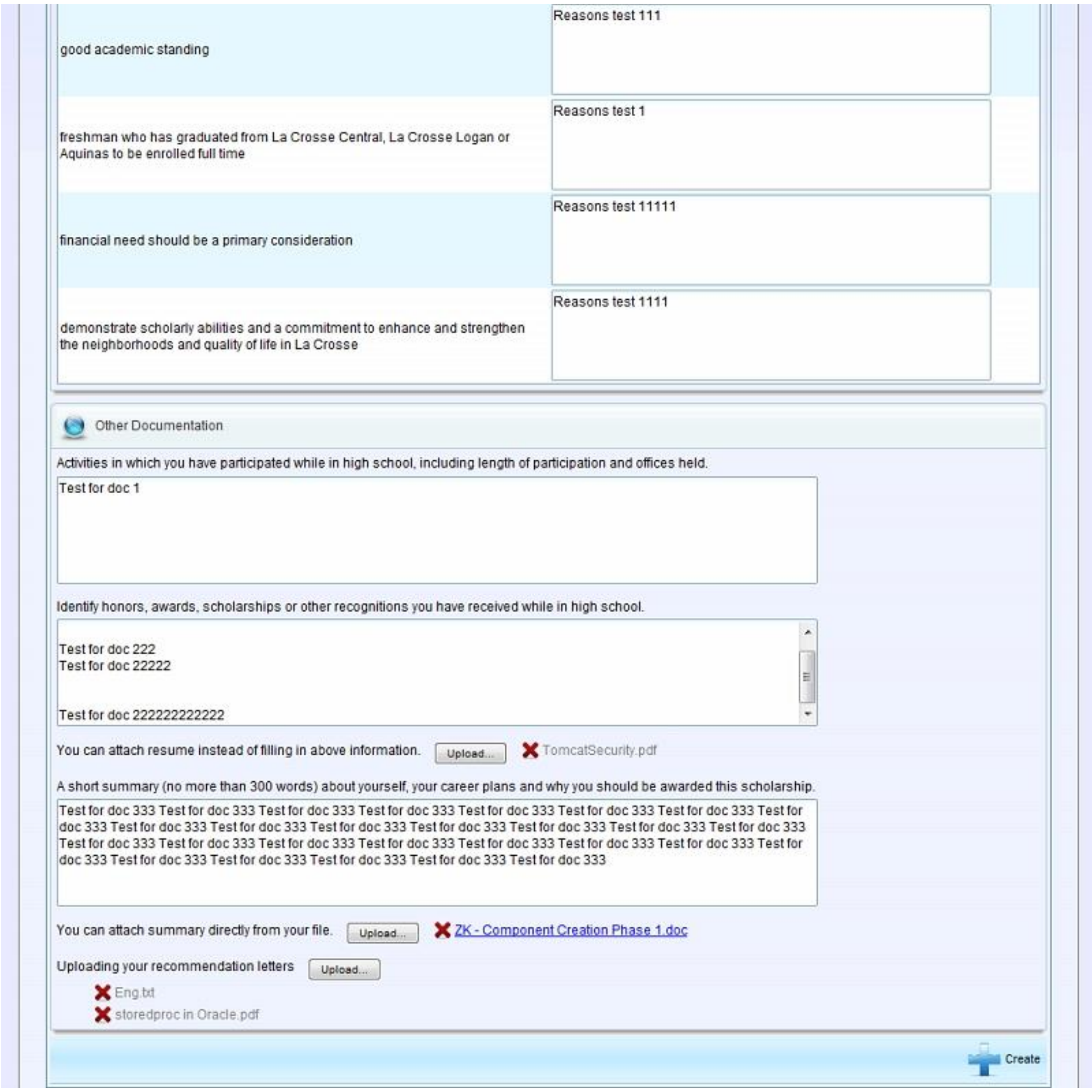


Add a new guidance director with the applicant

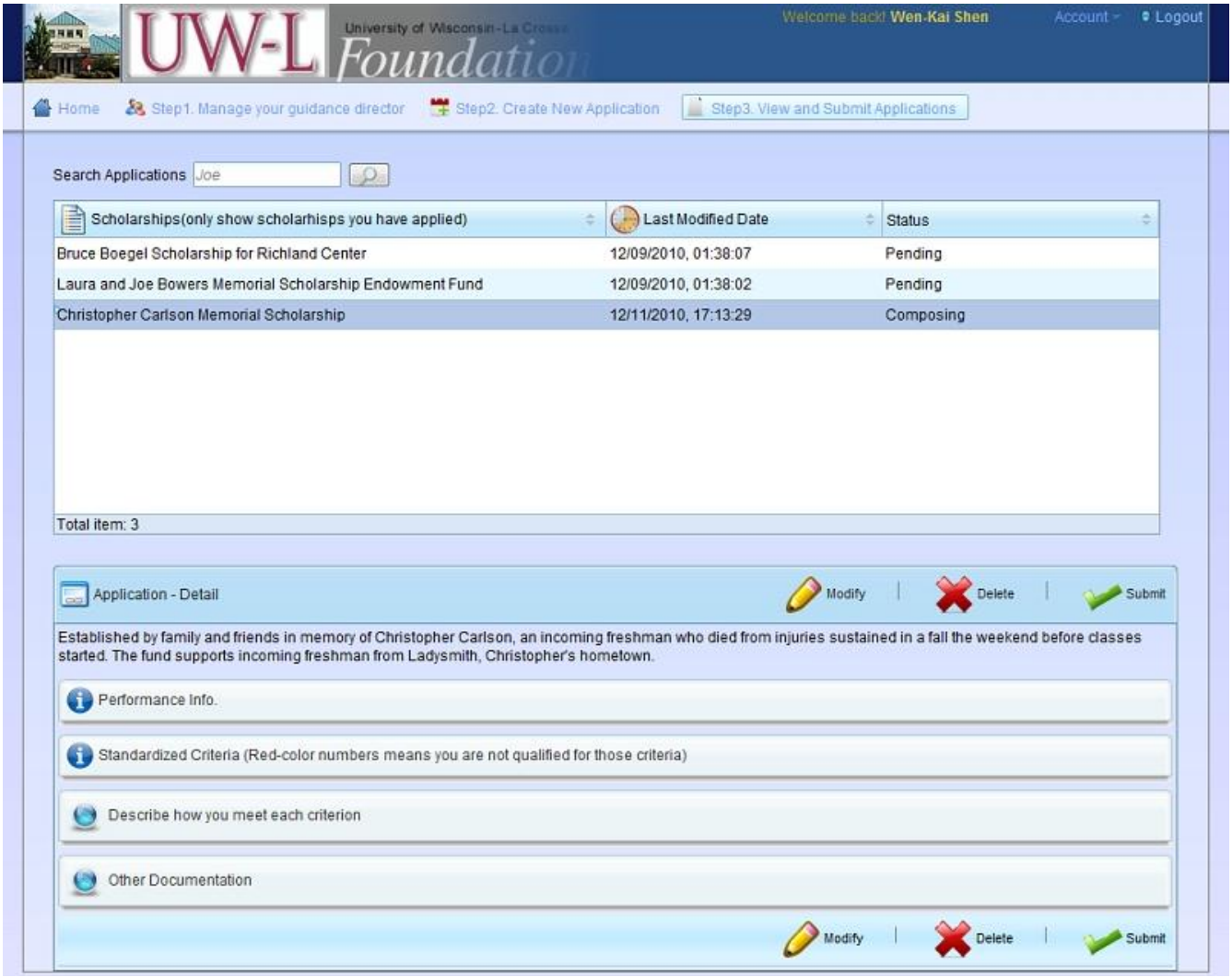
Remove the applicant's guidance director



