WEB-BASED STUDENT EVALUATION OF INSTRUCTION

A Thesis
Submitted to
the Department of Computer Science
and the Faculty of the
University of Wisconsin-La Crosse
La Crosse, Wisconsin-54601

by

Radhika Nayani

In Partial Fulfillment of the
Requirements for the Degree
of

Master of Software Engineering

December, 2010
WEB-BASED STUDENT EVALUATION OF INSTRUCTION

By Radhika Nayani

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. David Riley Date
Examination Committee Member

____________________________________ _______________________
Dr. Thomas Gendeau Date
Examination Committee Member
ABSTRACT


Student Evaluation of Instruction (SEI) is a survey instrument for collecting student feedback on the quality of instruction. In University of Wisconsin-La Crosse (hereafter referred to as the “UWL”), feedback from students is usually gathered through paper-based questionnaires. Such surveys consist of a standard set of questions addressing general aspects of teaching that involves both instructor and course. Student evaluation of teaching is important to the university for two primary reasons. First, student evaluations provide data used for administrative decisions such as tenure, promotion, and salary increases for faculty. Second, teaching evaluations provide feedback to help faculty improve teaching performance. There are several advantages of using an online system when compared to paper based evaluation; some of the important advantages include increased efficiency in data collection process, flexibility of questionnaire design, eco-friendly avoidance of paper wastage and immediate availability of data for analysis and reporting. The current process involves the great deal of labor to handle the student’s anonymity.

The project described in this report is called "Online Student Evaluation of Instruction" (hereafter referred to as the “OSEI”); it is a web-based system developed to collect teaching evaluation data for summative and formative purposes. The evaluations will be administered and collected in a manner that assures security. It addition, reports will be delivered in a timely fashion to administrators in a manner that is easy to understand and interpret. The OSEI system has been designed to allow flexibility in changing the questions, reports and evaluations.
I would like to express my sincere thanks to my project advisor Dr. Kasi Periyasamy for his valuable guidance. He was always available for reviewing my work throughout the process. Without his suggestions and help, this project would not have been possible. I would like to thank Dr. Robert Hoar (Professor of Mathematics & Faculty Assistant to the Provost) and Becky Yoshizumy (Academic Department Associate) for giving me more information for reports. I would like to express my gratitude to the Computer Science Department and the University of Wisconsin-La Crosse for providing the required information for my project. Finally, I would like to thank my husband and daughter, Ravindra and Risheeta, for their love, support and patience during the past three or so years it has taken me to graduate. I would like to thank my parents for their unending love and support.
Table of Contents

ABSTRACT .................................................................................................................................................. III

ACKNOWLEDGEMENTS ............................................................................................................................ IV

LIST OF FIGURES ........................................................................................................................................ VI

GLOSSARY ................................................................................................................................................... VII

1. BACKGROUND INFORMATION ............................................................................................................. 1
   1.1 CURRENT SYSTEM ................................................................................................................................. 1
   1.2 DISADVANTAGES OF THE CURRENT SYSTEM .................................................................................. 1
   1.3 PROPOSED SYSTEM .............................................................................................................................. 2

2. SOFTWARE LIFE CYCLE MODEL FOR OSEI ......................................................................................... 3

3. DEVELOPMENT OF THE OSEI .............................................................................................................. 5
   3.1 REQUIREMENT ANALYSIS ...................................................................................................................... 5
   3.2 REQUIREMENTS DEVELOPMENT ......................................................................................................... 5
   3.3 OSEI DESIGN AND IMPLEMENTATION ............................................................................................. 9
       3.3.1 TECHNOLOGIES USED IN SYSTEM DEVELOPMENT .................................................................. 11
       3.3.2 USER INTERFACE DESIGN ......................................................................................................... 13
       3.3.3 DATABASE DESIGN ....................................................................................................................... 14
       3.3.4 DETAILED ARCHITECTURE OF OSEI ....................................................................................... 21
       3.3.5 CHALLENGES FACED IN THE DEVELOPMENT OF OSEI ......................................................... 24
       3.3.6 TESTING THE OSEI SYSTEM ....................................................................................................... 26

4. LIMITATIONS ........................................................................................................................................... 29

5. CONTINUING WORK ............................................................................................................................. 30

6. CONCLUSION ......................................................................................................................................... 31

7. BIBLIOGRAPHY ..................................................................................................................................... 31

APPENDIX A: SAMPLE OSEI CLASS DIAGRAM ..................................................................................... 33

APPENDIX B: SAMPLE OSEI SCREENSHOTS ......................................................................................... 34
LIST OF FIGURES

