ENHANCEMENT OF SCHOOL OF NURSING DATABASE

A Thesis
Presented to the
Faculty of
San Diego State University

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

by
Mandeep B. Vasani
Summer 2010
SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the

Thesis of Mandeep B. Vasani:

Enhancement of School of Nursing Database

__________________________
Carl Eckberg
Chair
Department of Computer Science

__________________________
William A. Root
Department of Computer Science

__________________________
Catherine Todero
Department of School of Nursing

June 29, 2010
Approval Date
Copyright © 2010

by

Mandeep B. Vasani

All Rights Reserved
DEDICATION

I thank my grand-parents, parents, family and faculty of CS Department at SDSU for their faith in me and allowing me to be as ambitious as I wanted. It was under their watchful eye that I gained so much drive and ability to tackle challenges head on. I dedicate my work to my father Mr. Bharatbhai Vasani, my mother Mrs. Ila Bharatbhai Vasani, my brother Sandeep Vasani and friends without their endless support it would be a difficult task to achieve.
ABSTRACT OF THE THESIS

Enhancement of School of Nursing Database
by
Mandeep B. Vasani
Master of Science in Computer Science
San Diego State University, 2010

The goal for the enhancement of existing web based database application for the School of Nursing department at San Diego State University is to aid them in management of new student application process, track student application, track student’s course via MyMap feature, student’s graduation completion and generate reports required by school of nursing to submit statistical information in different survey forms given by government bodies. This application will serve the administrative personnel in the school of nursing by providing them easy access and better control of student information specific to their department.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INTRODUCTION</td>
</tr>
<tr>
<td>1.1</td>
<td>Background</td>
</tr>
<tr>
<td>1.2</td>
<td>Existing Database System</td>
</tr>
<tr>
<td>1.3</td>
<td>Enhanced Database System Features</td>
</tr>
<tr>
<td>2</td>
<td>TECHNOLOGIES USED</td>
</tr>
<tr>
<td>2.1</td>
<td>PHP: Hypertext Preprocessor</td>
</tr>
<tr>
<td>2.2</td>
<td>MySQL</td>
</tr>
<tr>
<td>2.3</td>
<td>JavaScript/Ajax</td>
</tr>
<tr>
<td>2.4</td>
<td>ROHAN Web Server</td>
</tr>
<tr>
<td>3</td>
<td>ARCHITECTURE</td>
</tr>
<tr>
<td>3.1</td>
<td>MVC Overview</td>
</tr>
<tr>
<td>3.2</td>
<td>Sondata system MVC details</td>
</tr>
<tr>
<td>4</td>
<td>SYSTEM DESIGN AND IMPLEMENTATION</td>
</tr>
<tr>
<td>4.1</td>
<td>System Overview</td>
</tr>
<tr>
<td>4.2</td>
<td>Table Design</td>
</tr>
<tr>
<td>4.3</td>
<td>Entity Relationship model</td>
</tr>
<tr>
<td>4.3.1</td>
<td>User Login</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Son Courses and MyMap</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Student System</td>
</tr>
<tr>
<td>4.4</td>
<td>Implementation Details</td>
</tr>
<tr>
<td>4.5</td>
<td>GUI</td>
</tr>
</tbody>
</table>
5 FUTURE ENHANCEMENTS .................................................................37
REFERENCES .................................................................................38
LIST OF TABLES