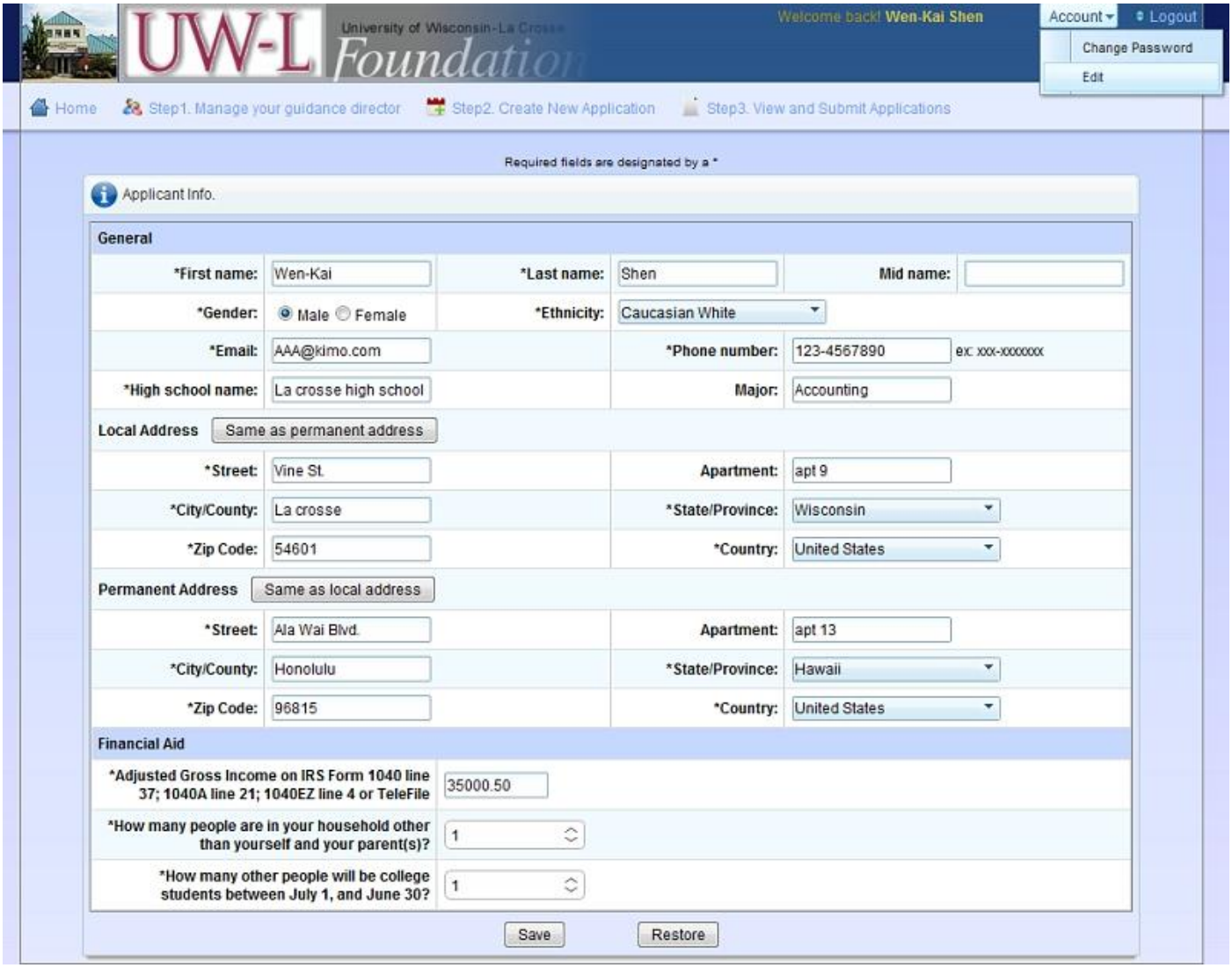
Create a new application part 1



Create a new application part 2

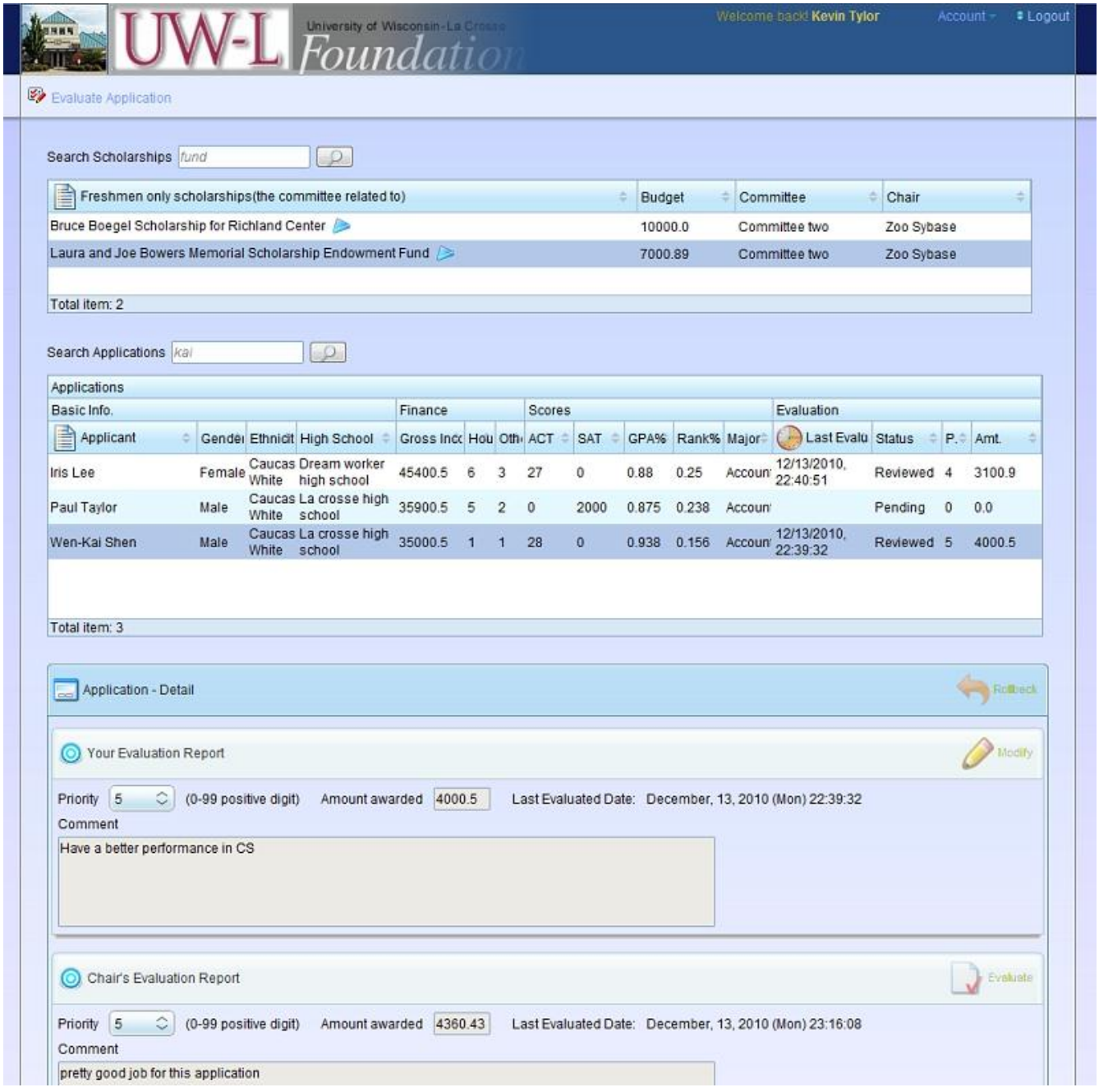


Modify, Delete and Submit an application

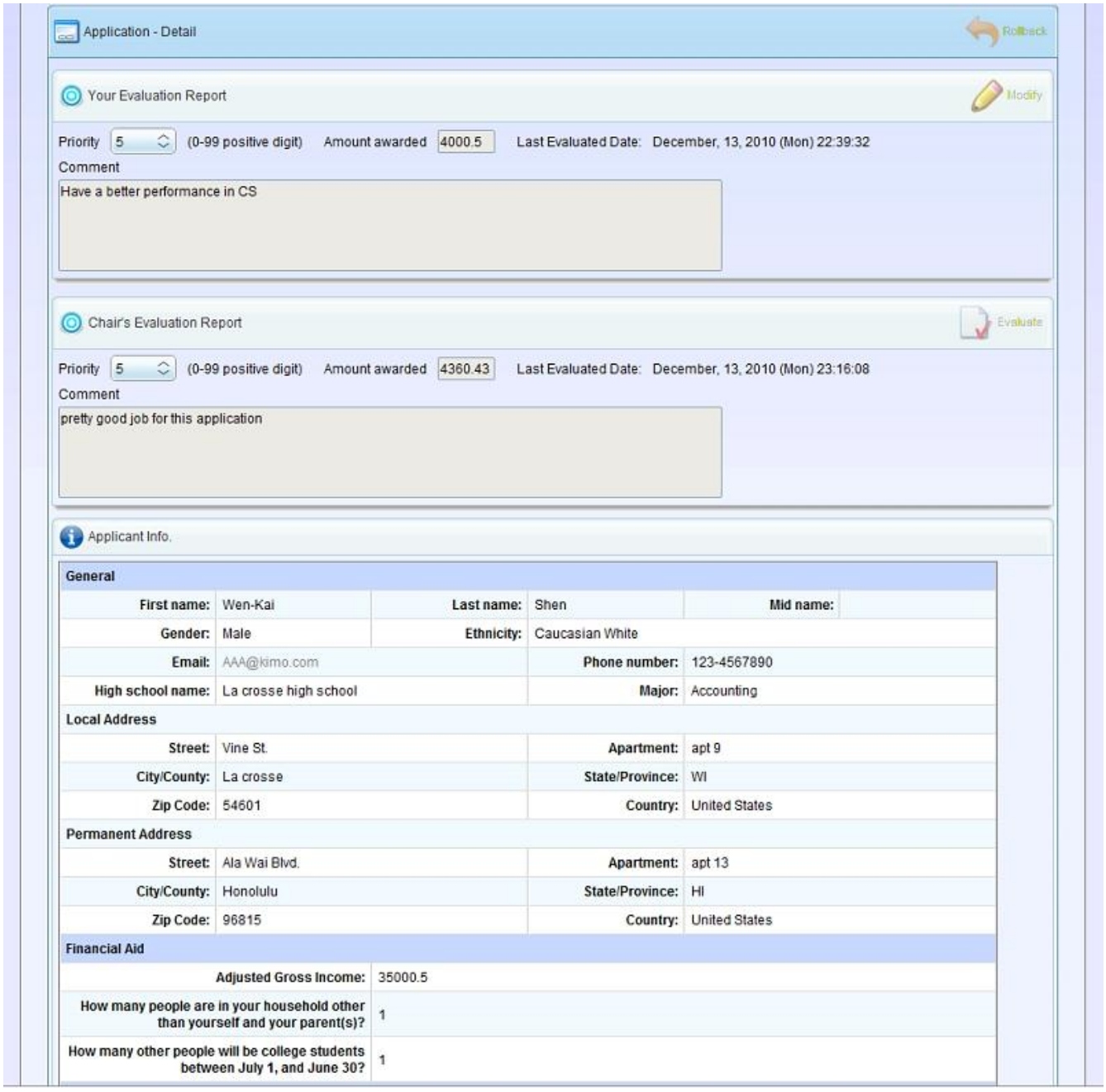


Edit personal information