Figure 1 : Incremental Life Cycle Model ............................................................... 4
Figure 2 : Use Case Diagram for OSEI ................................................................. 10
Figure 3 : Database Diagram for the OSEI .......................................................... 15
Figure 4 : OSEI Architectural Diagram ............................................................... 21
Figure 5 : OSEI Class Diagram ........................................................................... 33
Figure 6 : Administrator Login .......................................................................... 34
Figure 7 : Administrator Account ....................................................................... 35
Figure 8 : New Administrator Account ............................................................... 35
Figure 9 : Course Screen .................................................................................... 36
Figure 10 : New Course Screen .......................................................................... 37
Figure 11 : Instructor Screen ........................................................................... 38
Figure 12 : Modify Course Screen ....................................................................... 38
Figure 13 : Mandatory Questions Screen ............................................................ 39
Figure 14 : New Mandatory Question Screen ................................................... 40
Figure 15 : Optional Questions Screen ............................................................... 41
Figure 16 : New Optional Question Screen ....................................................... 41
Figure 17 : Comments Screen ............................................................................ 42
Figure 18 : New Comments Screen .................................................................... 42
Figure 19 : Students Screen ............................................................................. 43
Figure 20 : New Student Screen ........................................................................ 44
Figure 21 : Student Course Screen .................................................................... 44
Figure 22 : New Student Screen ........................................................................ 45
Figure 23 : Department Screen ......................................................................... 46
Figure 24 : New Department Screen .................................................................. 46
Figure 25 : Main Reports Screen ....................................................................... 47
Figure 26 : Department Report .......................................................................... 47
Figure 27 : Question Report ................................................................................ 48
Figure 28 : Course Report .................................................................................. 49
Figure 29 : Instructor Report .............................................................................. 49
Figure 30 : Evaluation Form Screen 1 ............................................................... 50
Figure 31 : Evaluation Form Screen2 ................................................................. 50
Figure 32 : Evaluation Form Screen3 ................................................................. 51
Figure 33 : Evaluation Form Screen4 ................................................................. 52
Figure 34 : Evaluation Form Screen5 ................................................................. 52
Figure 35 : Evaluation Form Screen6 ................................................................. 53
GLOSSARY

.NET Framework
The Microsoft .NET Framework allows developers to use common set of skills to rapidly build applications for the web, devices, services and more. It is intended to be used by most new applications created for the Windows platform.

AJAX
AJAX (Asynchronous JavaScript and XML), or Ajax, is a group of interrelated web development techniques used on the client-side to enhance the speed of interactive web applications.

ASP.NET
ASP.NET is a web application framework developed and marketed by Microsoft to allow programmers to build dynamic web sites, web applications and XML web services.

C#
C# is an object-oriented programming language developed by Microsoft as part of the .NET initiative. It is one of the programming languages designed for the Common Language Infrastructure.

CSS
Cascading Style Sheets (CSS) is a style sheet language used to describe the presentation of a document written in a markup language. Its most common application is to style web pages written in HTML and XHTML, but the language can be applied to any kind of XML document.

Entity-Relationship Diagram
Entity-Relationship Diagram is a specialized graphic that illustrates the relationships between entities in a database.
GUI
Graphical User Interface, the presentation layer for users to interact with an application.

HTML
HyperText Markup Language. A markup language designed for creating web pages and other information to view on a web browser.

Java Script
Java script is an object-oriented scripting language that enables programmatic access to computational objects within the host environment.

MySql
MySQL is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases.

Visual Studio
Microsoft Visual Studio is Microsoft's software development product. It provides an integrated development environment allowing programmers to create software applications (console and GUI applications including Windows Forms applications), web sites, web applications, and web services that run on any platforms supported by Microsoft's .NET Framework (for all versions after Visual Studio 6).
1. Background Information

Student Evaluation of Instruction (SEI) is an important source of information to assess the teaching performance. It provides the information assessing an instructor's course content and presentation. Additionally, SEIs are of value to administrators in assessing perceived effectiveness of instruction. This information is often used for both formative and summative purposes and occasionally to reward exemplary teaching [3]. The formative purpose of evaluation is used to facilitate the faculty growth, development, and self-improvement. The summative purpose is used for taking vital decisions related to promotion and salary raises.

1.1 Current System

The University of Wisconsin-La Crosse (UW-L), currently uses a paper-based process for SEI administration. At the end of the semester, the SEI forms are distributed to students in classes and the students are asked to fill out the forms within a short period of time, usually about 10 to 15 minutes. These SEI response forms are in machine readable format which is processed by a computer and a report is generated with the results. Additional comments will be written by student, on the back side of the form. Due to the high volume of courses and students, the entire process of collecting the forms, processing and report generation take several weeks to a few months. The evaluation forms are standardized by the Faculty Senate at UW-L.

1.2 Disadvantages of the Current System

The current (SEI) process is cumbersome in terms of paper work, resulting in manual survey distribution, collection, process and result distribution. Following are the disadvantages with the paper-based approach:

- Some students might be missing when in-class evaluations are conducted and there is no appropriate way to permit students to fill out the evaluation form.
- The collection and processing of student feedback data, using paper questionnaire sheets, has always been labor intensive and time consuming.
– This approach may not allow timely and flexible feedback of evaluation results to administrators and faculty.
– At any point, evaluation forms can be lost, misplaced, misused or incorrectly administered which compromises confidentiality and security.

1.3 Proposed System

The purpose of the OSEI project described in this report is to develop and implement an effective online system that is useful and beneficial for all stakeholders within the university — student, faculty, and administrators. The objectives for developing this project are as follows:
– to eliminate the paper based evaluations and provide a web-based solution to submit the evaluation forms online
– to reduce costs associated with personnel, printing, and administration of the evaluation;
– to increase system integrity, security and student anonymity.
– to allow students to complete course evaluations at their own time, without the time constraints of in-class surveys
– to provide flexibility in course assessment questionnaire design and modification;
– to make data readily available online for analysis and reporting
– to provide the desired report generation
2. Software Life Cycle Model for OSEI

A software development life cycle is a process of developing software in a series of activities beginning with the feasibility study and ending with maintenance of the completed project. There are several models developed to streamline the software development process; some of these models include waterfall model, spiral model, incremental development, reusable software model, throwaway prototyping model and evolutionary prototyping model [4]. Each model has its own advantages and disadvantages; it is up to the development team to select the model that suits its needs and appropriate for implementing the project. Typically, every life cycle model addresses the different phases of software development including requirements, design, implementation, integration, testing, operations and maintenance.