Table 4.1. SONDATA Table Description.................................................................11
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 3.1</td>
<td>Model view controller</td>
<td>7</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>ER diagram for user login system</td>
<td>12</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>ER diagram for SON courses and MyMap details</td>
<td>13</td>
</tr>
<tr>
<td>Figure 4.3</td>
<td>ER diagram for student details</td>
<td>14</td>
</tr>
<tr>
<td>Figure 4.4</td>
<td>Registration screen</td>
<td>16</td>
</tr>
<tr>
<td>Figure 4.5</td>
<td>Login screen</td>
<td>16</td>
</tr>
<tr>
<td>Figure 4.6</td>
<td>Password retrieval form</td>
<td>17</td>
</tr>
<tr>
<td>Figure 4.7</td>
<td>Student welcome screen</td>
<td>18</td>
</tr>
<tr>
<td>Figure 4.8</td>
<td>Student personal information screen</td>
<td>19</td>
</tr>
<tr>
<td>Figure 4.9</td>
<td>Student emergency information screen</td>
<td>20</td>
</tr>
<tr>
<td>Figure 4.10</td>
<td>Student high school information screen</td>
<td>21</td>
</tr>
<tr>
<td>Figure 4.11</td>
<td>Student pre-requisite courses information screen</td>
<td>22</td>
</tr>
<tr>
<td>Figure 4.12</td>
<td>Student military information screen</td>
<td>23</td>
</tr>
<tr>
<td>Figure 4.13</td>
<td>Students statistical information screen</td>
<td>24</td>
</tr>
<tr>
<td>Figure 4.14</td>
<td>Students application summary screen</td>
<td>25</td>
</tr>
<tr>
<td>Figure 4.15</td>
<td>Students application summary screen (continued)</td>
<td>26</td>
</tr>
<tr>
<td>Figure 4.16</td>
<td>Admin welcome screen</td>
<td>27</td>
</tr>
<tr>
<td>Figure 4.17</td>
<td>Admin application repository screen</td>
<td>28</td>
</tr>
<tr>
<td>Figure 4.18</td>
<td>Admin application repository screen</td>
<td>29</td>
</tr>
<tr>
<td>Figure 4.19</td>
<td>Admin set application period screen</td>
<td>30</td>
</tr>
<tr>
<td>Figure 4.20</td>
<td>Admin view MyMap screen</td>
<td>31</td>
</tr>
<tr>
<td>Figure 4.21</td>
<td>Admin update MyMap screen</td>
<td>32</td>
</tr>
<tr>
<td>Figure 4.22</td>
<td>Admin add courses to MyMap screen</td>
<td>33</td>
</tr>
<tr>
<td>Figure 4.23</td>
<td>Admin add grade students screen</td>
<td>34</td>
</tr>
<tr>
<td>Figure 4.24</td>
<td>Admin projection screen</td>
<td>35</td>
</tr>
<tr>
<td>Figure 4.25</td>
<td>Admin demographic report screen</td>
<td>36</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to Dr. Carl Eckberg and Dr. Todero for giving me this opportunity to work on their project and for providing constant support and motivation.

I am grateful to Dr. William A. Root for serving on my thesis committee and for his help and co-operation.

I would also like to express my sincere thanks to Dr. Charlie Nicodemus and Professor Joanne Ling of School Of Nursing department for trusting me with this project and providing his kind support, guidance and co-operation throughout the project.

Finally, I would also like to thank Vishal Mahaveer alumni of Computer Science department for trusting me with this project and giving me head start with detailed knowledge transfer about existing system.
CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

School of Nursing (SON) department of San Diego State University (SDSU) maintains students’ personal information in Microsoft Access database which is maintained by internal faculty members. SON faculty also maintains several excel sheets to capture information related to new student application. SDSU Student Information Management System (SIMS) maintains and keeps track of student course work, grades etc.

SON department is required to submit detailed reports regarding statistical information in the surveys conducted by government bodies. One of the very common report required is related to Demographic details, student classification on Age, Gender, Ethnicity, Nationality, Military experience etc. Due to the scattered nature of data and lack of right information it was difficult for them to work on these forms.

Hence there was a need for a system to capture and maintain student information in a central single repository with a friendly interface for querying the system and generating useful information for the statistical information. The trivial requirements to be fulfilled by the system were:

- Online System for students to apply for SON.
- Auto generation of MyMap (My Major Academic Plan) for new students accepted.
- Online Grading System.
- Centralize and cater data.
- Automatic report generation.
- Support for high end availability to access data.

The application is hosted on San Diego State University’s ROHAN web server and can be accessed with the domain name http://www.rohan.sdsu.edu/~sondata [1].
1.2 EXISTING DATABASE SYSTEM

SON database system termed as SONDATA is a Web based database system, which is hosted on SDSU’s Rohan server. SONDATA was implemented using HTML/JavaScript for client side scripting and PHP for server side scripting with MySQL as database. It supports four different types of users. The four access roles of the system are Superadmin, Admin, Query and Read-only. Superadmin is the single topmost user level with complete control of the system. System can have one or more Admin users who can add or modify student information, generate required reports, etc. Query access users can only add new students and generate reports etc. Read-only users have most restricted access and are both at the same level with READ-ONLY access to system.

As per SON current process, new students submit application in paper form. SON faculty stores all student details in excel sheet for processing each application. These excel sheet are stored on faculty’s personal computer, hence not accessible by everyone. Once student is accepted by SON, Admin or Query level users will use excel sheet to feed data into SONDATA. Due to this for student application details SONDATA is dependent on faulty member’s excel sheets.