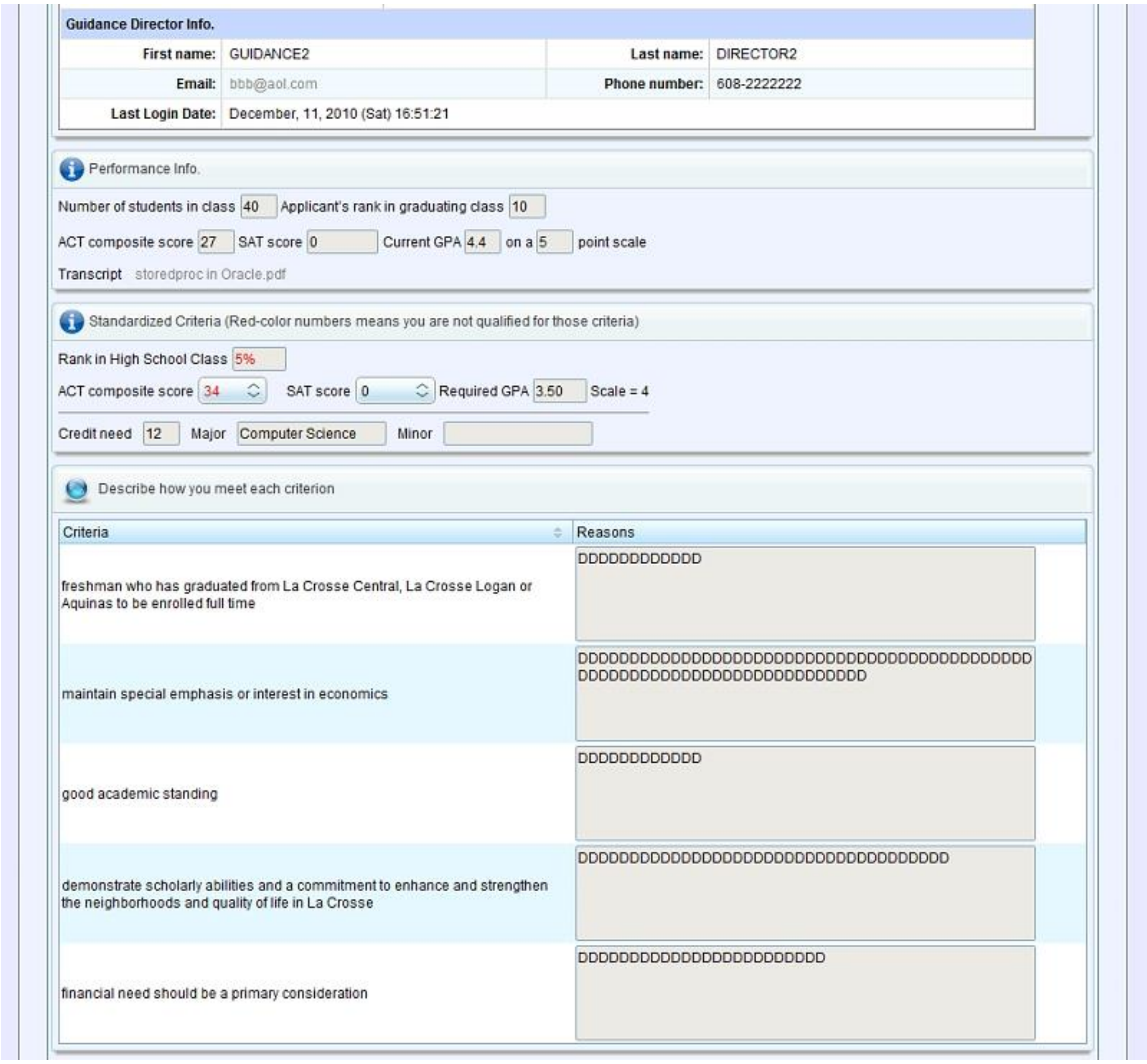
A.3 Screen Shots for sample of Committee functionalities



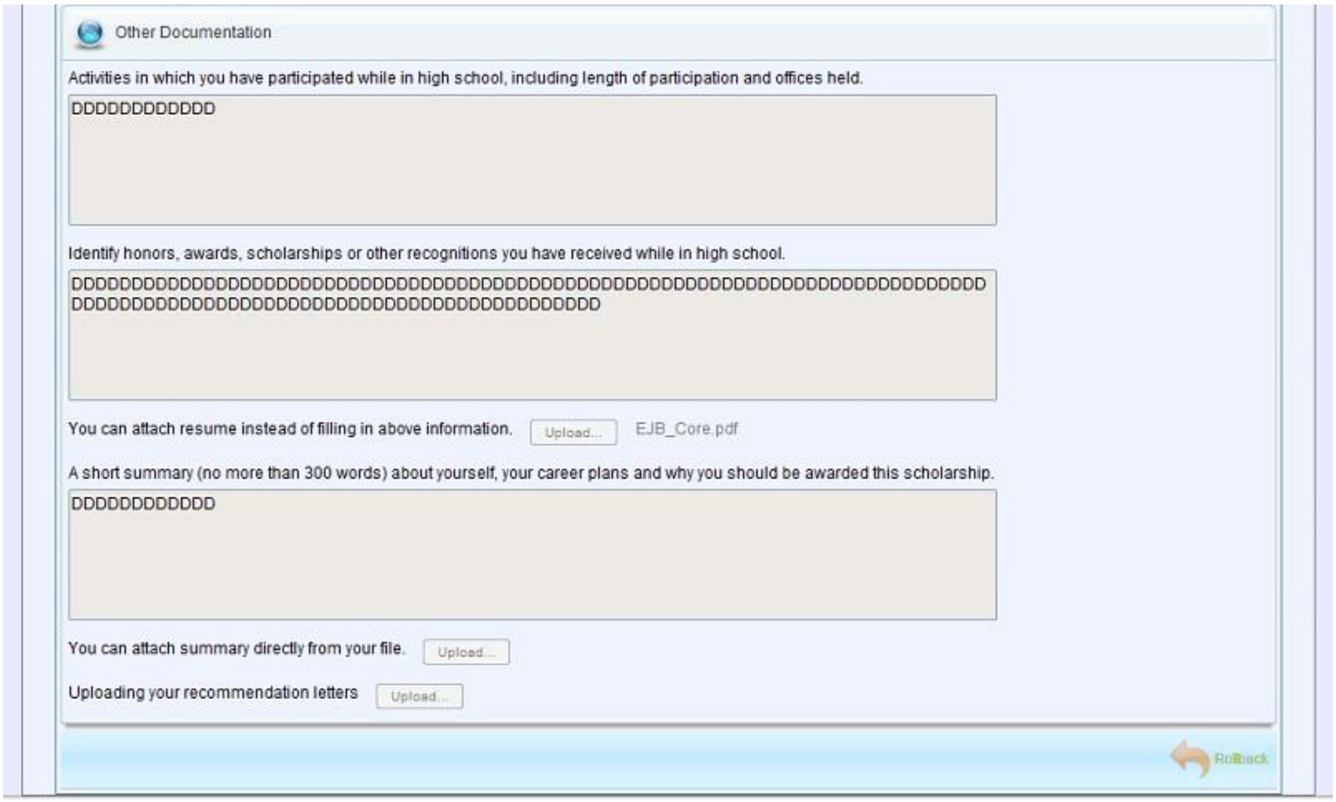
Review applications by Committee Member part 1



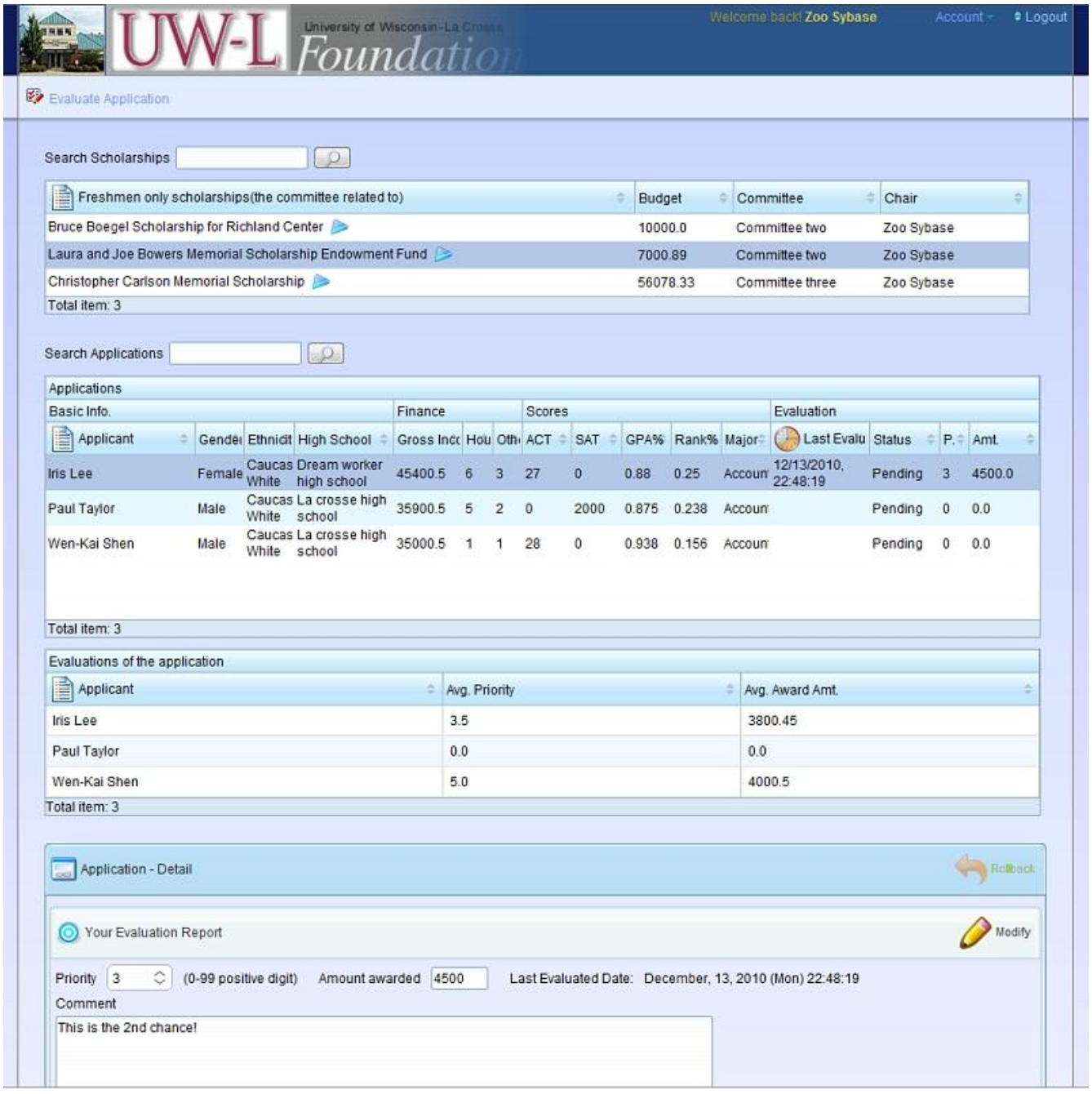
Review applications by Committee Member part 2