Both waterfall and incremental software development models were investigated for OSEI development. An incremental software development model was selected for OSEI because of advantages in accommodating the changes or new requirements as the product evolves. One of challenges in OSEI development was rapidly changing requirements. It was known that the OSEI project would start with small set of requirements and more requirements would evolve or change as the application is developed and tested. In the incremental model the development proceeds from initial subset set of requirements to more and more complete subsets, until the complete system is addressed [2]. The incremental model allows the developer and project sponsors to take the advantage of what was being learned during the development of earlier prototypes and deliver versions of the system. With the incremental model, the basic functionality of the system is implemented in the initial prototype. Functionality of the system is then enhanced in later prototypes. This model extends the waterfall model with several short cycles of analysis, design, implementation and testing. Each cycle provides feedback to the next cycle, in which a more refined and enhanced development is achieved.

An incremental software development model is shown in Figure-1 as per Ian Sommerville in his literature “Software Engineering”, all activities related to analysis, design, and
implementation and testing are carried out in more flexible way in this model than in classic waterfall model [6].

Figure 1: Incremental Life Cycle Model
3. Development of the OSEI

In any software development, requirements gathering and analysis is an important. Requirements’ gathering is a task of communicating with customers, project sponsors and stakeholders to determine the requirements. Since the requirements were not clear or complete in the beginning, the project has been started with the available requirements for Computer Science Department. Accordingly, project advisor Prof. Kasi Periyasamy and CS Department Associate Becky Yoshizumy were identified as stakeholders and customers for development of OSEI System. Professor Kasi Periyasamy and Becky Yoshizumy continued their assistance in requirements gathering and definition throughout the project.

3.1 Requirement Analysis

The initial set of requirements was gathered through informal discussions, interviews, emails and phone calls with the project stakeholders. These requirements were taken from the current SEI forms and questionnaire used by only the Computer Science Department. Consequently, the requirements do not reflect survey forms required by the entire university. It was at the same time the Faculty Senate was proposing a new questionnaire for student evaluation. This made a significant change in the OSEI requirements. After spending significant time with customers, to understand the objectives, deliverables and the scope of the project, the requirements gathering phase was successfully completed. While identifying the requirements, there was a need for checking the format of the evaluation form and that of the reports. The current paper-based evaluation system process was studied to understand and develop the requirements for formatting considerations.

3.2 Requirements Development

After the initial set of requirements, several new requirements were identified through consultation with Faculty Senate representative and other departments. The processing of scantron paper-based evaluation forms are done by a computer in Information Technology
Services (ITS) at UW-L. The precise statistical reporting algorithm was not fully understood and improperly documented. Consequently, in the new online system, there was a need to identify the processing and report generation. Lack of documentation caused the developer to conduct a number of interviews and discussions for identifying the processing algorithm.

Due to changing requirements, three incremental prototypes were developed. The first prototype was developed with the basic requirement of converting paper-based evaluation to an online system. Basic screens with backend processing for system login, logout, creating and modifying user accounts, course information, evaluation questions and comments were implemented within this prototype.

Requirements that followed the first prototype related to questionnaire design, GUI design, maintaining multiple accounts for administrators, roles and privileges for administrators and students, and requirements specific to students and courses resulting from demonstrating the first prototype to the customers. These requirements led to the development of the second prototype. To make the system more comprehensive, additional requirements related to reports and requirements specific to data archival were gathered prior to developing the third prototype. Following is a summary of the list of complete requirements defined for the OSEI tool:

- A student should able to login to the system, fill out, and submit an evaluation form for each course he/she has registered.
- Providing user friendly interface and comprehensive logical flow of pages which consists of main menu, administrator menu and student menu.
- Students should use their university ID and password when logging into the system.
- When a student submits his/her evaluation for a course, only the responses of the student for the questionnaire should be saved. No personal identification of the student should be saved.
- A student should be able to go back and forth between the pages of an evaluation for a particular course. Once the student submits the evaluation form for a course, then the
student/administrator can’t change the information entered. Hence, that course must be removed from the list of courses available for evaluation, for that student.

- Design an evaluation form consisting of multiple choice and open-ended questions asking for students' opinions about various aspects of a course.
- Questions are categorized into two types: mandatory and optional. Mandatory questions are common for all departments. Optional questions are specific to a department.
- All questions for comments should be optional. There should be a common set of questions that apply to all departments. In addition, each department may have specific questions pertaining to that department only. The system should be able to identify and separate the department specific questions for comments.
- The OSEI evaluation form should contain five sections
  - Section I: student related information
  - Section II: mandatory questions related to instructor and course
  - Section III: optional questions
  - Section IV: questions for comments
  - Section V: department specific questions for comments.

- Administrators have different levels of access to the system. Two different roles were assigned to the Administrators:
  **System-Level Administrator**: These users have access to system-wide information, which includes student, course, evaluation questionnaire and reports.
  **Department-Level Administrator**: These users have the same access as the System level administrators but they can handle only their department reports.

