

FACULTY SCHOLASTIC REPORT

by

Kedarnath Myadam
Bachelor of Computer Science, VTU, 2003

A Project

Submitted to the Graduate Faculty

of the

University of North Dakota

in the partial fulfillment of the requirements

for the degree of

Master of Science

Grand Forks, North Dakota

December

2006

This project document, submitted by Kedarnath Myadam in partial fulfillment of the requirements for the Degree of Master of Science from the University of North Dakota, has been read by the Faculty Advisor under whom the work has been done and is hereby approved.



(Faculty Advisor)

This report document meets the standards for appearance, conforms to the style and format requirements of the Computer Science Department of University of North Dakota, and is hereby approved.



Graduate Director

12/5/06

Date


PERMISSION

Title: Faculty Scholastic Report

Department: Department of Computer Science

Degree: Master of Computer Science

In presenting this report in partial fulfillment of the requirement for a graduate degree from the University of North Dakota, I agree that the Department of Computer Science shall make it freely available for Inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by the professor who supervised my work or, in his absence, by the Chairperson of the Department.

Signature 

Date 12/05/06

TABLE OF CONTENTS

LIST OF FIGURES	vi
LIST OF TABLES	viii
ACKNOWLEDGMENTS	ix
ABSTRACT	x
CHAPTER	
I. INTRODUCTION	1
1.1 Project Outline	1
1.2 Report Organization	1
II. REQUIREMENTS AND SPECIFICATION	3
2.1 Requirements	3
2.2 Specifications	6
III. SOFTWARE DESIGN	13
3.1 Entity-Relationship Model	13
3.2 Database Tables	15
3.3 UML Diagrams	18
IV. IMPLEMENTATION	23
4.1 Client Side Modules	23
4.2 Server Side Modules	28
V. SOFTWARE VERIFICATION AND VALIDATION	33
5.1 Unit Testing	33

5.2 Integration Testing	34
5.3 Validation Testing.....	35
VI. CONCLUSION.....	41
6.1 Future Considerations	41
REFERENCES	43
APPENDIX I	45
APPENDIX II.....	53

LIST OF FIGURES

Figure	Page
1. Tier 3 Template.....	6
2. Prototype of login screen	7
3. Prototype of registration screen	7
4. Prototype of menu screen.....	8
5. Prototype of search screen	9
6. Prototype of search result screen	9
7. Prototype of search item details screen.....	10
8. Prototype of update item screen.....	11
9. Prototype of delete item screen.....	11
10. Prototype of report generation screen.....	12
11. ER diagram of FSR.....	14
12. Use case diagram of FSR.....	20
13. Deployment diagram of FSR	21
14. DFD of FSR login system.....	22
15. DFD of FSR registered user options.....	22
16. FSR template.....	24
17. Form for adding a new grant.....	25
18. Form for searching a grant.....	26
19. User options before generating report.....	31

20.	Generated PDF report	32
21.	FSR index page before user logs into the system	36
22.	FSR index page after user logs into the system	36
23.	A test showing characters , and	40
24.	FSR index page	45
25.	FSR login page.....	46
26.	FSR registration screen.....	47
27.	FSR snapshot showing the menu	48

LIST OF TABLES

Table	Page
1. Structure of Users table.....	15
2. Structure of Ctaught table	16
3. Structure of Cdeveloped table.....	16
4. Structure of Ssupervised table	16
5. Structure of Publication table.....	17
6. Structure of Presentation table	17
7. Structure of Grants table	17
8. Structure of Proposals table	18
9. Structure of Misc table.....	18
10. Navigation to add an entry	48
11. Navigation to update an entry	49
12. Navigation to delete an entry	50
13. Navigation to search an entry	51
14. Navigation for report generation.....	52

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Ronald Marsh, Associate Professor in the department of Computer Science for providing excellent guidance through out my project and my client Dr. Kanishka Marasinghe, Associate Professor in the department of Physics for suggesting this project. Special thanks to my family for their constant love and understanding, without which it would have made this work impossible.

ABSTRACT

Maintaining the academic records and generating *October Supplement* has been a time consuming task. Utilizing the local system for this purpose would necessitate the user to be in presence of the same local system to access or modify the data. Faculty Scholastic Report is focused on the development of the online software system that is more flexible and less time consuming to maintain the academic records and to generate *October supplement*. The project is a web based system implemented on IIS (Internet Information Services) provided by Microsoft® Windows® XP.

The developed system will help the faculty to store, accesses and modify their data online and to generate *October Supplement* using a password protected database. It would be much flexible for the faculty to maintain their records in server instead of a local system.

CHAPTER I

INTRODUCTION

Every year the faculty of University of North Dakota submits their scholastic achievements (progress) via a document known as the *October Supplement* to the Graduate school. The faculty has to prepare this document, which includes the courses taught, courses developed, students supervised, publications, presentations, proposals, grants and research activity for each academic year. Preparing the document and organizing it, is a time consuming task and would take at least a week. When Dr.Kanishka Marasinghe explained me about the problem, I came up with the proposal of developing an online system “Faculty Scholastic Report”, where the faculty can enter their scholastic achievements time to time through a web based form and the creation of *October Supplement* would be by just selecting time interval and clicking a button.

1.1 Project Outline

The process involves requirement analysis, designing, verification and validation. A number of software’s and procedures are required to accomplish this task. Requirement analysis includes developing a model by acquiring information, function, and behavior of the system. The model is translated to data, architectural, and component-level designs. Verification and validation assures that the software system meets the user’s needs. Generating a set of test cases will facilitate the testing. The front end of the software

system shall be done using PHP, HTML, and JavaScript; the back end shall be MySQL. In addition to this some application software's will be used.

1.2 Report Organization

The next chapter describes the Requirements and Specifications of this application. Chapter 3 describes Design of the software system. Chapter 4 focuses on the Implementation. Chapter 5 describes Verification and Validation. Chapter 6 contains the Conclusion.

CHAPTER II

REQUIREMENTS AND SPECIFICATION

The main objective of the Faculty Scholastic Report (FSR) is to generate the *October supplement*. At this stage of SDLC (System Development Life Cycle), the client requirements are gathered and documented. Requirements are provided by Dr.Kanishka Marasinghe, Associate Professor in Physics.

2.1 Requirements

The *October Supplement* contains information such as courses taught, courses developed, students supervised, publications, presentations, proposals, grants and research activity. Preparing this document and organizing it is a time consuming task and takes at least a week. The project is focused on the development of an online software system that is more flexible, less time consuming to maintain, and to generate the *October Supplement*. The primary requirement of FSR is to maintain consistency with other websites at the University of North Dakota, so a template provided by University Relations shall be used to implement FSR.

The objectives of the FSR are to allow faculty to:

1. Access the system using a web browser
2. Authenticate themselves as users of the system
3. Add, update, and delete entries
4. Search by title, and search by year

5. Insert special symbols (viz. Greek alphabets) into the database
6. Obtain PDF (Portable Document Format) reports

FSR shall contain various sections and each section purpose is as follows:

- **Courses Taught:** This section shall allow the user to enter the details of the course which include course title, semester, number of students enrolled in the course, and year. These details will be saved, and the user will be able to retrieve the data whenever required. A search based on course title, semester, and year shall be provided.
- **Courses Developed:** This section shall allow the user to enter the details of the new course developed which include course description and year. These details will be saved, and the user will be able to retrieve the data whenever required. A search based on description, and year shall be provided.
- **Students Supervised:** This section shall allow the user to enter the details of students supervised which include student name, student degree, year, and description of the student work. These details will be saved, and the user will be able to retrieve the data whenever required. A search based on student name, student degree, and year shall be provided.
- **Publications:** This section shall allow the user to enter the details of publications which include publication title, author(s) name, journal name, volume, pages, year and date. These details will be saved, and the user will be able to retrieve the data whenever required. A search based on publication title, author(s) name, journal name, and year shall be provided.

- **Presentation:** This section shall allow the user to enter details of presentations which include presentation title, authors(s) name, event, venue, and date. The details will be saved, and the user will be able to retrieve the data whenever required. A search based on presentation title, author(s) name, event, and venue shall be provided.
- **Grants:** This section shall allow the user to enter the details of the grants awarded which include role of the user in the grant, title, starting date, ending date, source, amount and description of the grant. The role can be PI, Co PI, Salaried participant or Non salaried participant. A search based on the title of the grants, source, and comments shall be provided.
- **Proposals:** This section shall allow the user to enter the details of the grant proposals which include role of the user in the grant, title, starting date, ending date, source, and amount of the grant as well as description of the proposal. The role can be PI, Co PI, Salaried participant or Non salaried participant. A search based on the title of the proposal, source and comments shall be provided.
- **Miscellaneous:** This section shall allow the user to add an extra category other than the above mentioned categories. The interface includes title, content, and year. A search based on title, content, and year shall be provided.
- **Reports:** The users will be able to generate reports on each of the sections separately as well as the *October supplement* as a whole for the selected time interval. The generated report will be in PDF and the users shall be able to save it in their local machine.

2.2 Specifications

After understanding all the requirements of the client, the programming language chosen was PHP (Hypertext Preprocessor) which is an open source scripting language for producing dynamic web pages. The DBMS (Database Management System) chosen was MySQL which is an open source database. PDF reports will be generated using FPDF libraries.

Based on requirements, the specifications of FSR are as follows:

- **Specification 1:** FSR shall be a web-based online system.
- **Specification 2:** FSR shall be implemented using the “Tier 3 Template” provided by the University Relations of University of North Dakota. The “Tier 3 Template” is shown in Figure 1.



Figure 1: Tier 3 Template

- **Specification 3:** The system requires a username and password in order to use it. The login screen is depicted below in Figure 2.