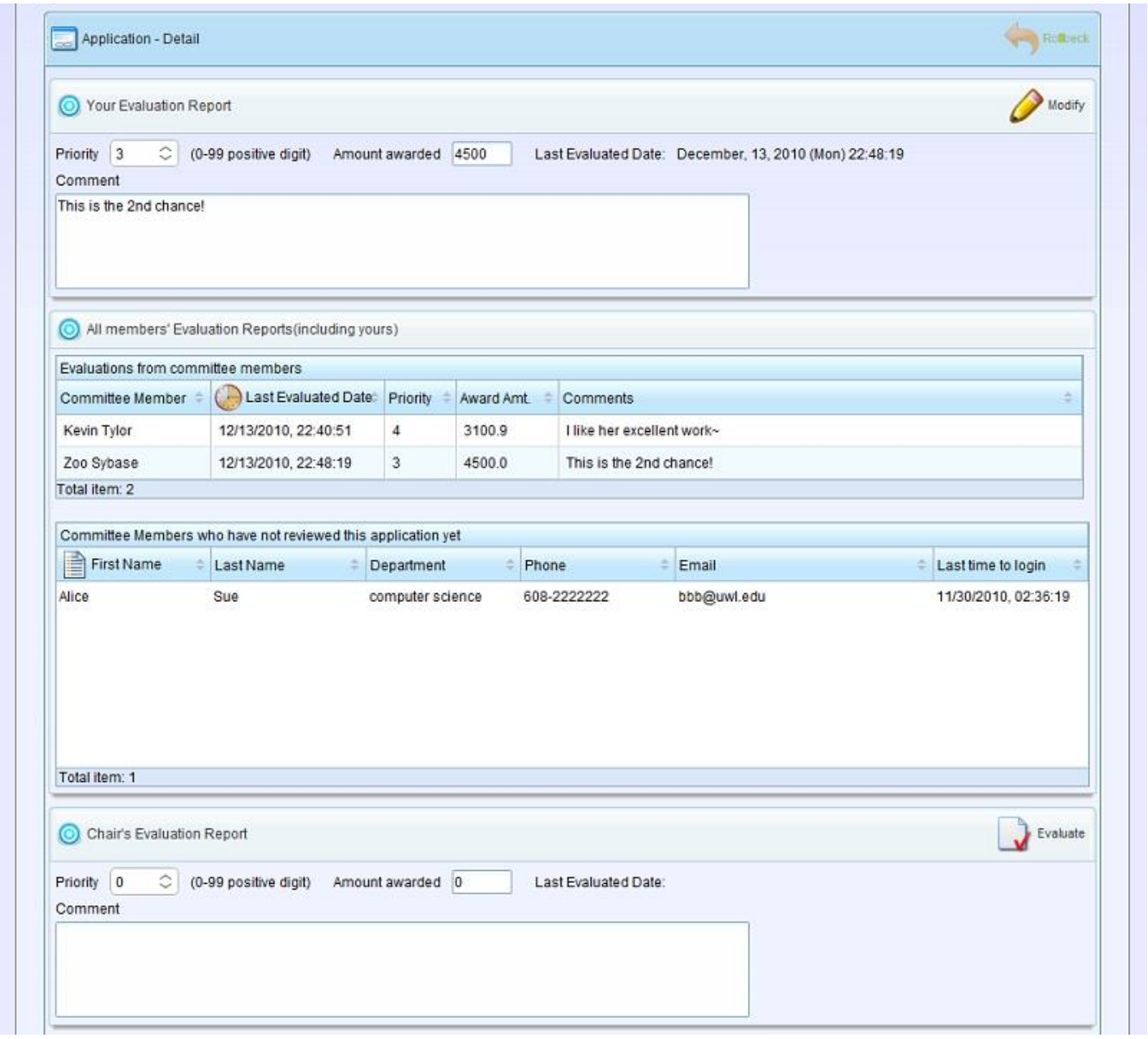
Review applications by Committee Member part 3



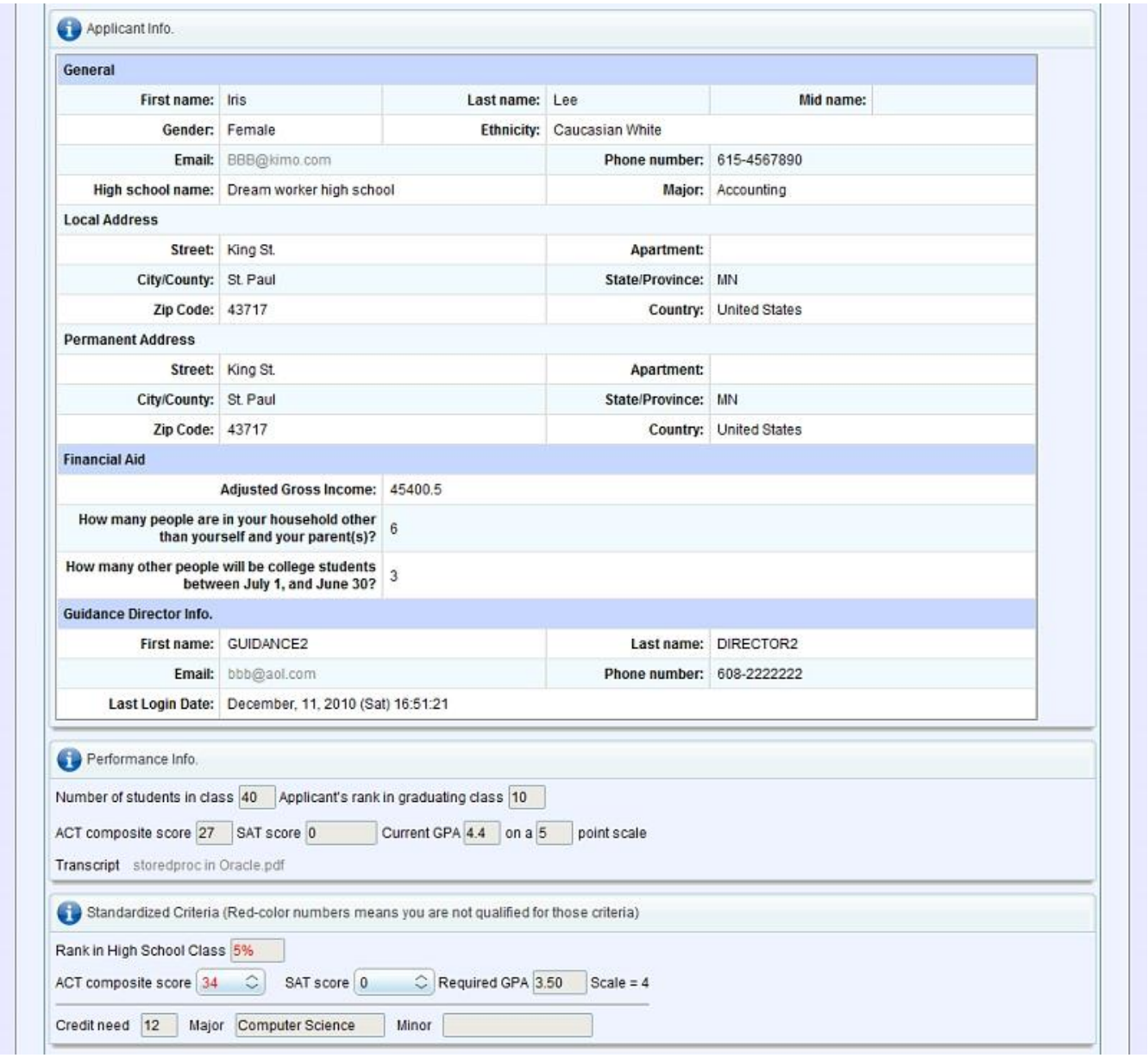
Review applications by Committee Member part 4



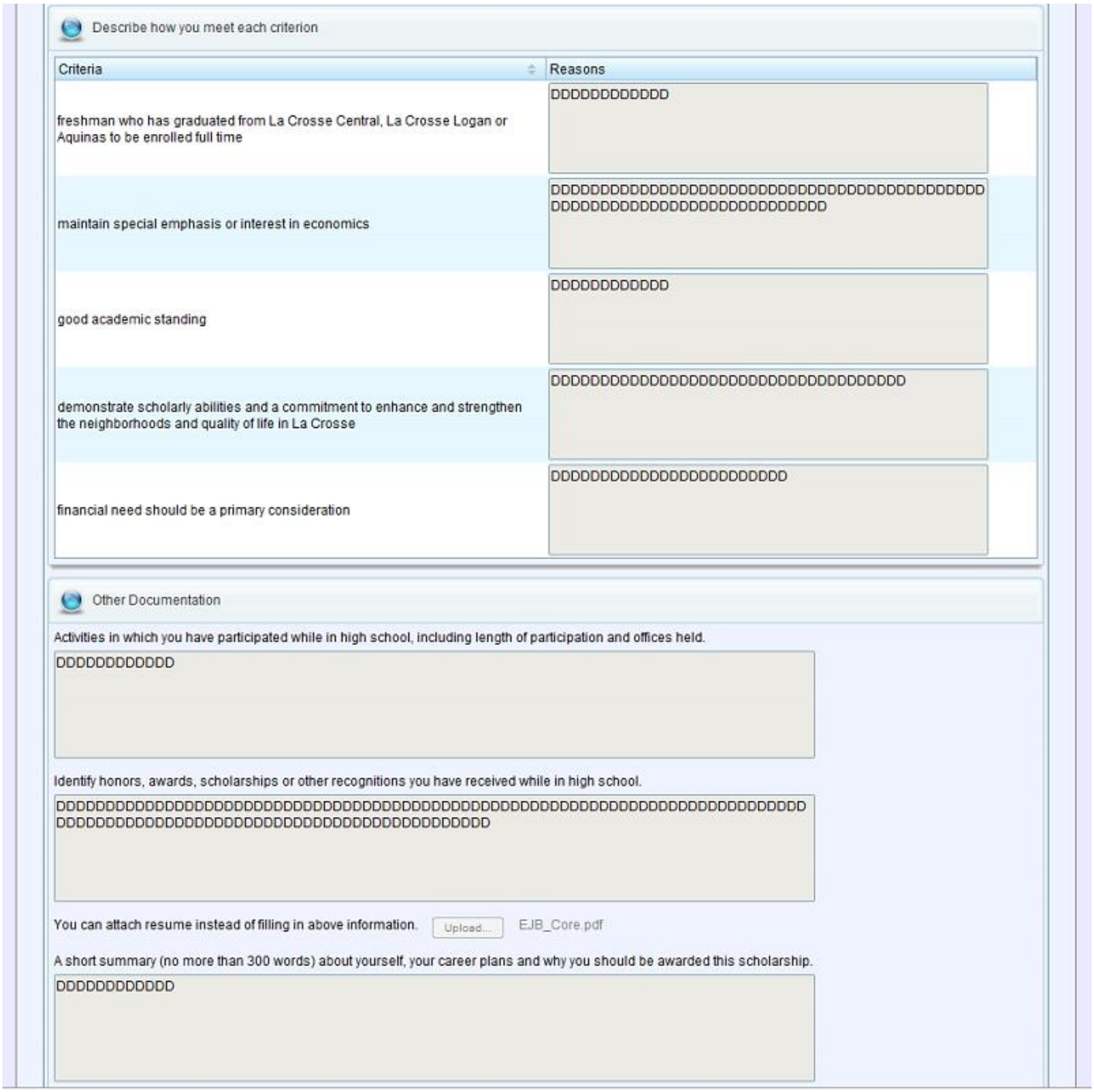
Evaluate applications by Committee Chair part 1