- A Department-Level Administrator should able to view and modify student, course, questions and comments pertaining to the particular department.
- A Department-Level Administrator should able to generate reports only specific to his/her department.
- A Department-Level Administrator should able to add, modify and delete optional questions and comments pertaining to the particular department.
A System-Level Administrator should be able to view and modify students, course, questions, and comments for all departments.

A System-Level Administrator should be able to generate reports depending on the department selected.

The database for the evaluation form submitted should have all the information except the data related to the student who has submitted the form.

Information for a student's course should also indicate whether the course has been evaluated or not.

All student comments, for a given course, should be saved in a text file.

All parts of the evaluation form should be modifiable.

Reports should be categorized based on instructor, course, departments, and questions.

Instructor reports should include the responses for every question for the courses taught by the instructor. Evaluation results in this report should include the number of responses, the fractional median, and the composite fractional median for each question within a response scale value of A (Strongly agree) - E (Disagree).

The department report should include evaluations for responses to the questions categorized according to the student classification, reason for student taking the course, and the grade student is expecting for the course. Evaluation results in this report should include the number of responses, the fractional median, and the composite fractional median for each question with a response scale value of A-E.

Course reports should include evaluations for responses to the questions categorized according to the student classification, reason for student taking the course, and the grade student is expecting for the course. Evaluation results in this report should include the number of responses, the fractional median, and the composite fractional median for each question with a response scale value of A-E.

Question reports should include evaluations for responses per question for all instructors. Evaluation results in this report should include an evaluation for comparison of department composite values.
3.3 OSEI Design and Implementation

The design goal of OSEI is to make it accessible to only authorized people, and at the same time make it customizable so that it can serve any department at UWL. Moreover, it is made user friendly. OSEI is designed as a web application requiring only a web browser for users to use. This makes the OSEI tool easily accessible for all the users.

Each user in OSEI is given a specific role called Student, System-Level Administrator or Department-Level Administrator. Students are permitted only to complete and submit evaluation forms. Both (System-Level and Department-Level) Administrators can create/modify/delete user accounts, courses, questions and comments for evaluation forms to be completed by the students. A System-Level Administrator can view the results of evaluation and generate reports for all departments. A Department-Level Administrator can view the results and generate reports that are specific to one department. Every user follows a specific sequence of tasks starting with login attempt. The streamlined workflow for each role makes it possible to implement an easy to use GUI (graphical user interface). A use case diagram is shown in Figure 2[8].
Figure 2: Use Case Diagram for OSEI
3.3.1 Technologies Used In System Development

Web applications usually consist of resource files (e.g. Images), web components, helper classes and libraries. Business logic processing is usually done on a server and browsers acts as a thin clients to communicate with the server. Data-oriented web applications are organized in a 3-tier architecture with web browser acting as a user interface on the client side; a dynamic web content technology, such as CGI, ASP or Java Servlets, used in at the functional (business logic) layer.

Web applications are dynamic extensions of a web or application server. Web applications are generally either service-oriented or presentation-oriented [1]. The OSEI system is developed as a presentation-oriented web application that produces interactive web pages written in mark up languages such as XML and HTML. The contents of each page are dynamically created as responses to the requests created by the users through interactive sessions. A service-oriented application will provide a loosely-coupled suite of services that are distributed over a network and can be reused to create business applications. These services communicate with each other by passing data from one service to another, or by coordinating an activity between one or more services. Presentation-oriented applications often are used as clients for service-oriented applications.

The OSEI system was implemented using the Microsoft web development technologies. It is an ASP.net web application designed and implemented using Microsoft Visual Studio 2008 Professional.NET development environment with C# as the programming language. Client side code was written in HTML, AJAX and JavaScript, with potentially embedded server-side script written in C#. Server Side code that process event handlers for events such as button click is also written using C# and ASP. MySQL5.0 is used to store all data in this application. A brief introduction on different technologies used in development of the OSEI system and the way these technologies are used are stated in subsequent paragraphs.
Microsoft .NET Architecture

Microsoft .NET 3.0 framework is used for building and deploying interoperable web solutions. The .NET framework mainly consists of two components, common language runtime (CLR) and a set of class libraries. The CLR is responsible for run-time services such as language integration, thread management and security enforcement. Class libraries provide standard functionalities such as input/output, string manipulation, network communication, and user-interfaces. OSEI is developed using various interfaces, enumerations and classes derived from the system namespace in .NET framework library. Namespaces like system, System.Collections, System.Data, System.Linq, System.Configuration, System.Web and System.XML from .NET framework library are extensively used within development of OSEI. The use of .NET class libraries in OSEI, helped in easy access to basic data types, access and manipulate MYSQL data, perform input and output and output operation, system security checks, create extravagant graphical user interface and create dynamic ASP.net pages. Another reason for using .Net framework in OSEI development is its interoperability

ASP.NET

ASP.NET is a language that allows dynamic, interactive web pages to be created on the web server. ASP provides for multithreading and handles a large number of users. The OSEI interface consists of many asp.net pages through which users interact with the system. Each ASP page is basically an HTML page with associated C# code that will be processed by the server before it is returned back to the browser. Regardless of what language is used for the webpage, an ASP.NET code is processed by the web server and is replaced with HTML before the response is sent back to the browser. To improve user experience and to make OSEI system more responsive, ASP.net AJAX technology was also used to retrieve data from server asynchronously and update parts of existing page.