SIMS database maintains students MyMap and course work details. At the end of each semester, SON will get an extract from SIMS database with student details containing course enrolled and grade obtained. Admin or Query level users will manually read SIMS extract and feed data into SONDATA. Again, there is dependency on SIMS extract.

Due to such dependencies and manual data entry the report generated where not accurate enough and also due to scattered source data it was also cumbersome to validate data generated by reports. Hence, there was need to re-design the existing system to eliminate dependencies and manual effort.

1.3 ENHANCED DATABASE SYSTEM FEATURES

1. Online Student application system: Online portal system to replace the existing paper based application, which facilitates new students to register in SONDATA system, fill SON’s new student application form, submit it online and track the application status; also existing students can view auto generated MyMap and track progress through MyMap.
2. For SON’s faculty it should facilitate to download all new student application details into excel sheet which would be easily used for processing application, processing application status, generate reports, etc.

3. MyMap maintained and controlled within SONDATA: Feature to create/update MyMap for semester, auto generate MyMap for each enrolled student, report future semester projection i.e. number of students eligible for each course, etc.

4. SON Courses detail: Features to add/update/delete courses and add/delete pre-requisite courses.

5. Grading System: Features for grading student’s courses per semester basis, monitor students MyMap and make appropriate updates to accommodate re-take of courses, adjust student’s Cohort etc.
CHAPTER 2

TECHNOLOGIES USED

SONDATA system is developed using LAMP technology. LAMP stands for Linux (operating system), Apache (HTTP server), MySQL (database software) and PHP scripting language for web development. LAMP is a popular open source web platform primarily used to build, deploy and run dynamic web sites. Features of LAMP stack solution are:

- Low cost: All the entities Linux, Apache, MySQL and PHP are available under Free software or Open source software license.
- Easy to develop: PHP has wide range API’s for web development and MySQL operations.
- Easy to deploy: Apache has got PHP’s standard module, which just requires uploading PHP files.

2.1 PHP: HYPERTEXT PREPROCESSOR

PHP stands for “Hypertext Preprocessor” which is an Open Source general purpose scripting language, designed primarily for rapid dynamic web development [2]. In web development, PHP is used for server side scripting language i.e. web server uses PHP processor module to transform PHP code into html web page, the resulting web page is sent to the browser for viewing. PHP primarily usage is for rapid web development and all modern day web servers have PHP processor module. However, PHP can also be used as general purpose programming language, for command line application to perform native operating system functions or graphical user based application.

ROHAN server has currently installed PHP 5.2.5 version. PHP’s session management feature, database communication APIs and form validating functions are extensively used in SONDATA. Few of the PHP plus points which makes ideal of SONDATA are:

- PHP is Open Source, available under free software license and Open Source license.
- PHP is compatible with most of the operating systems such as Linux family, Microsoft Windows, Mac OS X and many others.
• Most of the well known web server viz. Apache, Tomcat, IIS etc supports PHP readily and has PHP processor module.
• PHP is easy to embed in HTML file and supports dynamic HTML pages.
• PHP supports for wide range of databases including ORACLE, Sybase, and MySQL.

2.2 MySQL
MySQL is one of the most popular Open Source relational databases and high-end commercial database servers. It includes ability to manage very large quantities of data, high performance and availability, robust transactional support, strong data protection and most important comprehensive application development. Libraries for accessing MySQL databases are available in all major scripting languages with language specific APIs [3].

ROHAN server has currently installed MySQL version 5.1.35. Following features makes MySQL as a good choice for SONDATA:
• MySQL is Open source which means we can use it without paying anything. It is distributed under GNU General Public License.
• MySQL is compatible with most common operating systems viz. Linux, Unix, Macintosh, Microsoft Windows, etc.
• MySQL is a relational data base management system.
• MySQL is easily scalable, flexible supporting robust applications.
• MySQL is very fast, reliable and easy to use.
• MySQL is highly suitable for accessing databases on the internet. It is easy to build web applications with MySQL and scripting languages like PHP.

2.3 JavaScript/AJAX
JavaScript is a popular client side scripting language widely supported in most web browsers. The primary use of JavaScript is to add interactivity to HTML pages. It is light weight programming language embedded directly into HTML pages [4]. JavaScript when used with Web Browser’s Document Object Model (DOM) produces powerful dynamic web based application system.