Evaluate applications by Committee Chair part 2



Evaluate applications by Committee Chair part 3



Describe how you meet each criterion

Criteria	Reasons
freshman who has graduated from La Crosse Central, La Crosse Logan or Aquinas to be enrolled full time	DDDDDDDDDDDD
maintain special emphasis or interest in economics	DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD DDDDDDDDDDDDDDDDDDDDDDDDDDDD
good academic standing	DDDDDDDDDDDD
demonstrate scholarly abilities and a commitment to enhance and strengthen the neighborhoods and quality of life in La Crosse	DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
financial need should be a primary consideration	DDDDDDDDDDDDDDDDDDDDDDDD

Other Documentation

Activities in which you have participated while in high school, including length of participation and offices held.

DDDDDDDDDDDD

Identify honors, awards, scholarships or other recognitions you have received while in high school.

DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD

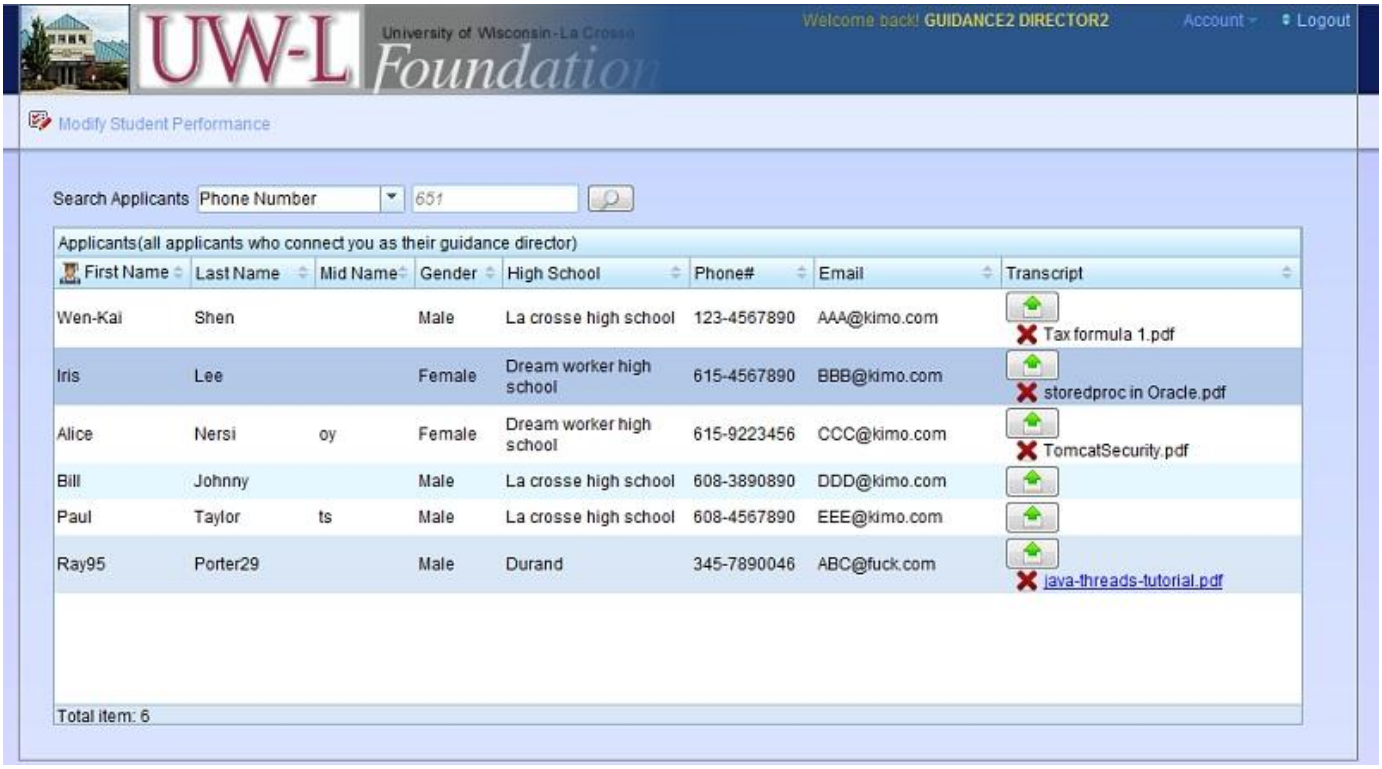
You can attach resume instead of filling in above information. Upload... EJB_Core.pdf

A short summary (no more than 300 words) about yourself, your career plans and why you should be awarded this scholarship.

DDDDDDDDDDDD

Evaluate applications by Committee Chair part 4

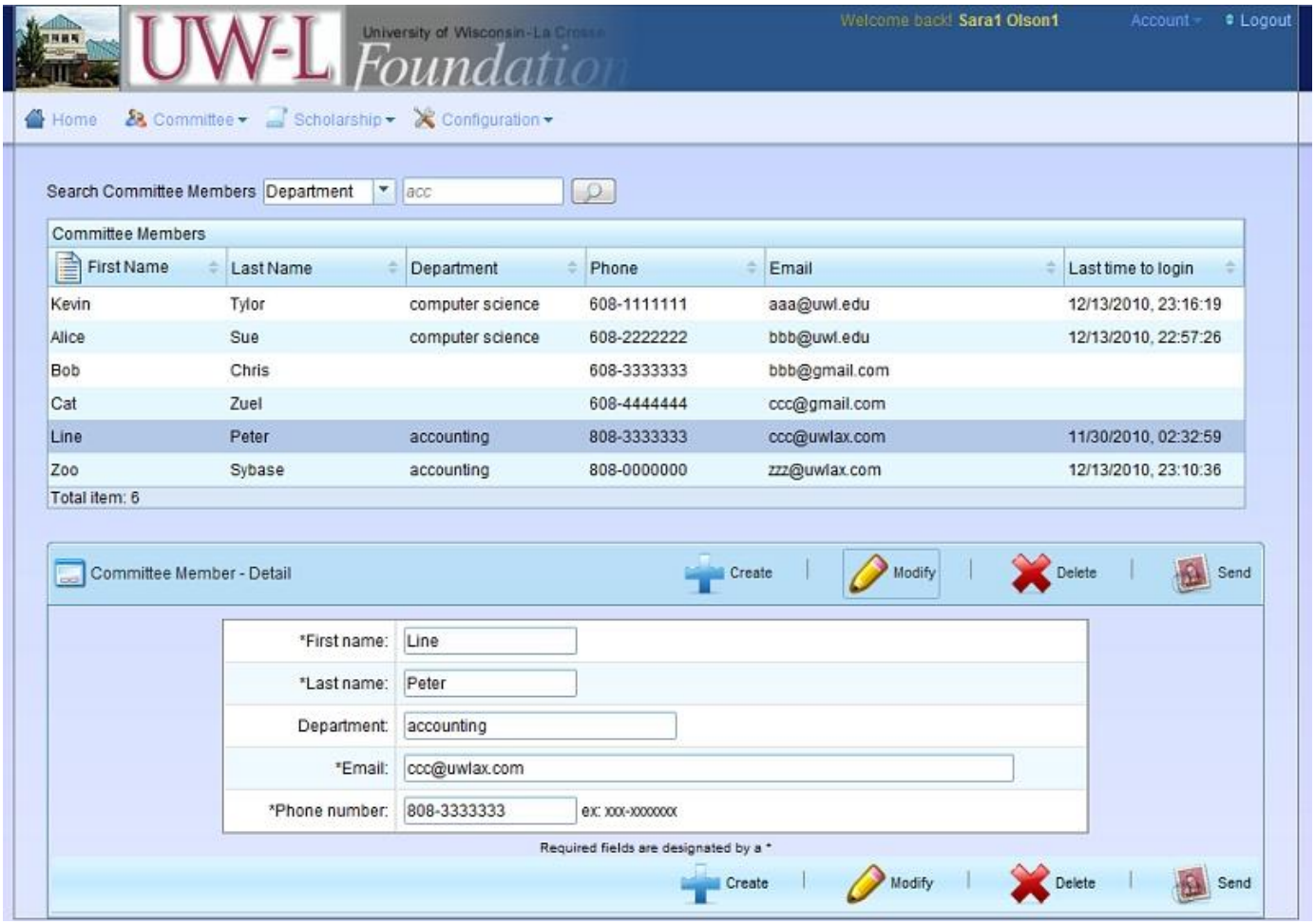
A.4 Screen Shots for sample of Guidance Director functionalities