ASP.NET AJAX Suite

Within OSEI system development, AJAX technology is used to enable a client with rich server side processing capabilities without requiring post-back. AJAX stands for Asynchronous
JavaScript and XML. AJAX is based on JavaScript and HTTP requests, which are generally used for exchanging the data with server and changing parts of webpage without reloading the whole page. Microsoft has developed a set of AJAX extensions for ASP.NET developers implementing the .NET solutions. ASP.NET AJAX extensions is a suite of AJAX library, Server framework and AJAX templates. AJAX library is JavaScript library that provides the features for the client portion of the ASP.NET AJAX framework. UpdatePanel is an ASP.NET server control that updates portions of a web page without reloading it. The server portion of the ASP.NET AJAX framework uses ASP.NET server controls like an Extender and script control for adding client capabilities to server controls.

MySQL

The OSEI development project is a database driven web system. Large volume of data will be collected and accessed using this system, and so a secure and reliable database was required for the implementation of this project. Microsoft Access and MySQL were examined as potential RDBMS candidates for development of system. Microsoft Access was used during the initial stage of the project, but the project later moved to MySQL because of it is open-source. In addition, Microsoft Access has some limitations with regard to data encryption, scalability, stability and reliability. MySQL is a multithreaded, multi-user, SQL relational database server and popular due to reports of fast performance, high reliability and ease of use. Although MySQL lacks some of optimizer sophistication and parallel features of other commercially available databases, these are not considered relevant to this project.

3.3.2 User Interface Design

The OSEI system contains a web-based graphical user interface. Designing a user interface that is easy to understand is crucial for OSEI system as it provides a mechanism for different users with varying background to interact with the system. OSEI interface consists of ASP pages through which users interact. The screens are developed in ASP.NET, which are basically HTML pages with associated code that is processed by the server before it is returned to the client. OSEI interface starts with a login screen, which allows the registered users to log into the system. Once a user logs into the system, depending on their role and privileges,
respective screens will be displayed to the user. For example, a user with role “student” has only
privileges to access the student screens. More information about user roles and privileges and
their access levels is described in design and implementation chapter. All OSEI web pages are
AJAX enabled. A script manager control is used to enable the following features of Microsoft
AJAX for ASP.NET

- Client-script functionality of the Microsoft Ajax Library.
- Partial-page rendering, which enables regions on the page to be independently
  refreshed without a post-back.

With partial-page rendering, individual regions of the web page are refreshed
asynchronously without reloading the whole web page with every post-back. This makes the
page more responsive to the user. ‘UpdatePanel’ control is used in OSEI web pages to achieve
partial-page rendering.

### 3.3.3 Database Design

A database diagram is used to visualize the system and represent the user’s requirements.
The database diagram in Figure-3 represents different OSEI entities and how they relate to one
another. Each table within the OSEI database is represented by a box in the database diagram.
Primary keys for each table are represented in bold, italic and underlined. The relationships are
represented by arrowed lines.

Lines start from the foreign key in one table and end at the related table having the
primary key. For example, every record in the 'student_course' table must have a unique student
record in the 'student' table. The relation is maintained in such a way that, it is not possible to
delete any student in the 'Student' table if it is having the related records (courses have been
taken by the student) in 'Student_Course' table.
Figure 3: Database Diagram for the OSEI

Entire OSEI database is represented by the following four major sections.

1. User Accounts
2. Evaluation
3. Reports
4. Archives

User Accounts:

Administrator

1. **Table Name**: tbladministrator
   
   Column Attributes: UserId, Password, Dept
Description: This table is used to store all administrators in the system. The access level depends on the department assigned to the user. The user with Department as ‘UWL’, will be the System-Level Administrator who can access all the department functionalities. The Department-Level Administrators (ex: Dept=’CS’) can access reports for only his/her department. Administrators will have a unique user id which is the primary key for this table. Dept field is used to store the department code.

Student

2. **Table Name**: tblstudent
   Column Attributes: UserId, Password, StdId, email

Description: This table is used to store all students in the system. Each student will have a unique user id which is the primary key for this table. User id must be in 8.4 format used by UW-L. Password must have at least one numeric value. Student id must have exactly 9 digits or numeric characters. Email id will be created automatically from user id (<<username>>@uwlax.edu, where <<username>> is the email ID of the student in 8.4 format).

Evaluation

3. **Table Name**: tblcourse
   Column Attributes: courseId, semester, year, section, title, code

Description: This table is used to store all courses in the system. Each course will have a unique id created from the department code (ex: ‘CS742’). The primary key for this table is a composition of courseid, semester and year. A course title is stored in the ‘title’ field and instructor code is stored in the field ‘code’.

4. **Table Name**: tblstdcourse
   Column Attributes: stdid, courseId, semester, year, section, eval
Description: This table is used to store all courses assigned to the students in the system. The primary key is a composition of stdid, courseid, semester and year. ‘eval’ is a flag used to indicate whether or not the course has been evaluated.

5. **Table Name**: tblappl_man_qst  
   Column Attributes: QNum, QDescr

Description: This table is used to store all mandatory questions for the evaluation form. Each question will have a unique number (QNum), which is the primary key for the table.

6. **Table Name**: tblappl_opt_qst  
   Column Attributes: QNum,QDescr,QDept

Description: This table is used to store all department questions for the evaluation form. Each question will have a unique number (QNum), which is the primary key. The department code (QDept) indicates the department for which the optional questions will be displayed in the evaluation form.