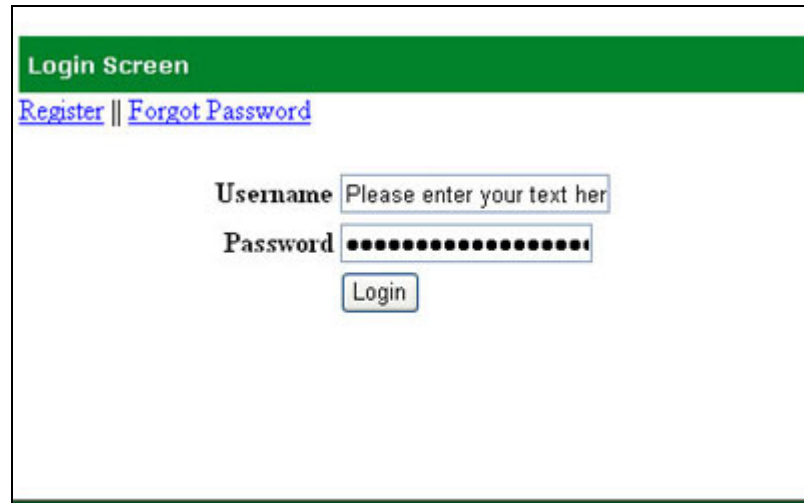


Figure 2: Prototype of login screen

- **Specification 4:** New users have to register before using the system. A valid email address is required because the FSR activation link is sent to the email address given during registration. The registration is complete only after the user clicks the activation link. The registration screen is depicted in Figure 3.

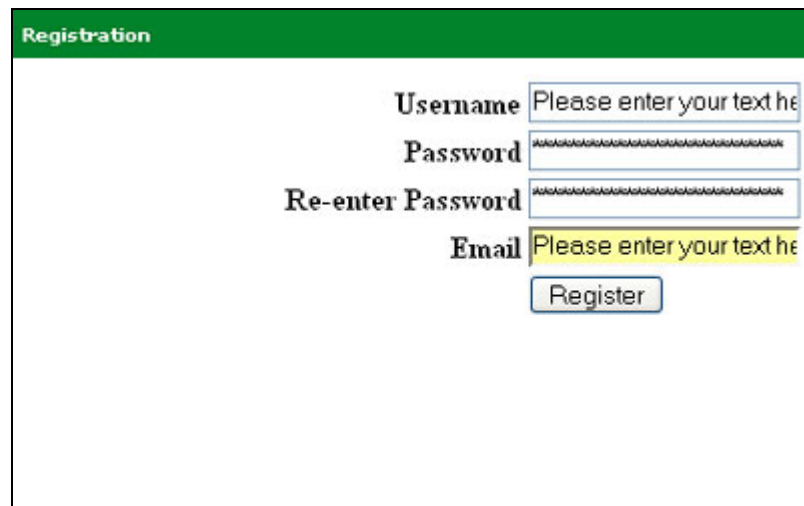


Figure 3: Prototype of registration screen

- **Specification 5:** Web pages shall be created for ten different categories. Each category serves some specific purpose. The ten categories are:

1. Courses Taught

2. Courses Developed
3. Students Supervised
4. Publications
5. Presentations
6. Grants
7. Proposals
8. Miscellaneous
9. Reports
10. Help

Navigation to all the above categories shall be provided as a menu. The menu is depicted below in Figure 4.

Courses Taught
Courses Developed
Students Supervised
Publications
Presentations
Grants
Proposals
Miscellaneous
Reports
Help

Figure 4: Prototype of menu screen

- **Specification 6:** Each category shall have its own search feature as depicted in Figure 5.

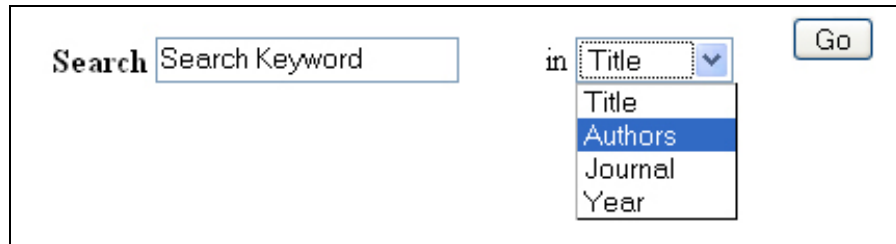


Figure 5: Prototype of search screen

By default, the system assumes that the user wants to search by “Title”. The user can change the search according to their choice. The search is executed by clicking the “Go” button. When the system finishes executing the search, a screen depicted below in Figure 6 appears.

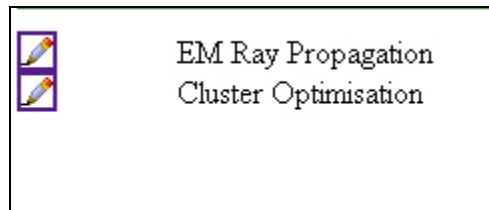


Figure 6: Prototype search result screen

From the search screen, the user may decide to view the complete details of a particular item by clicking on the item of interest. This is shown in Figure 7.

Grants Details	
Update Delete	
Title	EM Ray Propagation
Role	PI
Starting date	19-02-2005
Ending date	17-09-2009
Source	Army
Amount	\$550 K
Comments	Graphical analysis of EM RAY Propagation

Figure 7: Prototype of search item details screen

In the Search Item Details screen, the user can perform the following activities:

1. Update the details of the entry. By clicking on the “Update” link at the top of the page, the user is taken to a new page which is depicted below in Figure 8.

Figure 8: Prototype of update item screen

2. Delete the details of the entry. By clicking on the “Delete” link at the top of the page, the user is taken to a new page which is depicted in the Figure 9.

Figure 9: Prototype of delete item screen

- **Specification 7:** The users shall have a password recovery option, in which they need to enter their email address to retrieve their password.

- **Specification 8:** The user's password shall be encrypted before saving to the database. The encryption shall be done using the standard Unix DES-based (Data Encryption Standard) encryption algorithm.
- **Specification 9:** The user can enter special symbols (viz. Greek alphabets) into the database by entering the corresponding code for the symbol. An interface with codes for each symbol shall be provided.
- **Specification 10:** PDF reports are generated dynamically using FPDF libraries. The user shall select the starting date and ending date to generate the report. The screen for generating reports is depicted in Figure 10.

Report Generation Step 1

Starting Date MM 1 DD 1 YYYY 1900

Ending Date MM 1 DD 1 YYYY 1900

Publications

Presentation

Grants

Proposals

Miscellaneous

Select All/Unselect All

Show >>>>

Figure 10: Prototype of report generation screen

CHAPTER III

SOFTWARE DESIGN

Software Design involves analyzing the requirements and designing the project with a modeling language. FSR is modeled using the Entity-Relationship (ER) model and the Unified Modeling Language (UML).

3.1 Entity-Relationship Model

The ER model is a conceptual data model that views the real world as a construct of entities and relationship between entities. A basic component of the model is the Entity-Relationship diagram (ER diagram) [2]. An ER diagram is a detailed, logical and graphical representation between entities, associations and data elements [3]. Entities are the principal data object about which information is to be collected. An association shows the relationship between entities. Data elements are the attributes of the entity.

In the Entity Relationship diagram we can define three types of relationships between entities:

- **One-to-One:** One instance of an entity is associated with another instance of an entity.
- **One-to-Many:** One instance of an entity is associated with zero, one or many other instances of another entity.
- **Many- to-Many:** One, zero or many instance of an entity is associated with zero, one or many other instances of another entity.

All the relationships in the ER diagram of FSR are one-to-many (represented as 1: N). The primary keys (that uniquely identify each record in the table) are underlined for each *entity* in the ER diagram of FSR. The ER diagram of FSR is shown below in Figure 11.