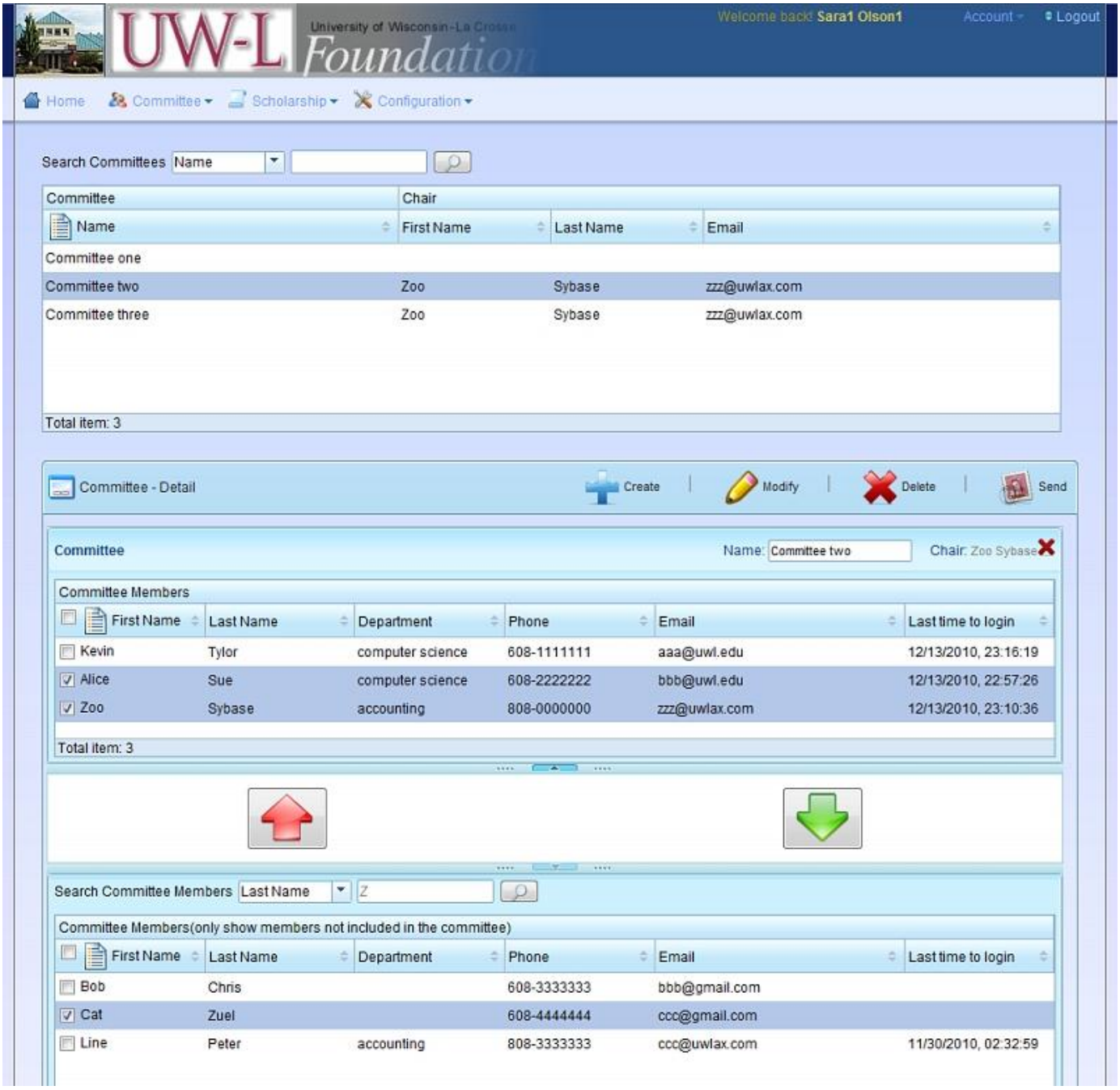
Modify student performance

Edit contact information

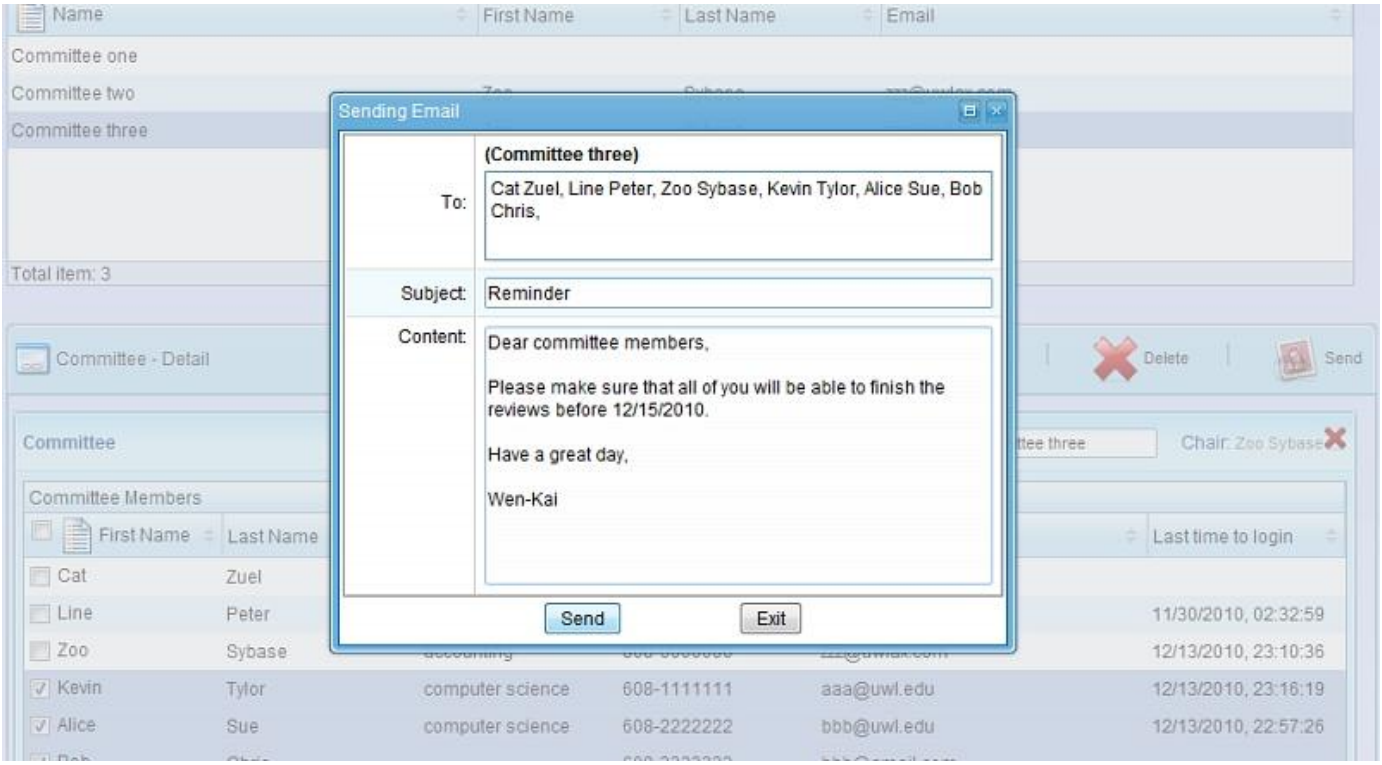
A.5 Screen Shots for sample of Administrator functionalities