7. **Table Name**: tblcomments  
   Column Attributes: cnum, cdescr

Description: This table is used to store all common comments for the evaluation form. Each comment will have a unique number (cnum), which is the primary key.

8. **Table Name**: tbldeptcomments  
   Column Attributes: cnum, cdept, cdescr

Description: This table is used to store all department comments for the evaluation form. Each comment will have a unique number (cnum), which is the primary key. The department code
(cdept) indicates the department for which the department comments will be displayed in the evaluation form.

9. **Table Name**: tblevaluation
   
   Column Attributes: evalId, courseId, semester, year, section, inscode, answers, comments

   Description: This table is used to store the information from the evaluation form submitted by the student. Each evaluation will have a unique number (evalId), which is the primary key. Each record that is stored in this table will have the course details, instructor code, answers entered and comments given (if any) by the student.

**Reports**

10. **Table Name**: tbldepartment
    
    Column Attributes: code, descr

    Description: This table is used to store all the departments’ information in the university. Each record will have a unique code, which is the primary key.

11. **Table Name**: tblinstructor
    
    Column Attributes: code, fname, lname

    Description: This table is used to store all instructors’ information in the university. Each record will have a unique code, which is the primary key.

12. **Table Name**: tbldeptreport
    
    Column Attributes: dept, semester, year, qnum, counta, countb, countc, countd, counte, countf, type, fraqmed.
Description: This table is used to store the department report. The primary key is the composition of dept, semester, year, qnum and type. The field ‘countx’ is used to store number of responses given for ‘x’, for a particular question. The field ‘frqmed’ stores fractional median for the question. Each report belongs to a category (Ex: Graduate, Senior, Undergrad etc.) which is stored as ‘type’ in the table.

13. **Table Name:** tblquestionreport
   
   Column Attributes: dept,semester,year,qnum,insCode,counta, countb, countc, countd, counte, countf, fraqmed, medfivemax.

Description: This table is used to store the question report. The primary key is the composition of dept, semester, year, qnum and insCode. The field ‘countx’ is used to store number of responses given for ‘x’, for a particular question. The field ‘frqmed’ stores fractional median for the question. 'The field 'medfivemax' is used to store the calculated fractional median from the values entered for fifth question to last question.

14. **Table Name:** tblcoursereport
   
   Column Attributes: courseId,semester,year,section, qnum,insCode,type, counta, countb, countc, countd, counte, countf, fraqmed.

Description: This table is used to store the course report. The primary key is the composition of courseid, semester, year, qnum and insCode. The field ‘countx’ is used to store number of responses given for ‘x’, for a particular question. The field ‘frqmed’ stores fractional median for the question. Each report belongs to a category (Ex: Graduate, Senior, Undergrad etc.) which is stored in the field ‘type’.

15. **Table Name:** tblinstructorreport
   
   Column Attributes: courseId,semester,year,section, qnum,insCode,type, counta, countb, countc, countd, counte, countf, fraqmed, medfivemax
Description: This table is used to store the instructor report. The primary key is the composition of courseid, semester, year, qnum and insCode. The field ‘countx’ is used to store number of responses given for ‘x’, for a particular question. The field ‘fracmed’ stores fractional median for the question. The field ‘medfivemax’ is used to store the composite fractional median for the values given for questions five to the last question in evaluation form.

16. **Table Name**: tblinstructordeptreport  
   **Column Attributes**: dept, semester, year, section, qnum, insCode, type, counta, countb, countc, countd, counte, countf, fracmed.

Description: This table is used to store the department report. The primary key is the composition of dept, semester, year, qnum and insCode. The field ‘countx’ is used to store number of responses given for ‘x’, for a particular question. The field ‘fracmed’ stores fractional median for the question.

**Reports**

17. **Table Name**: student_archive  
   **Column Attributes**: UserId, Password, StdId, email  
   Description: This table is used to store all students that has been deleted from the system.

18. **Table Name**: course_archive  
   **Column Attributes**: courseId, semester, year, section, title, code  
   Description: This table is used to store all courses that has been removed from the system.

19. **Table Name**: studentcourse_archive  
   **Column Attributes**: stdid, courseId, semester, year, section, eval  
   Description: This table is used to store all archived data of the courses taken by the students. once the data from 'tblstdcourse' table is deleted, the information is archived in to this table.
3.3.4 Detailed Architecture of SEI