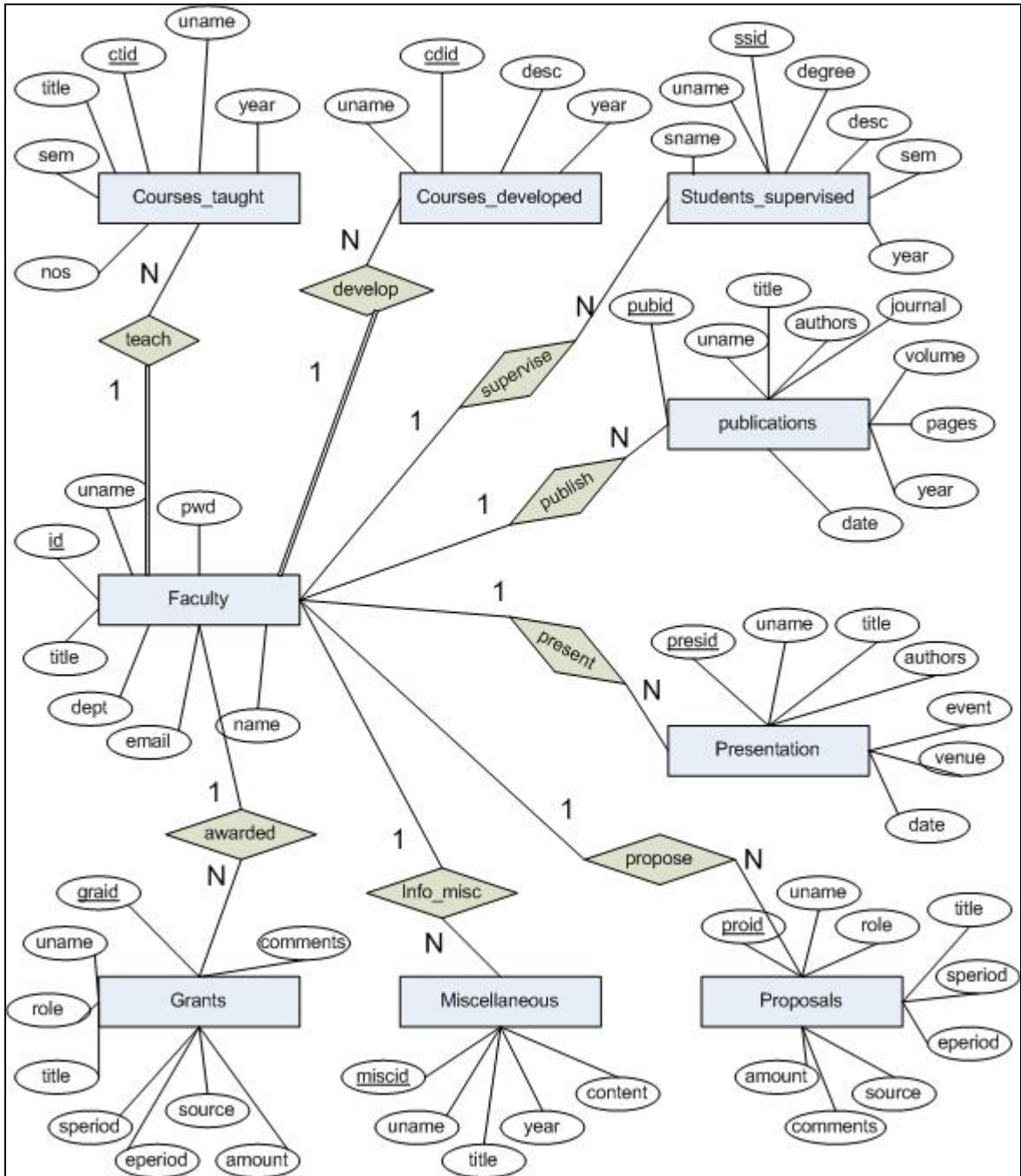


Figure 11: ER diagram of FSR

3.2 Database Tables

The ER diagram helps to identify the relationship between tables and define the database. Databases are typically organized into tables, which are collections of related items [4]. The tables in the FSR database are as follows:

1. Users
2. Ctaught
3. Cdeveloped
4. Ssupervised
5. Publications
6. Presentation
7. Grants
8. Proposals
9. Misc

The Users table contains information about the user personal information and login details. The structure of the table is shown in Table 1.

Table 1: Structure of Users table

Field	Type	Null	Extra	Key
Id	Int(11)	No	Auto increment	-
Username	Varchar(25)	No	-	Primary
Password	Varchar(25)	No	-	-
Fname	Varchar(255)	No	-	-
Lname	Varchar(255)	No	-	-
Title	Varchar(255)	No	-	-
Department	Varchar(255)	No	-	-
Email	Varchar(255)	No	-	-

The Ctaught table contains information about the courses taught by the faculty. There are six fields in this table. The structure of the table is shown in Table 2.

Table 2: Structure of Ctaught table

Field	Type	Null	Extra	Key
Ctid	Int(11)	No	Auto increment	Primary
Username	Varchar(255)	No	-	Foreign
Title	Varchar(255)	Yes	-	-
Semester	Varchar(255)	Yes	-	-
Year	Year(4)	Yes	0000	-
Nos	Int(11)	Yes	-	-

The Cdeveloped table contains information about courses developed by the faculty. There are four fields in this table. The structure of the table is shown in Table 3.

Table 3: Structure of Cdeveloped table

Field	Type	Null	Extra	Key
Cdid	Int(11)	No	Auto increment	Primary
Username	Varchar(255)	No	-	Foreign
Description	Text	Yes	-	-
Year	Year(4)	Yes	0000	-

The Ssupervised table contains information about students supervised. There are seven fields in this table. The structure of the table is shown in Table 4.

Table 4: Structure of Ssupervised table

Field	Type	Null	Extra	Key
Ssid	Int(11)	No	Auto increment	Primary
Username	Varchar(255)	No	-	Foreign
Sname	Varchar(255)	Yes	-	-
Degree	Varchar(255)	Yes	-	-
Description	Text	Yes	-	-
Semester	Varchar(255)	Yes	-	-
Year	Year(4)	Yes	0000	-

The Publication table contains information about publications. There are nine fields in this table. The structure of the table is shown in Table 5.

Table 5: Structure of Publication table

Field	Type	Null	Extra	Key
Pubid	Int(11)	No	Auto increment	Primary
Username	Varchar(255)	No	-	Foreign
Title	Text	Yes	-	-
Authors	Varchar(255)	Yes	-	-
Journal	Varchar(255)	Yes	-	-
Volume	Varchar(255)	Yes	-	-
Pages	Varchar(255)	Yes	-	-
Year	Varchar(255)	Yes	-	-
Date	Date	Yes	0000-00-00	-

The Presentation table contains information about presentations. There are seven fields in this table. The structure of the table is shown in Table 6.

Table 6: Structure of Presentation table

Field	Type	Null	Extra	Key
Presid	Int(11)	No	Auto increment	Primary
Username	Varchar(255)	No	-	Foreign
Title	Varhcar(255)	Yes	-	-
Authors	Varchar(255)	Yes	-	-
Event	Varchar(255)	Yes	-	-
Venue	Varchar(255)	Yes	-	-
Date	Date	Yes	-	-

The Grants table contains information about grants. There are nine fields in this table. The structure of the table is shown in Table 7.

Table 7: Structure of Grants table

Field	Type	Null	Extra	Key
Graid	Int(11)	No	Auto increment	Primary
Username	Varchar(255)	No	-	Foreign
Role	Varchar(255)	Yes	-	-
Title	Varchar(255)	Yes	-	-
Start period	Date	Yes	0000-00-00	-
End period	Date	Yes	0000-00-00	-
Source	Varchar(255)	Yes	-	-
Amount	Varchar(255)	Yes	-	-
Comments	Text	Yes	-	-

The Proposals table contains information about proposals. There are nine fields in this table. The structure of the table is shown in Table 8.

Table 8: Structure of Proposals table

Field	Type	Null	Extra	Key
Proid	Int(11)	No	Auto increment	Primary
Username	Varchar(255)	No	-	Foreign
Role	Varchar(255)	Yes	-	-
Title	Varchar(255)	Yes	-	-
Start period	Date	Yes	0000-00-00	-
End period	Date	Yes	0000-00-00	-
Source	Varchar(255)	Yes	-	-
Amount	Varchar(255)	Yes	-	-
Comments	Text	Yes	-	-

The Misc table contains miscellaneous information. There are five fields in this table. The structure of the table is shown in Table 9.

Table 9: Structure of Misc table

Field	Type	Null	Extra	Key
Miscid	Int(11)	No	Auto increment	Primary
Username	Varchar(255)	No	-	Foreign
Title	Varchar(255)	Yes	-	-
Year	int(5)	Yes	-	-
Content	Text	Yes	-	-

3.3 UML Diagrams

The Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the structure of software systems. UML uses mostly the graphical notations to express the design of software projects [5].

3.3.1 Use Case Diagram

A *use case* diagram displays the relationship among *actors* and *use cases*. An *actor* represents a user or another system that will interact with the system being modeled. A *use case* is an external view of the system that represents some action the user might perform in order to complete a task [6]. The *actor* and *use cases* of the FSR are represented in the *use case* diagram in Figure 12. The *actor* in FSR is:

- Faculty

The *use cases* in FSR are:

- Access the system
- Add a record
- Update a record
- Delete a record
- Search a record
- Generate reports

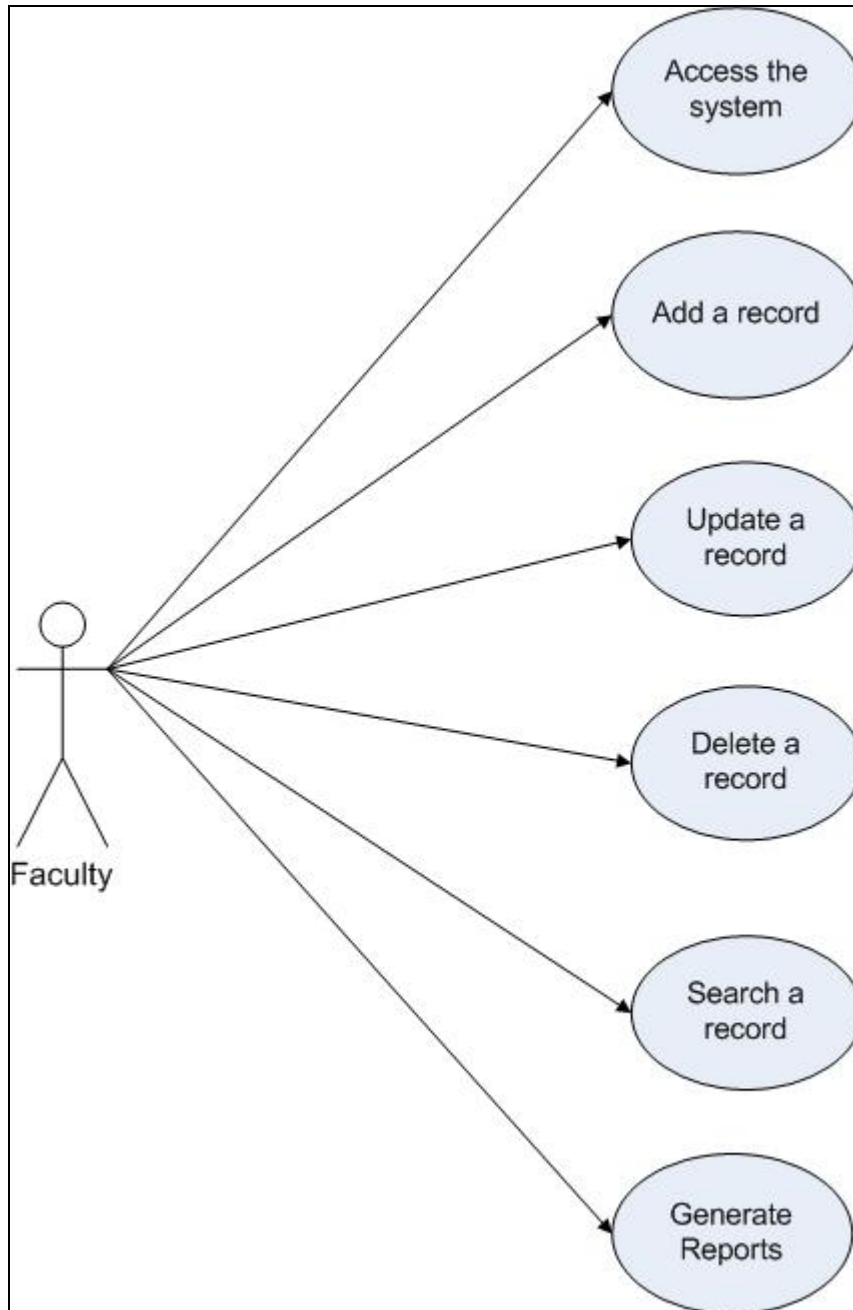


Figure 12: Use case diagram of FSR

3.3.2 Deployment Diagram

Deployment diagrams serve to model the hardware used in system implementations, the components deployed on the hardware and the associations between