AJAX stands for Asynchronous JavaScript and XML, is web development methodology used on Client-side to create dynamic interactive web applications. AJAX based web application retrieve data from server in background without reloading the existing web page i.e. part of web page data is modified with interfering with display and behavior of
web page. In AJAX, browser request data using `XMLHttpRequest` object. Data received is accessed by JavaScript and dynamically displayed using DOM and CSS.

Hence, JavaScript and XMLHttpRequest together provides an interface for client-side browser to interact with server asynchronously and load only small specific parts of a web page.

### 2.4 ROHAN Web Server

ROHAN is the academic computing web server at San Diego State University and is administered by the university library staff. ROHAN’s system is SunFire 4900 with Solaris 10 operating system having 8 dual core processors. ROHAN's World Wide Web (WWW) service with streaming audio/video is hosted on a Sun T5120 server (Ultra Sparc T2 processor, 16GB RAM), accessed by `www.rohan.sdsu.edu` as the domain name and powered by Apache web software [1].
CHAPTER 3

ARCHITECTURE

SONDATA system uses *Model-View-Controller* (MVC) as architecture and design pattern, as depicted in Figure 3.1. MVC is fundamental design pattern for separation of user interface logic from business logic [5]. Each terms of M-V-C are further explained below:

The *Model* manages information and notifies observer when information updated. Model handles domain-specific representation of the data, uses domain logic to add meaning to raw data received. When a model changes its state, notifies the views so it would be refreshed.

The *View* simply renders the model into a suitable form for user interactions.

The *Controller* receives input from user, interprets the input and informs model and/or view to act on input received.

![Figure 3.1. Model view controller.](image)

As depicted in diagram, view and controller both depends on Model directly. However, Model is independent from both controller and view, due to this model can be built and tested separately from visual representation. In general view and controller can be designed independent or dependent where controller drives the view. However, for Web applications separation between view (browser) and controller (server side handling HTTP requests) is very clear.
3.1 MVC OVERVIEW

- User interacts with system via user interface (view)
- The controller handles the input event from user interface and converts into appropriate user action, which model can understand and process it.
- The controller notifies the model of the user action, resulting in change of model state.
- View is automatically notified by model of changes in state, which requires screen updates.
- The user interface waits until next user action, at which cycle restarts.

MVC when used for web applications has the view either as HTML or XHTML generated by the application. The controller receives GET or POST input and decides what to do with it, handing over to domain objects (i.e. the model) that contain the business rules and know how to carry out specific tasks such as processing a new subscription.

3.2 SONDATA SYSTEM MVC DETAILS

*Model* is collection of PHP script files intended to separate the database from user interface.

*View* is represented by HTML which is rendered by browser.

*Controller* is Apache web server which processes the PHP files, converts it into HTML web page and forwards the converted web page as response to browser for rendering. When user action is performed on web page, it receives data either in GET or POST form which forwarded to appropriate PHP file to process user action.
CHAPTER 4

SYSTEM DESIGN AND IMPLEMENTATION

4.1 SYSTEM OVERVIEW

SONDATA system supports 2 different levels of user access. Each user level has permissions to perform predefined operations. The user-level category:

- ADMIN
- STUDENT

STUDENT OPERATIONS:

- Register to SONDATA – student level users are required to register to system by using unique username and providing personal email id.
- Add/Modify application details:
  SON’s new student application is divided into following information criteria:
  1. Personal Information – regarding gender, date of birth, ethnicity, nationality etc.
  2. Emergency Contact – information of the person to be contacted in case of emergency.
  3. High School Detail – student’s high school details regarding school name, address, year degree awarded etc.
  4. Previous College/Universities – list of all schools, colleges and universities attended by student in past.
  5. Pre-requisite courses information – grades obtained in the pre-requisite courses namely Psychology, Sociology, Chemistry, etc.
  6. Military Information – information regarding military experience, military service, military branch, etc.
  7. Statistical Information – health care experience and TEAS score details.

SON faculty evaluates each student depending on pre-requisite courses, health care experience, military experience, TEAS score, etc and assigns cumulative scores. This score is used to decide whether student is selected to join SON.

- View Application summary – student can view the entire application form any time. Application summary details could be printed by using the print feature.
• Submit application – student can submit the application once all the necessary information is filled. Student will be allowed to submit application only if application period is open.

• Change Password – feature for admin user to update their password

ADMIN OPERATIONS:

• Set Application Period – SON accepts new student application once a year for a specific interval of days. Admin user will be able to set the application period. Admin user can control the application period completely i.e. system provides options to either close the period earlier than schedule or extending the period beyond scheduled period, etc.