The architectural diagram for the OSEI system is shown in above Figure-5. The OSEI system is developed as a web based application, in which users interact through web pages. Each web page in OSEI system provides a distinct functionality of the system. The OSEI system consists of different functionalities that are listed in Table-1. Before using any of the functionalities of the system, the first step is to login into the system. Initially, the system is installed with a predefined username and password for a System Level Administrator. This user should be able to create other users in the system. Once a user logs into the system, other functionalities of the system are accessible depending on the privileges that he/she has.
<table>
<thead>
<tr>
<th><strong>Functionalities</strong></th>
<th><strong>Purpose</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Administrator Login</td>
<td>Allows System-Level Administrator and Department-Level Administrator to log in to the system.</td>
</tr>
<tr>
<td>Module</td>
<td></td>
</tr>
<tr>
<td>2 Student Login</td>
<td>Allows students to log in to the OSEI system. Login information used by student will not be stored for security purposes. Administrator and Student will have their own menus which will be displayed depending on their login information.</td>
</tr>
<tr>
<td>3 Change Password</td>
<td>Allows user to change his/her password. Both Administrators and students are allowed to change their passwords at anytime once they login into the system.</td>
</tr>
<tr>
<td>4 Log out</td>
<td>Allows users to log out of the OSEI system.</td>
</tr>
<tr>
<td>5 Submit Evaluation Form</td>
<td>Allows a student to fill up and submit the evaluation form for an unevaluated course.</td>
</tr>
<tr>
<td>6 Add/Delete Administrator</td>
<td>Allows a System-Level Administrator to add/delete all other Administrators. Department-Level Administrators can add/delete other Department-Level Administrators only.</td>
</tr>
<tr>
<td>7 Add/Delete/Modify Course</td>
<td>Allows both System-Level and Department-Level Administrators to add/delete/modify courses. A course cannot be deleted if at least one student is associated with the course.</td>
</tr>
<tr>
<td>8 Add/Delete Student</td>
<td>Allows both System-Level and Department-Level Administrators to add/delete students. A student cannot be deleted if any course is associated with the student.</td>
</tr>
<tr>
<td>9 Add/Delete Student-Course</td>
<td>Allows both System-Level and Department-Level</td>
</tr>
<tr>
<td>Relationship</td>
<td>Administrators to add/modify/delete the relationships between courses and students.</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>10</strong> Add/Modify/Delete Questions</td>
<td>Allows both System-Level and Department-Level Administrators to add/modify/delete questions on the evaluation form. At least one question has to be maintained in a form.</td>
</tr>
<tr>
<td><strong>11</strong> Add/Modify/Delete Optional Questions.</td>
<td>Allows both System-Level and Department-Level Administrators to add/modify/delete questions for the evaluation form for a given department.</td>
</tr>
<tr>
<td><strong>12</strong> Add/Modify/Delete Comments.</td>
<td>Allows both System-Level and Department-Level Administrators to add/modify/delete comments in the evaluation form.</td>
</tr>
<tr>
<td><strong>13</strong> Add/Modify/Delete Department Comments.</td>
<td>Allows both System-Level and Department-Level Administrators to add/modify/delete comments in the evaluation form for given department.</td>
</tr>
<tr>
<td><strong>14</strong> View Course Comments</td>
<td>Allows both System-Level and Department-Level Administrators to view the course comments made by a student for a given department, semester and year.</td>
</tr>
<tr>
<td><strong>15</strong> Add/Delete Departments</td>
<td>Allows both System-Level and Department-Level Administrators to add/delete departments in the university. A department cannot be deleted if a course is assigned to the department.</td>
</tr>
<tr>
<td><strong>16</strong> Add/Delete Instructors</td>
<td>Allows both System-Level and Department-Level Administrators to add/delete instructors in the university. An instructor cannot be deleted if he/she is assigned to a course.</td>
</tr>
<tr>
<td><strong>17</strong> View Department Report</td>
<td>Allows Administrator to view department report for a given department, semester and year. All System-</td>
</tr>
</tbody>
</table>
### Challenges faced in the development of OSEI

There were many challenges faced in the development of the online OSEI System. The challenges faced during the development stage and solutions adopted to overcome the challenges are listed below:

1. **Changing Requirements.** In the beginning it was hard to understand the scope of the project mainly because there was no uniformity across the departments in using the evaluation system. Because of lack of uniformity in the evaluation process, the development was started with a vague description of the overall project. The requirements continued changing throughout the project.

   **Solution:** To handle the changing requirements, the incremental prototyping approach was used in the development of OSEI, which helped in demonstrating intermediate products of the OSEI to the customer before it reached the final stage. A total of three

<table>
<thead>
<tr>
<th>No.</th>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>View Question Report</td>
<td>Allows Administrator to view question report for a given department, semester and year. All System-Level Administrators can view this report but Department-Level Administrator can view the reports pertaining to the department.</td>
</tr>
<tr>
<td>19</td>
<td>View Course Report</td>
<td>Allows Administrators to view course reports for a given department, semester year and course.</td>
</tr>
<tr>
<td>20</td>
<td>View Instructor Report</td>
<td>Allows Administrators to view instructor report for a given department, semester year and instructor. All System-Level Administrators can view this report but Department-Level Administrator can view only the reports pertaining to the department.</td>
</tr>
</tbody>
</table>
incremental prototypes were delivered during the development process. This approach assisted in satisfying customers.

2. **Inconsistent usage of evaluation forms across departments within UWL.** Some departments use online evaluation forms and others use manual evaluations. Content of evaluation forms across departments also differed significantly. It presented a challenge to define the evaluation form that is acceptable for all departments.

   **Solution:** To handle inconsistent usage of evaluation forms across departments the evaluation form is categorized into two sections. The first section consists of questions that are common for all departments, and the second section consists of optional questions that can be customized to be specific for a department.

3. **Reports design and implementation.** Requirements for reports were not clear at the beginning regarding the data and the formats of reports. They were developed based on the information collected from the Computer Science department, but later the scope was expanded to cover all departments within UWL. Currently, the reports are designed as defined by the Faculty Senate.