those components. The elements used in deployment diagrams are nodes, components and associations. [7].

The deployment diagram of FSR is shown in Figure 13. It has three nodes namely a web browser, a web server and a database server. The web browser is the client which sends requests to the web server, the client can be any system from where the user accesses FSR. The web server processes the client request by fetching the necessary data from the database server and returning the processed data to the client. The nodes are represented as cubes, the components are shown as a rectangular box, with two rectangles protruding from the left side and associations are shown with arrows.

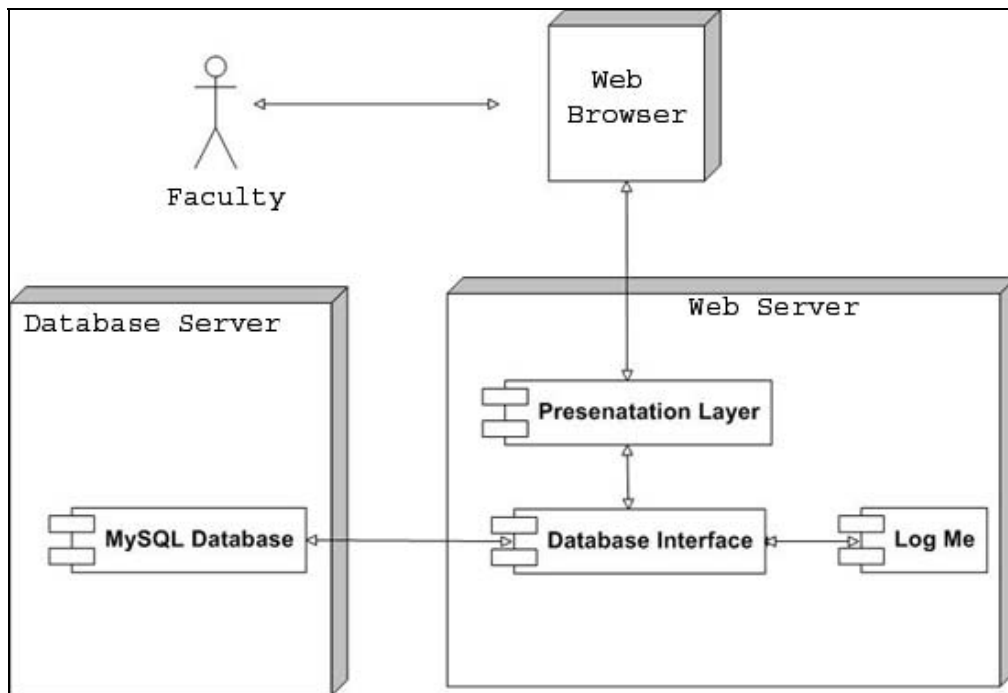


Figure 13: Deployment diagram of FSR

3.3.3 Data Flow Diagram

A Data Flow Diagram (DFD) is the graphical representation of the flow of data through an information system. It is the description and processing of the data as it moves and changes from one stage to the next. The DFD of the FSR authentication process is

shown in Figure 14. A valid login will make the session active allowing the user to access FSR.

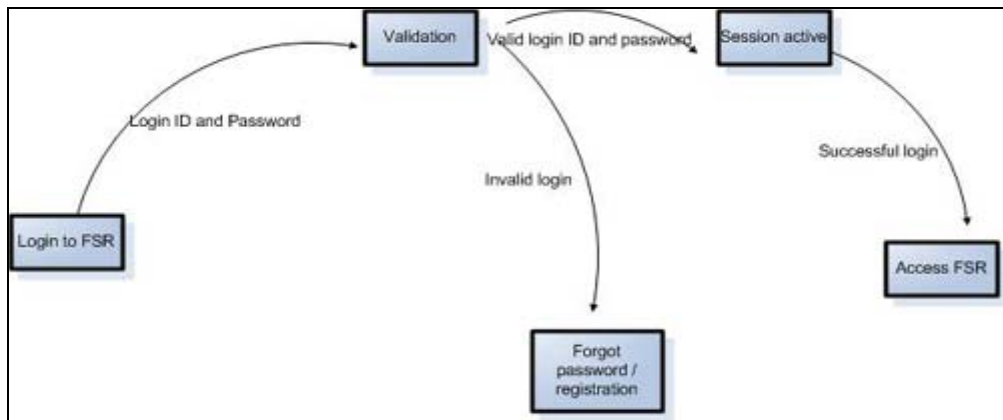


Figure 14: DFD of FSR login system

The DFD for a registered user is shown in Figure 15. The operations insert, delete and update can be performed on courses taught, courses developed, students supervised, publications, presentations, grants, proposals, miscellaneous, and personal info.

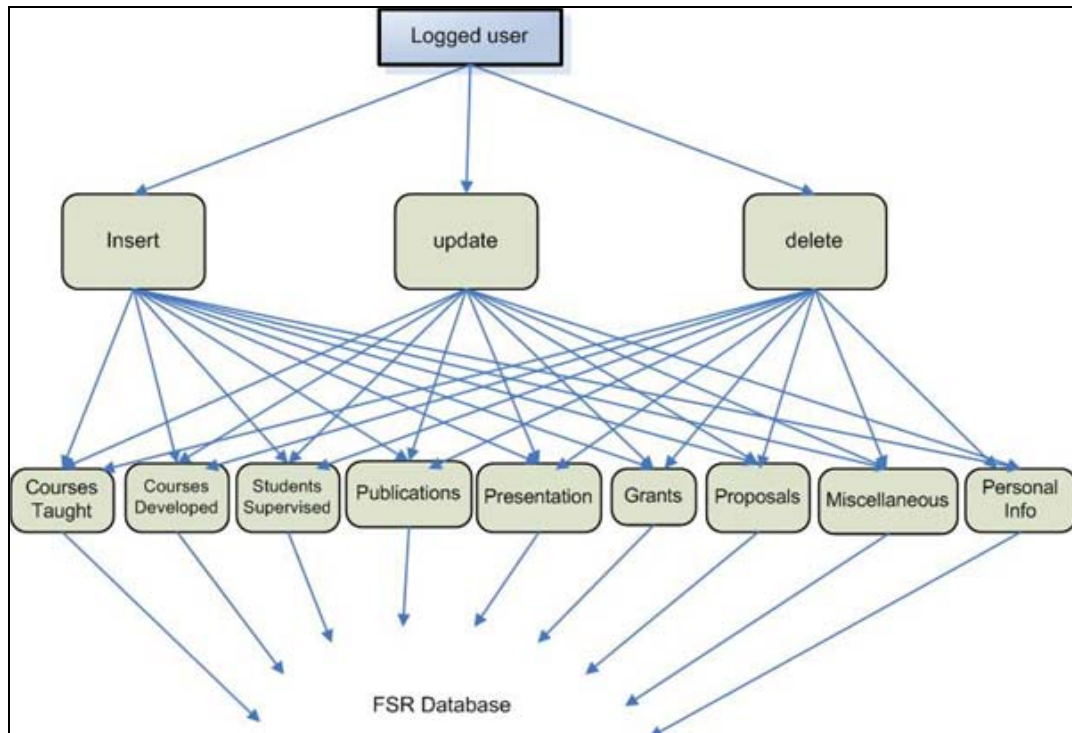


Figure 15: DFD of FSR registered user options.

CHAPTER IV

IMPLEMENTATION

FSR was implemented on IIS (Internet Information Services) provided by Microsoft® Windows® XP Professional. The client side graphical user interface is implemented using HTML (Hyper Text Markup Language) and JavaScript. The server side modules are implemented using the open source scripting language PHP (Hypertext Pre-processor). The implementation is divided into two categories:

1. Client Side Modules
2. Server Side Modules

4.1 Client Side Modules

The client side implementation is done using HTML and JavaScript. Client side implementation involves:

1. Designing the Template
2. Creating the HTML Forms
3. Validating the Forms

4.1.1 Designing the Template

The FSR is developed for the faculty of the University of North Dakota, so I used the “Tier 3 Template” design provided by University Relations of the University of North Dakota. The template designed for FSR using “Tier 3 Template” is shown in the Figure 16.