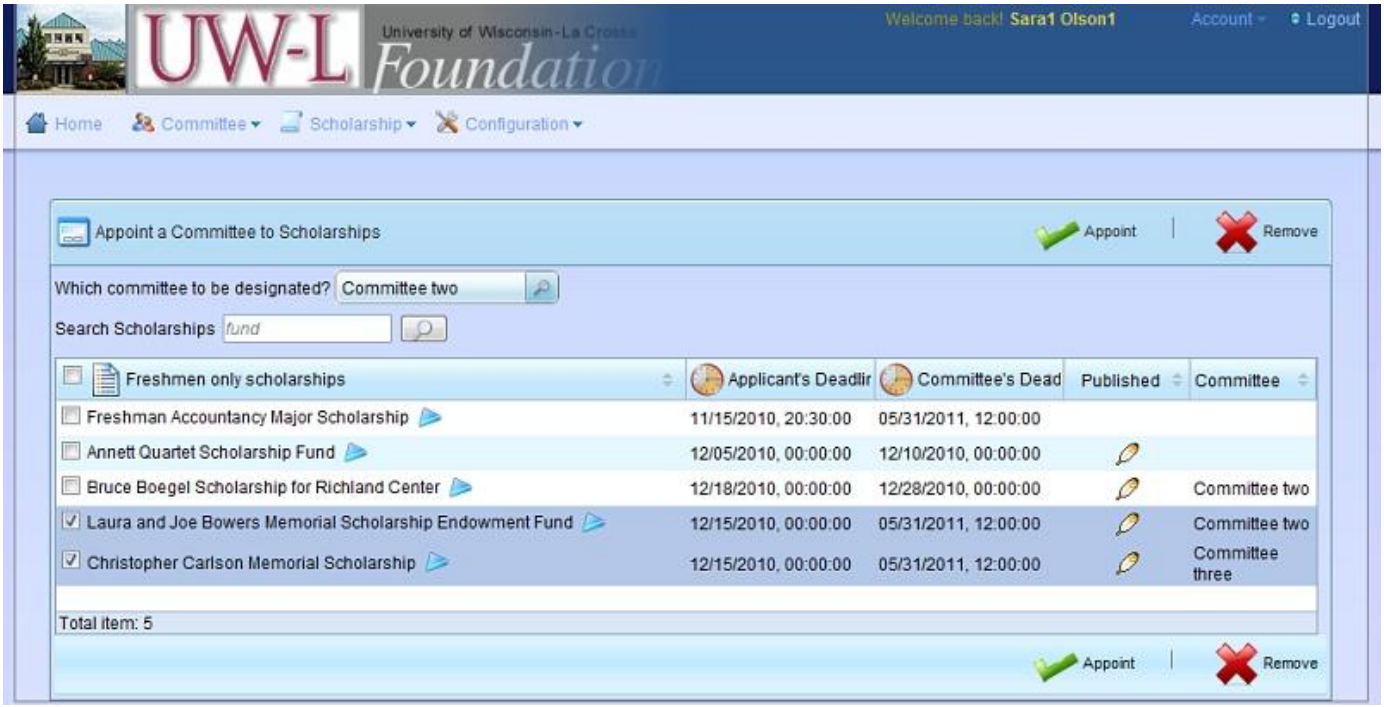
Manage committee members



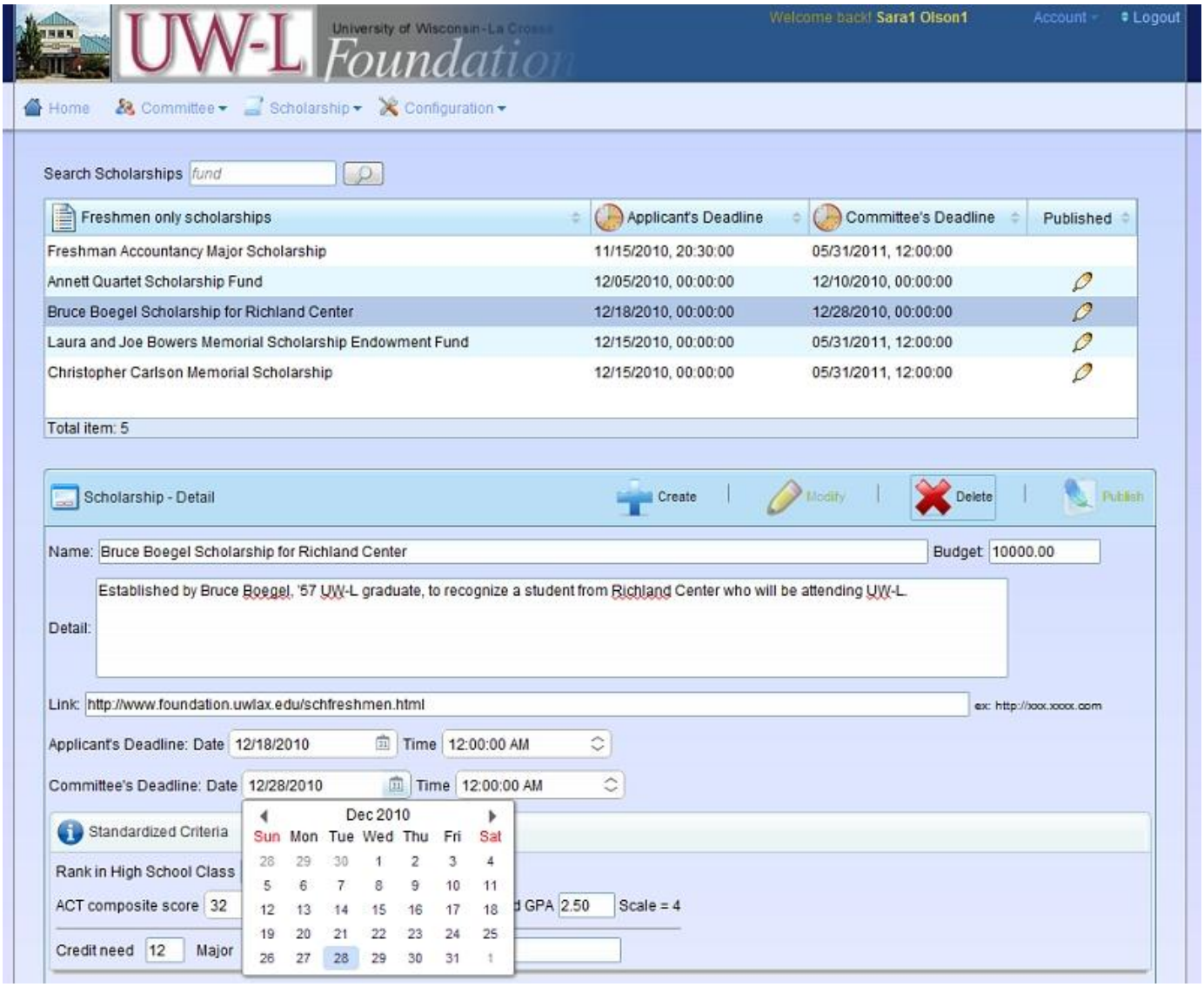
Manage committees



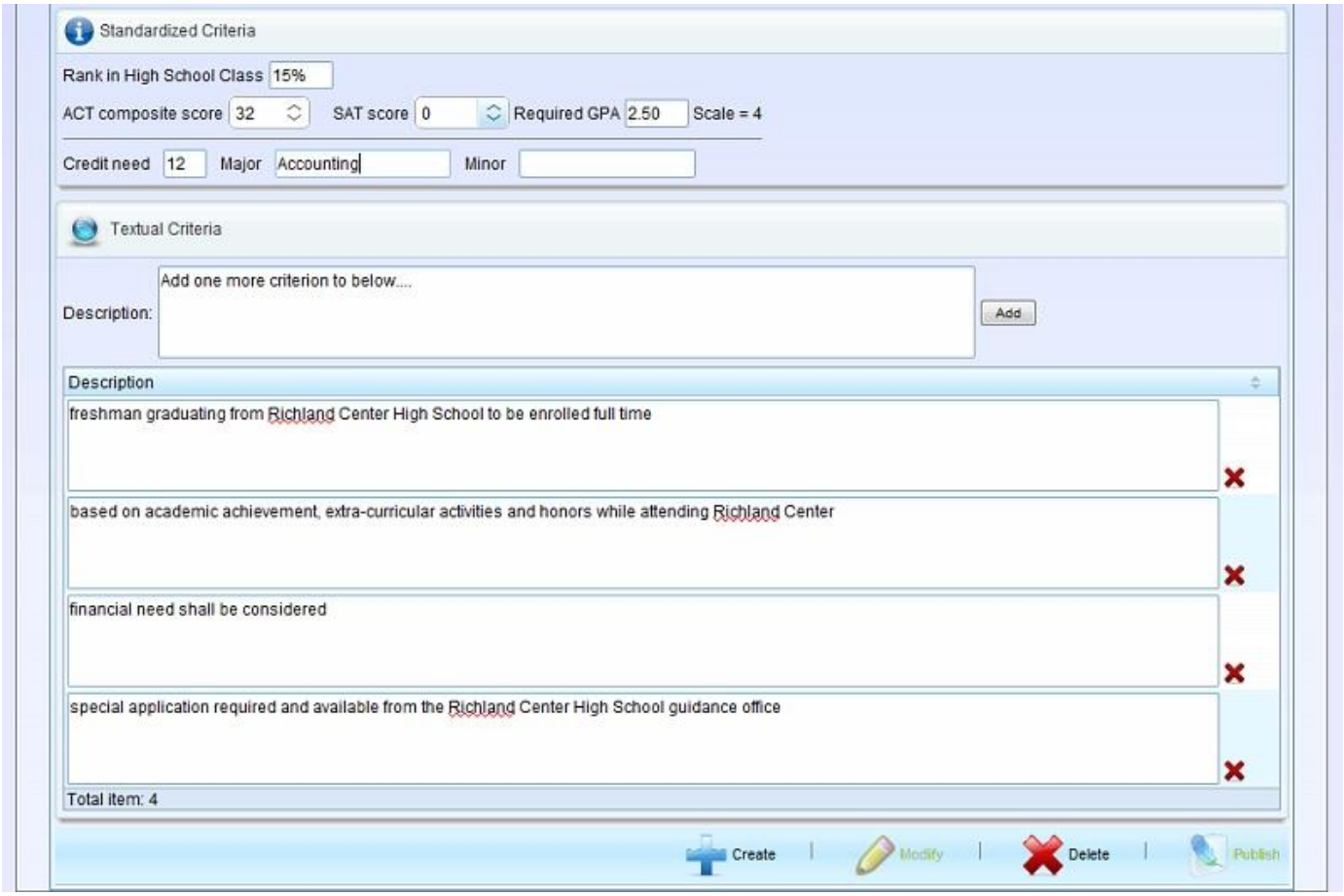
Send reminder to a committee



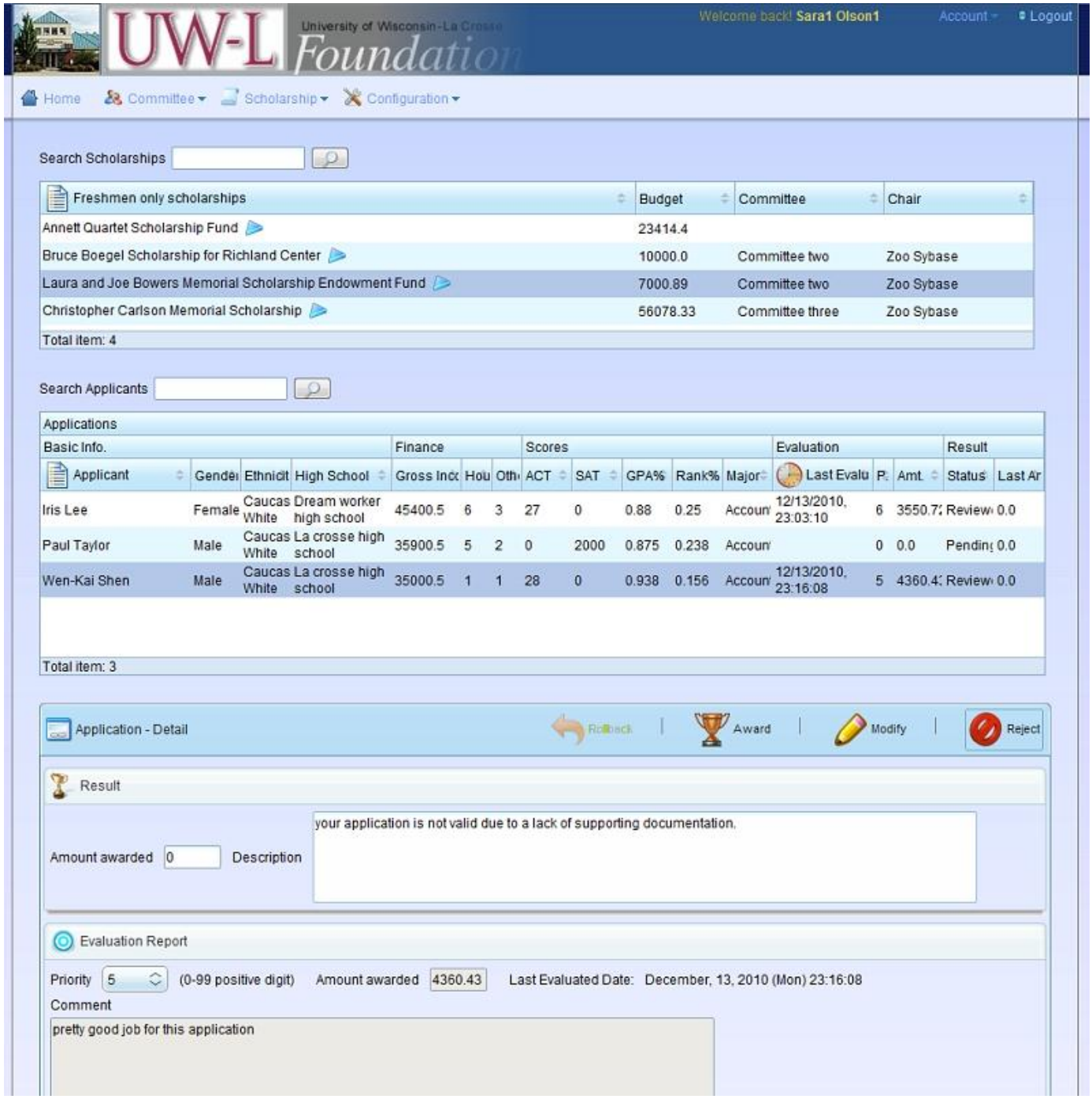