   **Solution:** During the initial phase of the project, there was no clear definition on format of reports and the data to be presented in the reports. Reports generated through existing paper-based evaluation use spreadsheets. After evaluating these reports and several discussions with customer, the number of reports to be generated and the format for the reports were identified. There was still confusion over the formulas used for calculating Average, median, fractional median and composite values for the responses to the questions in the evaluation form. The developer then met with Dr. Robert Hoar, Professor in Mathematics, UWL. The Mathematics Department used an online system as a pilot project. Dr. Hoar suggested the logic to make these formulas. After working on the logic, the new formulas were created. Even those formulas also went through several changes while testing with different types of data.
4. **Implementing flexibility in structure of the evaluation form:** There are no standard sections for evaluation form; it may vary in number of questions or comments.

   **Solution:** Flexibility is implemented in creating the evaluation form so that the number of pages, the number of questions and the number of comments can be changed depending upon the contents selected.

5. **Addressing or responding to student concerns for privacy and anonymity.** This is a major concern that students may have as they think their evaluations and responses would not be confidential in an online system.

   **Solution:** The OSEI system was designed to allow the user to login to get the unevaluated courses but after the submission of the evaluation student anonymity is ensured. The SEI System has been developed using standard log-on, user administration, authorization checks and encryption. Students use their own university IDs and passwords when logging into the online system. Students can evaluate only their own registered (unevaluated) courses and they can submit the form only once per course. Student responses are separated from the student information thus maintaining the anonymity of students. The results are stored electronically and can be accessed by only the authorized people. To view the reports, the administrator should use his/her unique user id and password.

### 3.3.6 Testing the OSEI System

The OSEI system was tested for functionality and integration using black box testing technique. At code level unit tests were executed focusing on the algorithmic details of the OSEI
modules and the data flow across interfaces. From all the functionalities that had been tested, a few are listed below:

- Student Fill up and submit the application form:
  1. Entered all the required information and submitted the form.
  2. Entered for only questions by leaving the comments unattended.
  3. Entered only few questions and comments by leaving few questions unattended.
  4. Student selected ‘Cancel’ on last page.

- Administrator changes the list of questions:
  1. Change the order of the list.
  2. Delete one or more questions from the list.
  3. Change the data for one or more questions.
  4. Add a new question.

- Administrator changes the list of comments:
  1. Change the order of the list.
  2. Delete a comment from the list.
  3. Change the data for a comment.
  4. Add a new comment.

- Administrator maintains the courses:
  1. Add new course information.
  2. Trying to delete a course which has been associated with some students.
  3. Change the information for a course.
• Administrator retrieves a department report:

1. Click on the report button with no information given.

2. Click on the report button by selecting only one department.

3. Click on the report button by selecting a department, semester and year.
4. Limitations

The following limitations are identified within the OSEI system.

- Personal information of the users is not maintained in this version.
- Reports are generated but distribution and other functionalities such as conversion of reports in a particular format or printing the reports are not included in this project.
- There is no functionality to send the remainder email notifications to the students about submitting the evaluation forms once they finish taking the course.
- OSEI currently supports MySQL database. It is possible to support other databases by changing the database connection string within the code.
5. Continuing Work

This section lists the requirements and enhancement to be implemented in future versions of this product:

- Email notification to students for reminding the evaluation form.
- Generate evaluation reports in printable version.
- Deploying the OSEI system at a university website.
- Survey to compare online system with paper based system by considering the response rates from the students and their comments.
6. Conclusion

The goal of this project was to develop an online student evaluation system to replace the paper-based evaluation system currently used at University of Wisconsin, La Crosse. The OSIE system was developed using incremental software development model to accommodate the changing and new requirements as the system evolved. Key features including the development of evaluation forms and analysis of the submitted forms for report statistics are successfully implemented. Functionalities of OSEI are designed and implemented in such a way that they are accessible only to the registered user as per their system privileges.

Each user in OSEI is given a specific role. Students can fill the evaluation forms and submit them. Both (System-Level and Department-Level) Administrators can create/modify/delete user accounts, courses, questions and comments for evaluation forms to be completed by the students. A System-Level Administrator can view the results of evaluation and generate reports for all departments. A Department-Level Administrator can view the results and generate reports that are specific to their respective department. The system was tested for functionality and browser efficiency. Future work is required for the deployment of the system and implementation of email notifications.
7. Bibliography


APPENDIX A: Sample OSEI Class Diagram

Figure 5: OSEI Class Diagram
APPENDIX B: Sample OSEI Screenshots

Figure 6: Administrator Login
Figure 7: Administrator Account

Figure 8: New Administrator Account
Figure 9: Course Screen
Figure 10 : New Course Screen
Figure 11: Instructor Screen

Figure 12: Modify Course Screen
Figure 13: Mandatory Questions Screen
Figure 14: New Mandatory Question Screen
Figure 15: Optional Questions Screen

Figure 16: New Optional Question Screen
Figure 17: Comments Screen

Figure 18: New Comments Screen
Figure 19: Students Screen
Figure 20: New Student Screen

Figure 21: Student Course Screen
Figure 22: New Student Screen
Figure 23: Department Screen

Figure 24: New Department Screen
Figure 25: Main Reports Screen

Figure 26: Department Report
Figure 28: Course Report

![Course Report Image](image_url)

Figure 29: Instructor Report

![Instructor Report Image](image_url)
Figure 30: Evaluation Form Screen 1

Figure 31: Evaluation Form Screen 2
Figure 32: Evaluation Form Screen3
Figure 33: Evaluation Form Screen 4

Figure 34: Evaluation Form Screen 5
Figure 35: Evaluation Form Screen6