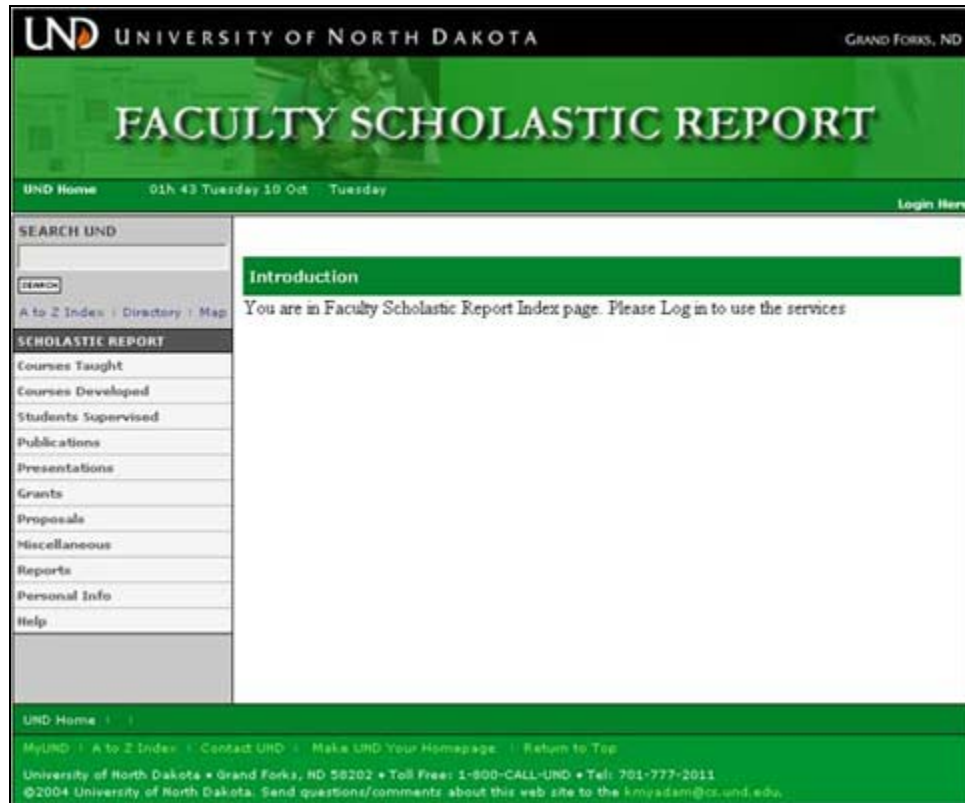


Figure 16: FSR template

4.1.2 Creating HTML Forms

HTML was used to develop the forms. Since FSR has ten different categories, forms were designed for each category according to the requirement. Figure 17 shows a form for adding a new grant.

The screenshot shows the 'Grants Add' form on the University of North Dakota website. The form is titled 'Grants Add' and is located on the right side of the page. The left side of the page contains a navigation menu with categories like 'SCHOLASTIC REPORT', 'Courses Taught', 'Courses Developed', 'Students Supervised', 'Publications', 'Presentations', 'Grants', 'Proposals', 'Miscellaneous', 'Reports', 'Personal Info', and 'Help'. The 'Grants Add' form includes the following fields:

- Role:** Radio buttons for PI, Co PI, Salaried participant, and Non salaried participant.
- Title:** A text input field.
- Starting Date:** A date picker with MM, DD, and YYYY dropdowns.
- Ending Date:** A date picker with MM, DD, and YYYY dropdowns.
- Source:** A text input field.
- Amount:** A text input field.
- Comments:** A text input field.
- Add:** A button to submit the form.

Figure 17: Form for adding a new grant

Search forms were created for every category in FSR. The search form for a grant is shown in Figure 18.

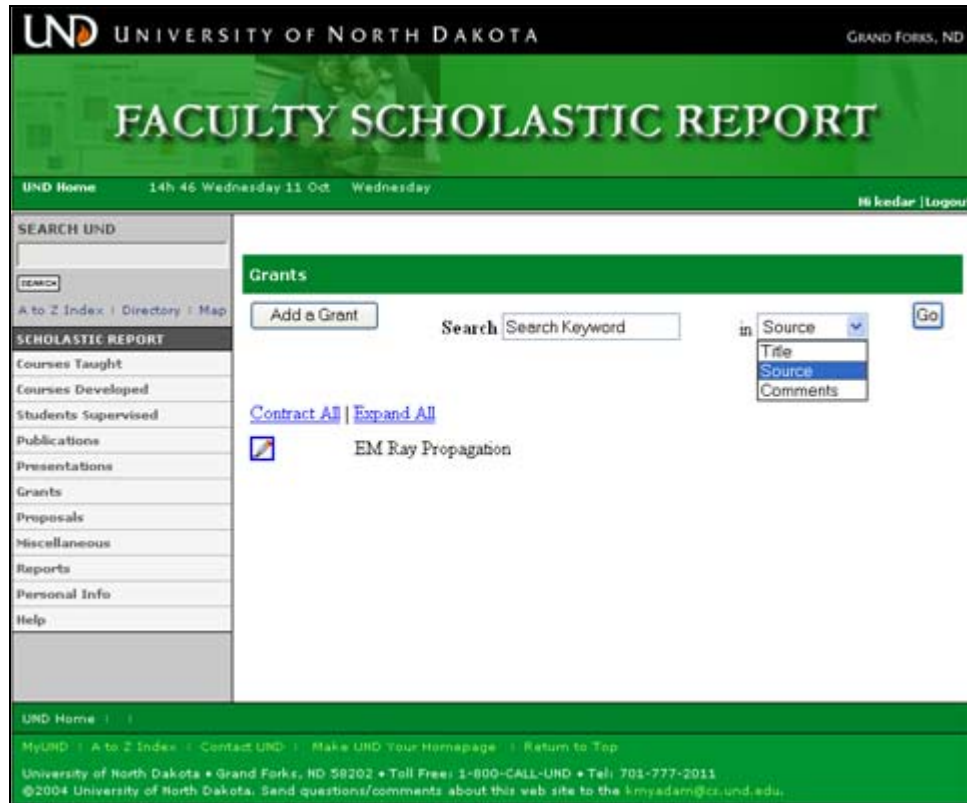


Figure 18: Form for searching a grant

4.1.3 Validating Forms

Client side validation is done using JavaScript. Form validation is the process of checking that a form has been filled in correctly before it is processed. Client side (JavaScript) validation is easier to do and quicker too (the browser doesn't have to connect to the server to validate the form, so the user finds out instantly if they've missed a required field) [8]. The following script is used to validate the email address given by the user during registration in FSR. This script makes sure that the email address:

1. Has one '@'
2. Has at least one '.'
3. Has no spaces
4. Has no extra '@'s or a '.'

```

<script language="JavaScript" type="text/JavaScript">
function echeck(str) {
    var at="@";
    var dot=".";
    var lat=str.indexOf(at);
    var lstr=str.length;
    var ldot=str.indexOf(dot);
    if (str.indexOf(at)==-1){
        alert("Invalid E-mail ID")
        return false
    }

    if (str.indexOf(at)==-1|| str.indexOf(at)==0 || str.indexOf(at)==lstr){
        alert("Invalid E-mail ID")
        return false
    }

    if(str.indexOf(dot)==-1||str.indexOf(dot)==0 || str.indexOf(dot)==lstr){
        alert("Invalid E-mail ID")
        return false
    }

    if (str.indexOf(at,(lat+1))!=-1){
        alert("Invalid E-mail ID")
        return false
    }

    if (str.substring(lat-1,lat)==dot || str.substring(lat+1,lat+2)==dot){
        alert("Invalid E-mail ID")
        return false
    }

    if (str.indexOf(dot,(lat+2))==-1){
        alert("Invalid E-mail ID")
        return false
    }

    if (str.indexOf(" ")!=-1){
        alert("Invalid E-mail ID")
        return false
    }

    return true
}

function ValidateForm(){

```

```

var emailID=document.register.email
if ((emailID.value==null)||(emailID.value==""))
{
    alert("All fields are required !!")
    return false
}
if (echeck(emailID.value)==false)
{
    emailID.focus()
    return false
}
return true
}
</script>

```

4.2 Server Side Modules

Server side modules are implemented using PHP. Server side implementation involves and the some:

1. Verify Email Address
2. Encryption
3. Sessions
4. Database Connectivity
5. Generating Reports

4.2.1 Verify Email Address

When a user registers with FSR, a randomly generated activation link is sent to the email address that is provided during the registration. The user can use the services of FSR only after activating his/her account by clicking the link. The function for generating random code is given below.

```

function createRandomCode()
{
    $chars = "abcdefghijklmnopqrstuvwxyz023456789";
    srand((double)microtime()*1000000);
    $i = 0;
    $pass = " ";

```

```

while ($i <= 10)
{
$num = rand() % 33;
$tmp = substr($chars, $num, 1);
$pass = $pass . $tmp;
$i++;
}

```

4.2.2 Encryption

User passwords are encrypted using the PHP crypt function. The crypt function returns an encrypted string using the standard Unix DES-based (Data Encryption Standard) encryption algorithm [13]. The code for encrypting the password is given below.

```

$salt = substr($password, 0, 2);
$encrypted_pswd = crypt($password, $salt);

```

4.2.3 Sessions

A session is the sequence of pages that a user visits in a website, from the moment they log on to the site until they log out. A session is set when the user successfully logs into the system. The code for setting a session in PHP is shown below.

```

$_SESSION ['userid'] = $userid;

```

When a user tries to access a page in FSR, the system checks the session status. If the session is not set, it redirects the user to the login page. The following code gets executed each time a user accesses a page in FSR.

```

<?
session_start();
if (!isset($_SESSION['userid']))
{
header("Location: ../index.php");
}
?>

```

When a user logs out of FSR, the session is unset. The code for unsetting a session is shown below.

```
unset ($_SESSION['userid']);
```

4.2.4 Database Connectivity

The “mysql_connect” function is used for connecting PHP with the MySQL database. It takes three arguments namely - Hostname/Server, Database username and password. The function returns a “link identifier” when the connection is successfully made with the server [9]. The PHP function “mysql_select_db” is used for opening the database. It takes two arguments - Database name and the “link identifier” (Optional). This function returns “True” on success and “False” in case of error. The PHP function “mysql_close” is used for closing a database connection which has been opened. It takes the “link identifier” as its argument and returns “True” on success and “False” on Error [9]. The script for connecting to database is 'dbconnect.php', which is shown below.

```
<?
$db_host="localhost";
$db_user="username";
$db_password="password";
$db_dbname="octsup";
$db_connection=mysql_connect($db_host,$db_user,$db_password) or die
("Connection to database failed: " .mysql_error());
mysql_select_db($db_dbname) or die ("Could not select database $db_dbname");
?>
```

4.2.5 Generating Reports

The users can generate a report for each category separately as well as the *October supplement* as a whole for the selected time interval. Figure 19 shows the user options for generating report.