Appoint a committee to scholarships



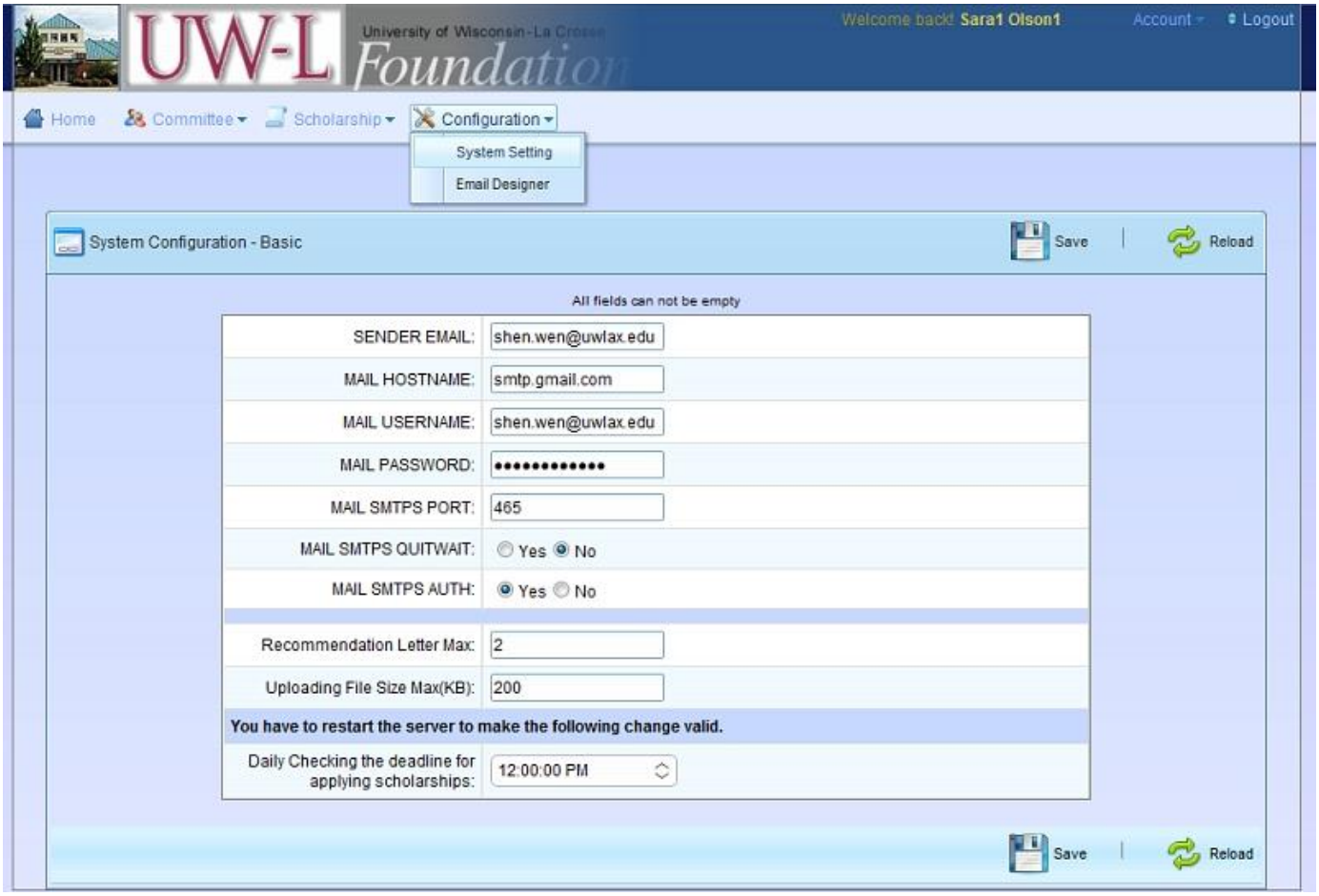
Manage scholarships part 1



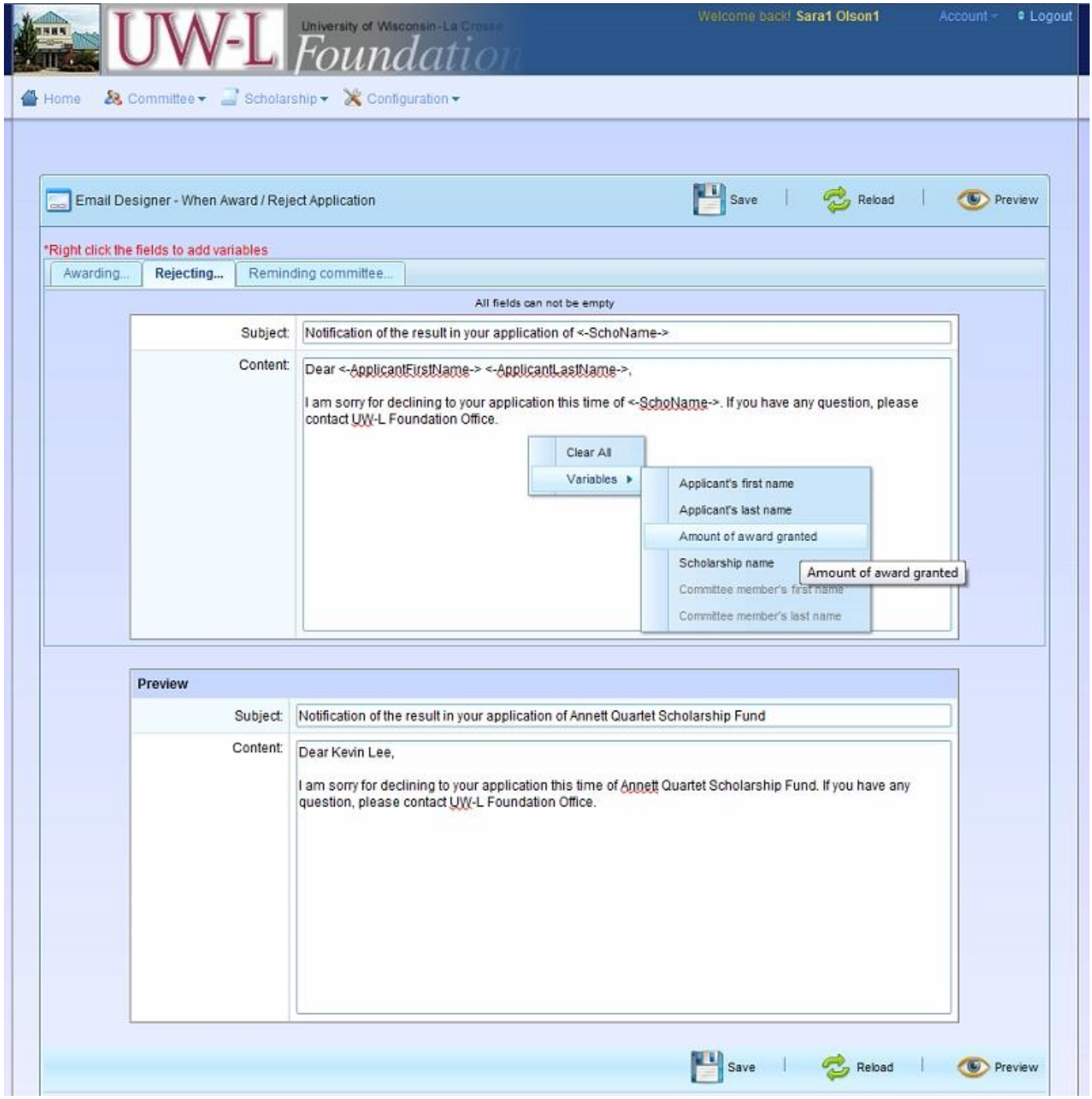
Manage scholarships part 2



Award scholarship



Configure system settings



Email Designer