UNIVERSITY OF NORTH DAKOTA GRAND FORKS, ND

FACULTY SCHOLASTIC REPORT

UND Home 18h 30 Wednesday 11 Oct Wednesday Hi keefar Logout

SEARCH UND

SEARCH

A to Z Index | Directory | Map

SCHOLASTIC REPORT

- Courses Taught
- Courses Developed
- Students Supervised
- Publications
- Presentations
- Grants
- Proposals
- Miscellaneous
- Reports
- Personal Info
- Help

Report Generation Step 1

Starting Date MM 4 DD 1 YYYY 2000

Ending Date MM 9 DD 25 YYYY 2006

Courses Taught
 Courses Developed
 Students Supervised
 Publications
 Presentation
 Grants
 Proposals
 Miscellaneous

Select All Unselect All

Show >>>>

UND Home | |

MyUND | A to Z Index | Contact UND | Make UND Your Homepage | Return to Top

University of North Dakota • Grand Forks, ND 58202 • Toll Free: 1-800-CALL-UND • Tel: 701-777-2011
 ©2004 University of North Dakota. Send questions/comments about this web site to the kmyedam@cc.und.edu.

Figure 19: User options before generating report

When the user requests for a PDF report, the system fetches the applicable data from the database, formats the data and generates the report. The report is generated in HTML initially, and is converted to PDF using FPDF (Free Portable Document Format). FPDF is a PHP class which generates PDF files [10]. The PDF file then opens in the web browser. Figure 20 shows the generated PDF file. The following PHP code shows how to convert an HTML file to PDF.

```

<? require('../fpdfdemo/html2fpdf.php');
$pdf= new HTML2FPDF();
$pdf->AddPage();
$file=$username.'!."html";
$f=fopen($file,'r');
$buffer=fread($f,filesize($file));
fclose($f);
$pdf->WriteHTML($buffer);
$pdf->Output(); ?>

```

ACADEMIC RECORD OCTOBER 2006 SUPPLEMENT		ROUTING: COPY to Chair of the Department COPY to Dean of the College COPY to Dean of the Graduate School COPY to Provost Office
Name: Kedar Myadam	Rank: Student	Department: Computer Science
Courses Taught:		
"Introduction to Computers". Fall 2006		
Courses Developed:		
An introduction to Nano Science. This course covers all the basics necessary for the Nano science.		
Students Supervised:		
Kedar Myadam, M.S Expected to graduate in Dec 2006		
Publications:		
"EM ray propagation,", Kedar Myadam. Ray propagation analysis		
Presentations:		
"Software Engineering Techniques,", Kedar Myadam. Software Engineering workshop, Las Vegas		
Proposals(pending or rejected):		
"A proposal for Survey on Hearing Impaired", Center for Physically handicapped. \$550 K, 01/2006- 01/2007		

Figure 20: Generated PDF report

CHAPTER V

SOFTWARE VERIFICATION AND VALIDATION

‘Software Verification & Validation’ helps the system designers and test engineers to confirm that the right product is build the right way throughout the development process and to improve the quality of the software product. Verification is a process that makes sure the software product is developed the right way, the software should confirm to its predefined specifications. As the product development goes through different stages, an analysis is done to ensure that all required specifications are met. Validation is the process of determining if the product being built is “right”. The software product should functionally do what it is supposed to, it should satisfy all the functional requirements set by the user. Validation is done during or at the end of the development process in order to determine whether the product satisfies specified requirements [11]. Several types of testing methods are carried out during the validation process. Test plan, test suits and test cases are developed, which are used during the various phases of validation. The following testing strategies are used in FSR:

1. Unit Testing
2. Integration Testing
3. Validation Testing

5.1 Unit Testing

Unit testing focuses the verification effort on the smallest unit of software design – the software component or module [12]. In FSR each webpage is considered to be a unit. Each page is tested for corresponding JavaScript functions, links, missing or wrong HTML tags, and PHP code. The JavaScript functions were tested by opening the page in the browser and check whether the appropriate function is invoked by the user action on the page. Each link on the page is tested by clicking on the link and check whether the link leads to appropriate page. Missing HTML tags were difficult to find because the browser does not show any errors when a page is displayed, thus I used the “W3C Markup Validation Service” to find missing or wrong HTML tags. Since errors in PHP code are displayed when the page is opened in the browser, any errors were identified and rectified.

5.2 Integration Testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with the interfaces. The strategy is to take unit tested components and build a program structure that has been dictated by the design [12]. While integrating web pages into the project, there were many broken links (links on the World Wide Web that points to a webpage that are permanently unavailable) and missing images. To avoid this, a variable was declared globally (so that it is available to all the files in FSR) which takes the root path of the project, such that all the links and image paths are specified with reference to root path. The following code shows how this solution was implemented in FSR.

```
<?
$rootpath = ".." ;
```

```
< a href="$rootpath/login.php"> Login Here </a>  
 </img>  
>
```

5.3 Validation Testing

Software validation is achieved through a series of black-box tests that demonstrate conformity with requirements. Like conventional validation, the validation of the Web-based systems focuses on user-visible actions and user recognizable output from the system. The different black-box tests used for FSR to check for conformity with requirements are as follows:

1. Menu Enabled/ Disabled
2. Add, Update, and Delete Entries
3. Search Operation
4. Inserting Special Symbols into Database
5. Generating PDF Reports.

5.3.1 Menu Enabled/Disabled

The user can access the menu items only after logging into the system. The menu links are disabled until the user logs into the system. A test case was developed to test this feature. Figure 21 shows the FSR index page where it prompts the user to log into the system to use the FSR services.

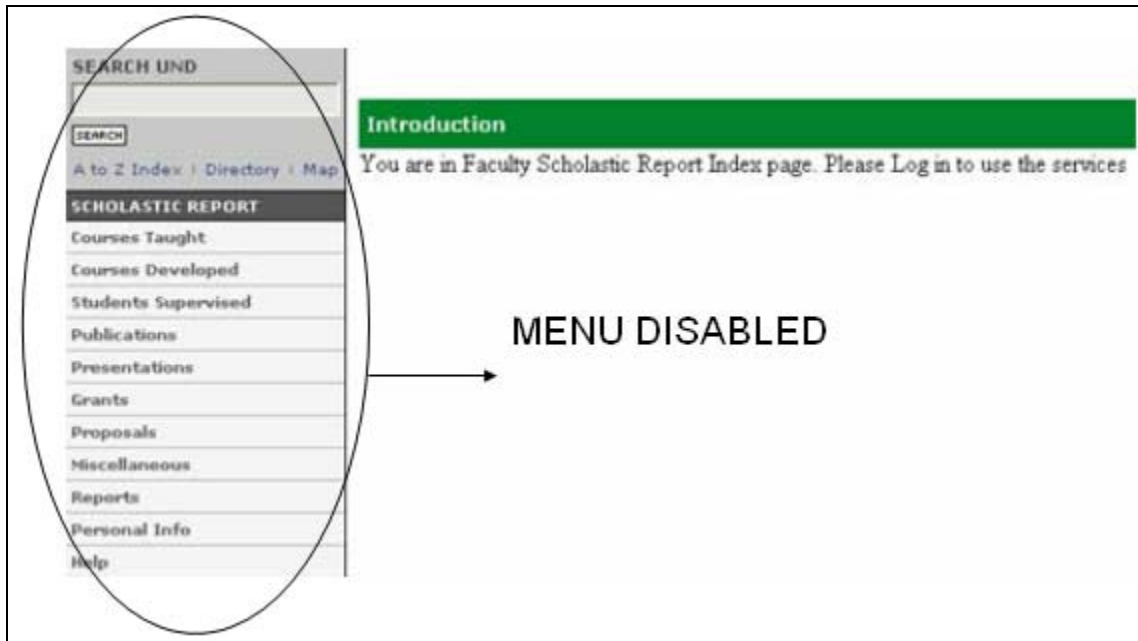


Figure 21: FSR index page before the user logs into the system

When the user successfully logs into the system, the menu is enabled and the user can use the services of FSR. Figure 22 shows this feature.

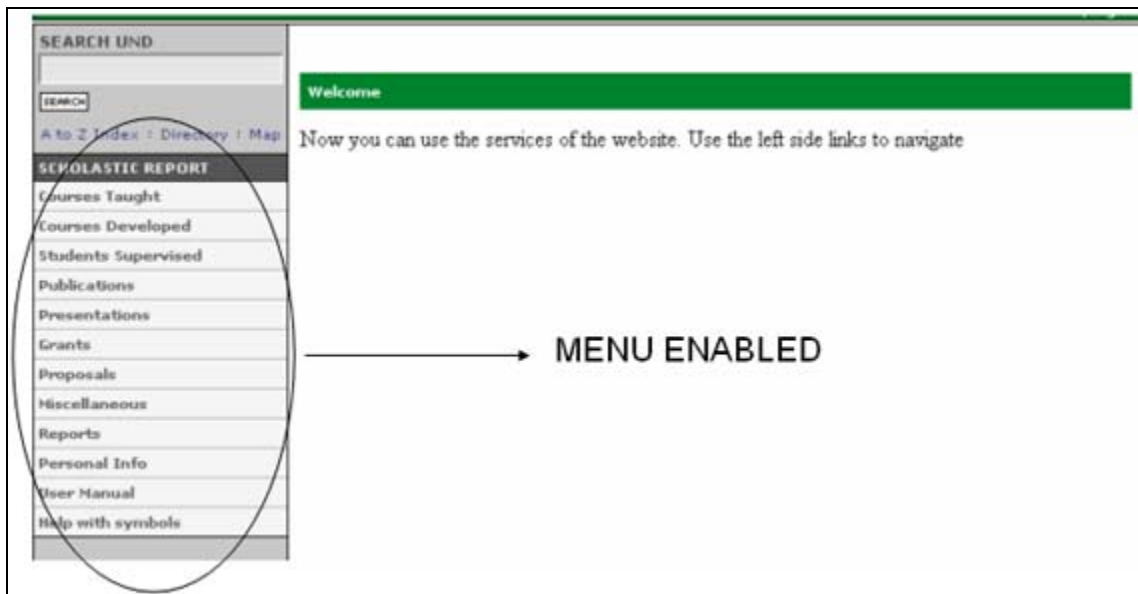


Figure 22: FSR index page after the user logs in the system

5.3.2 Add, Update and Delete Entries

The user can add, update and delete the entries whenever required. There are eight different categories in FSR that have option of adding, updating and deleting an entry. So, test cases are designed to test the features Add, Update and Delete operations. The following test case shows how the Add feature was tested in FSR:

1. Open FSR in the browser
2. Log into FSR
3. Select a category from the menu on the left
4. In the category selected, click the button “Add”
5. Verify that the form for adding details appears
6. Fill the textboxes and click the button “Submit”
7. Check whether the details are added successfully into the database
8. Select a different category from the menu on the left and repeat steps from 4 to 7
9. Repeat the step 8 until every category in FSR is tested

The following test case shows how the Update feature was tested in FSR:

1. Open FSR in the browser
2. Log into FSR
3. Select a category from the menu on the left
4. In the category selected, click on title which has be updated
5. Verify that the page showing update option for the selected title appears
6. Click the update link
7. Verify that the form for updating the details appear
8. Update the necessary fields and click “Submit” button

9. Check whether the details are updated successfully into the database
10. Select a different category from the menu on the left and repeat steps from 4 to 9
11. Repeat the step 10 until every category in FSR is tested

The following test case shows how the Delete feature was tested in FSR:

1. Open FSR in the browser
2. Log into FSR
3. Select a category from the menu on the left
4. In the category selected, click on title which has be deleted
5. Verify that the page showing delete option for the selected title appears
6. Click the delete link
7. Verify that the form for deleting the details appear
8. Confirm the deletion by clicking “Yes” button
9. Check whether the details are deleted successfully from the database
10. Select a different category from the menu on the left and repeat steps from 4 to 9
11. Repeat the step 10 until every category in FSR is tested

5.3.3 Search Operation

The search operation is implemented in eight categories in FSR. The test cases were designed to test each search operation by changing keywords and search type. The following test case shows how the Search operation was tested in FSR:

1. Open FSR in the browser
2. Log into FSR
3. Select a category from the menu on the left
4. In the category selected, enter the key word to search

5. Choose the search type
6. Click the “Search” button
7. Verify whether the search was successful
8. Select a different category from the menu on the left and repeat steps from 4 to 7
9. Repeat the step 8 until every category in FSR is tested

5.3.4 Inserting Special Symbols into Database

The FSR allows the user to insert special characters (viz. Greek alphabets) into the database. Figure 23 shows the successful entry of the special characters into the database. The following test case shows how the inserting of a symbol into the database was tested:

1. Open FSR in the browser
2. Log into FSR
3. Click the link “Help with Symbols” on the menu
4. Verify that a new page with symbols and codes opens in the browser
5. Copy the corresponding code for the symbol to be inserted into the database
6. Select a category from the menu on the left
7. In the category selected, click the “Add” button
8. Verify that the form for adding details appears
9. Enter the symbol code in the textbox and click the button “Submit”
10. Check whether the details are added successfully into the database
11. Verify that the inserted symbol appears when viewed in a browser
12. Select a different symbol code and repeat steps 7 to 11
13. Repeat step 12 until every symbol code is tested

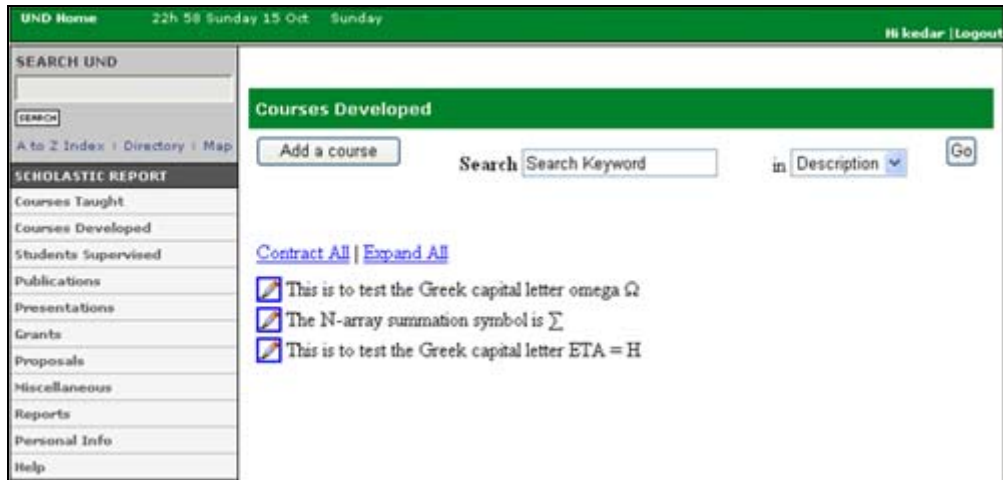


Figure 23: A test showing characters Ω , Σ , and η

5.3.5 Generating PDF Reports

The FSR implementation allows the user to generate PDF reports online. The following test case shows how generation of PDF reports in FSR is tested:

1. Open FSR in the browser
2. Log into FSR
3. Click the link “Reports” on the menu
4. Select the starting date and ending date
5. Select the categories that should appear in the report
6. Click the button “Generate PDF”
7. Verify that the PDF file opens in a new browser window
8. Verify that all the selected categories appear in the PDF file

The test cases resulted in following conclusions:

1. The user’s system should have “Adobe Acrobat Reader” installed on their machine.
2. The generation of PDF works well in “Internet Explorer” and “Mozilla Firefox”.

CHAPTER VI

CONCLUSION

The final project meets all the requirements requested by the client Dr.Kanishka Marasinghe. The project was implemented by using software engineering principles and methodologies. The developed system will help the faculty to store, accesses and modify their data online using a password protected database and generate *October Supplement*.

6.1 Future Considerations

Following are the future considerations for FSR:

1. The system is implemented using PHP and JavaScript. The PHP code for generating the dynamic HTML content is mixed with the static HTML code. If the file has a lot of PHP code mixed with static HTML code, it is difficult to debug and add new code. Future developers may consider developing the entire system using JSP and Java Beans, as JSP lets you embed special active elements into HTML pages. Another consideration for a programming language is ASP.
2. The system is developed for faculty at the University of North Dakota. The authentication process for accessing FSR is different from other web based systems at the university. A common authentication process could be implemented to access FSR using LDAP (Lightweight Directory Access Protocol). This way the user can use the university general e-mail system's username and password to access FSR.

3. PDF reports are generated using FPDF libraries and sometimes the generation of PDF file will result in error if it contains some special characters like `&` or `.`. Instead, commercial software could be used to generate the PDF reports.

REFERENCES

- [1] Flatworld Solutions Pvt. Ltd. *Requirement Analysis Process – Requirements Definition & Gathering*. Retrieved August 17, 2006, from <http://www.outsource2india.com/software/RequirementAnalysis.asp>
- [2] The University of Texas. *Introduction to Data Modeling*. Retrieved September 2, 2006, from <http://www.utexas.edu/its/windows/database/datamodeling/dm/erintro.html>
- [3] Jupitermedia Corporation. *Entity relation diagram*. Retrieved September 7, 2006, from http://isp.webopedia.com/TERM/E/entity_relationship_diagram.Html
- [4] Macromedia. *Introduction to Database and SQL*. Retrieved September 6, 2006, from <http://livedocs.macromedia.com/coldfusion/6.1/htmldocs/sql2.htm>
- [5] Braun, D., Sivilis, J., Shapiro, A., Versteegh, J. *What is UML?* Retrieved September 15, 2006, from http://pigseye.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/what_is_uml.htm
- [6] Braun, D., Sivilis, J., Shapiro, A., Versteegh, J. *Use Case Diagrams*. Retrieved September 15, 2006, from http://pigseye.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/use_case.htm

- [7] Wikipedia. *Deployment Diagram*. Retrieved October 16, 2006, from
http://en.wikipedia.org/wiki/Deployment_diagram
- [8] Elated Communication Ltd. *Form Validation with JavaScript*. Retrieved
October 2, 2006, from
http://www.elated.com/tutorials/programming/javascript/form_validation/
- [9] Jandus Technologies. *PHP and MySQL Database*. Retrieved September
23, 2006, from
<http://www.smartwebby.com/PHP/aboutdbconnect.asp>
- [10] FPDF. *FPDF Library*. Retrieved September 29, 2006, from
<http://www.fpdf.org/>
- [11] Buzzle.com. *Software Verification and Validation Model*, Retrieved
October 10, 2006, from
<http://www.buzzle.com/editorials/4-5-2005-68117.asp>
- [12] Pressman, R.S. *Software Engineering: A Practitioner's Approach*.
McGraw-Hill, New York, 2001.
- [13] The PHP group. *Crypt*, Retrieved October 22, 2006, from
<http://us3.php.net/crypt>

APPENDIX I

USER MANUAL

FSR is an online software system to maintain academic records and generate *October Supplement*. The first interface the user sees on opening FSR is shown in Figure 24.



Figure 24: FSR index page

Click the link “login here”, which takes the user to the login screen. The login screen is depicted in Figure 25.



Figure 25: FSR login page

A registered user can enter the username and password and click the button “Login”. A new user has to click the link “Register of new user” to register with FSR. If a registered user forgets the password then he/she can click the link “Forgot Password” to retrieve the new password. The registration screen is depicted in Figure 26.

The screenshot shows the 'FACULTY SCHOLASTIC REPORT' registration page. At the top, the University of North Dakota logo and 'GRAND FORKS, ND' are visible. The page title is 'FACULTY SCHOLASTIC REPORT'. Below the title, the date and time are shown as '16h 11 Monday 20 Nov Monday' and 'Login Here'. On the left side, there is a 'SEARCH UND' section with a search box and a 'UNID' dropdown menu. Below this is a 'SCHOLASTIC REPORT' menu with various categories: Courses Taught, Courses Developed, Students Supervised, Publications, Presentations, Grants, Proposals, Miscellaneous, Reports, Personal Info, User Manual, and Help with symbols. The main content area is titled 'Registration' and contains the following fields: First Name, Last Name, Title, Department, Email, User ID, Password, and Re-enter Password. A 'Register' button is located at the bottom of the form. The footer of the page includes 'UND Home', 'Return to Top', and contact information for the University of North Dakota.

Figure 26: FSR registration screen

The registration is complete only when the user provides all the information that is requested in Figure 26 and clicks the button “Register”. Once the user is done with the registration, an activation link is sent to the email address that is provided during the registration. The user has to click the activation link before using the FSR services.

The user can choose any one of the categories from the menu on the left. The menu is activated only when the user is logged in successfully. The menu is marked in the Figure 27.



Figure 27: FSR snapshot showing the menu

Operations in FSR

Add

There are eight different categories in FSR where the user can add the details into the database, the navigation in each category is shown in Table 10.

Table 10: Navigation to add an entry

Category	Navigation
Courses Taught	Start by selecting “Courses Taught” from the menu on the left → Click the button “Add a course” → Fill the form elements → Click the button “Add” → Add status.
Courses Developed	Start by selecting “Course Developed” from the menu on the left → Click the button “Add a course” → Fill the form elements → Click the button “Add” → Add status.
Students Supervised	Start by selecting “Students Supervised” from the menu on the left → Click the button “Add a student” → Fill the form elements → Click the button “Add” → Add status.
Publications	Start by selecting “Publications” from the menu on the left → Click the button “Add a publication” → Fill the form elements → Click the button “Add” → Add status.
Presentations	Start by selecting “Presentations” from the menu on the left → Click the button “Add a presentation” → Fill the form elements

	→ Click the button “Add” → Add status.
Grants	Start by selecting “Grants” from the menu on the left → Click the button “Add a grant” → Fill the form elements → Click the button “Add” → Add status.
Proposals	Start by selecting “Proposals” from the menu on the left → Click the button “Add a proposal” → Fill the form elements → Click the button “Add” → Add status.
Miscellaneous	Start by selecting “Miscellaneous” from the menu on the left → Click the button “Add a miscellaneous” → Fill the form elements → Click the button “Add” → Add status.

Updating

There are nine different categories in FSR where the user can update the details.

The navigation in each category is shown in Table 11.

Table 11: Navigation to update an entry

Category	Navigation
Courses Taught	Start by selecting “Courses Taught” from the menu on the left → Click the course you want to update → Click the link “Update” → Update the required fields → Click the button “Update” → Update status.
Courses Developed	Start by selecting “Courses Developed” from the menu on the left → Click the course you want to update → Click the link “Update” → Update the required fields → Click the button “Update” → Update status.
Students Supervised	Start by selecting “Students Supervised” from the menu on the left → Click the student you want to update → Click the link “Update” → Update the required fields → Click the button “Update” → Update status.
Publications	Start by selecting “Publications” from the menu on the left → Click the publication you want to update → Click the link “Update” → Update the required fields → Click the button “Update” → Update status.
Presentations	Start by selecting “Presentations” from the menu on the left → Click the presentation you want to update → Click the link “Update” → Update the required fields → Click the button “Update” → Update status.
Grants	Start by selecting “Grants” from the menu on the left → Click the grant you want to update → Click the link “Update” → Update the required fields → Click the button “Update” → Update status.
Proposals	Start by selecting “Proposals” from the menu on the left → Click the proposal you want to update → Click the link “Update” →

	Update the required fields → Click the button “Update” → Update status.
Miscellaneous	Start by selecting “Miscellaneous” from the menu on the left → Click the miscellaneous you want to update → Click the link “Update” → Update the required fields → Click the button “Update” → Update status.
Personal Info	Start by selecting “Personal Info” from the menu on the left → Click the link “Update” → Update the required fields → Click the button “Update” → Update status.

Delete

There are eight different categories in FSR where the user can delete the details.

The navigation in each category is shown in Table 12.

Table 12: Navigation to delete an entry

Category	Navigation
Courses Taught	Start by selecting “Courses Taught” from the menu on the left → Click the course you want to delete → Click the link “Delete” → Confirm delete by choosing option “Yes” and click the button “Ok” → Delete status.
Courses Developed	Start by selecting “Courses Developed” from the menu on the left → Click the course you want to delete → Click the link “Delete” → Confirm delete by choosing option “Yes” and click the button “Ok” → Delete status.
Students Supervised	Start by selecting “Students Supervised” from the menu on the left → Click the student you want to delete → Click the link “Delete” → Confirm delete by choosing option “Yes” and click the button “Ok” → Delete status.
Publications	Start by selecting “Publications” from the menu on the left → Click the publication you want to delete → Click the link “Delete” → Confirm delete by choosing option “Yes” and click the button “Ok” → Delete status.
Presentations	Start by selecting “Presentations” from the menu on the left → Click the presentation you want to delete → Click the link “Delete” → Confirm delete by choosing option “Yes” and click the button “Ok” → Delete status.
Grants	Start by selecting “Grants” from the menu on the left → Click the grant you want to delete → Click the link “Delete” → Confirm delete by choosing option “Yes” and click the button “Ok” → Delete status.
Proposals	Start by selecting “Proposals” from the menu on the left → Click the proposal you want to delete → Click the link “Delete” →

	Confirm delete by choosing option “Yes” and click the button “Ok” →Delete status.
Miscellaneous	Start by selecting “Miscellaneous” from the menu on the left → Click the miscellaneous you want to delete→ Click the link “Delete” → Confirm delete by choosing option “Yes” and click the button “Ok” →Delete status.

Search

There are eight different categories in FSR where the user can search the details.

The navigation in each category is shown in Table 13.

Table 13: Navigation to search an entry

Category	Navigation
Courses Taught	Start by selecting “Courses Taught” from the menu on the left → Type the keyword in the “Search Keyword” textbox→ Choose the search type → Click the button “Go” →Search results.
Courses Developed	Start by selecting “Courses Developed” from the menu on the left → Type the keyword in the “Search Keyword” textbox→ Choose the search type → Click the button “Go” →Search results.
Students Supervised	Start by selecting “Students Supervised” from the menu on the left → Type the keyword in the “Search Keyword” textbox→ Choose the search type → Click the button “Go” →Search results.
Publications	Start by selecting “Publications” from the menu on the left → Type the keyword in the “Search Keyword” textbox→ Choose the search type → Click the button “Go” →Search results.
Presentations	Start by selecting “Presentations” from the menu on the left → Type the keyword in the “Search Keyword” textbox→ Choose the search type → Click the button “Go” →Search results.
Grants	Start by selecting “Grants” from the menu on the left → Type the keyword in the “Search Keyword” textbox→ Choose the search type → Click the button “Go” →Search results.
Proposals	Start by selecting “Proposals” from the menu on the left → Type the keyword in the “Search Keyword” textbox→ Choose the search type → Click the button “Go” →Search results.
Miscellaneous	Start by selecting “Miscellaneous” from the menu on the left → Type the keyword in the “Search Keyword” textbox→ Choose the search type → Click the button “Go” →Search results.

Reports

FSR can generate PDF, and HTML reports. The navigation for generating these reports is shown in Table 14.

Table 14: Navigation for report generation

Report type	Navigation
PDF	Start by selecting “Reports” from the menu on the left → Click the button “Go” → Choose the starting date, ending date, and the categories → Click the button “Show” → Edit the content if necessary → Click the button “Generate PDF” → Click the button “Confirm generate PDF” → PDF report opens in a new window.
HTML	Start by selecting “Reports” from the menu on the left → Click the button “Go” → Choose the starting date, ending date, and the categories → Click the button “Show” → Edit the content if necessary → Click the button “Generate PDF” → Click the link “View HTML” → HTML report opens in a new window.

Inserting special symbols

The following steps show how to use the special symbols:

1. Click the link “Help with Symbols” from the menu on the left.
2. Verify that a new page with symbols and codes opens in the browser.
3. Copy the corresponding code for the symbol to be inserted into the database.
4. Paste the code wherever the symbol needs to be appeared.

APPENDIX II

SOURCE CODE

The compact disk labeled as “Faculty Scholastic Report” contains the source code of the project.