• Process New Student Applications – feature is provided to view new student records. There is an option to print minimum application details like First Name, Last Name, Red ID, etc. There is another feature to download new application details into pre-defined formatted Excel sheet. The excel sheet is used by SON faculty to evaluate total points for each student.

• Add/Modify MyMap – SON has pre-defined MyMap which each student has to follow and complete the course work. At any time, there is always only one active MyMap in SON system. Newly accepted student’s MyMap will be auto-generated using the MyMap configured in system.

• Add/Modify/Delete Courses – maintain SON’s course details.

• Add/Modify/Delete Pre-requisite courses – maintain SON’s pre-requisite course details.

• Grade Students – at end of each semester admin user has to input grades obtained by each student.

• Reports:
  1. Next Semester projection – project number of students eligible for each particular course using MyMap as source.
  2. COHORT report – number of students accepted to SON in each semester.
  3. Courses report – semester wise report which provides information regarding student’s course grade and cumulative GPA for this particular semester.
  4. Demographic report – report which provides student classification on the basis of Gender, date of birth, ethnicity, nationality, etc.

• Add New ‘Admin Level’ user – SONDATA system is designed where Student level users can register and create account. However, Admin level user has to be created by existing admin users.

• View All Users – report displays existing user’s username, email and role.

• Display Logs – SONDATA system logs the details for few selected operations.
• Change Password – feature for admin user to update their password.

4.2 TABLE DESIGN

The following section discusses in detail the table design for the SONDATA system. Table 4.1 shows the list of tables with their short description and Primary key.

Table 4.1. SONDATA Table Description

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Description</th>
<th>Primary Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>login_details</td>
<td>Stores username, password, emailed and role.</td>
<td>username</td>
</tr>
<tr>
<td>user_personal_details</td>
<td>Stores user’s First Name and Last name.</td>
<td>Auto Generated Integer</td>
</tr>
<tr>
<td>enum_detail</td>
<td>Stores SONDATA specific fixed string values.</td>
<td>Red ID</td>
</tr>
<tr>
<td>student_personal</td>
<td>Stores personal details for new student application</td>
<td>Auto generated Integer, First Name, Last Name</td>
</tr>
<tr>
<td>student_emergency</td>
<td>Stores emergency details for new student application</td>
<td></td>
</tr>
<tr>
<td>student_highschool_detail</td>
<td>Stores high school details for new student application</td>
<td></td>
</tr>
<tr>
<td>student_all_school_detail</td>
<td>Stores previous colleges/universities attended by students.</td>
<td></td>
</tr>
<tr>
<td>student_pre_req_classes</td>
<td>Stores details regarding pre-requisite courses required to apply for SON.</td>
<td></td>
</tr>
<tr>
<td>student_military</td>
<td>Stores details regarding military experience of the students.</td>
<td></td>
</tr>
<tr>
<td>student_stat_info</td>
<td>Stores details regarding Health care experience and TEAS scores.</td>
<td></td>
</tr>
<tr>
<td>student_application_status</td>
<td>Maintains entire history of student applications.</td>
<td></td>
</tr>
<tr>
<td>adm_course_detail</td>
<td>Stores SON course details.</td>
<td>CourseNo</td>
</tr>
<tr>
<td>adm_prereq_course_detail</td>
<td>Stores SON pre-requisite course details.</td>
<td></td>
</tr>
<tr>
<td>adm_mymap</td>
<td>Stores SON’s MyMap detail</td>
<td></td>
</tr>
<tr>
<td>student_courses_detail</td>
<td>Stores individual student’s MyMap and grading information.</td>
<td></td>
</tr>
<tr>
<td>Logdb</td>
<td>Stores logging details for SONDATA system</td>
<td></td>
</tr>
</tbody>
</table>
4.3 ENTITY RELATIONSHIP MODEL

The Entity Relationship Diagram graphically represents the conceptual schema of a database. Basically, it describes tables and relationship between the tables used in a database. The SONDATA system is divided into 3 broad categories, Figures 4.1 to 4.3 depicts ER diagrams for each individual sub system.

4.3.1 User Login

This sub-system maintains information regarding each user’s credentials and role assigned to user. When new user is added to SONDATA, either using self-register feature i.e. Student level user) or admin user creates new Admin level user, following tables are updated with appropriate details.

![Figure 4.1 ER diagram for user login system.]

4.3.2 Son Courses and MyMap

This sub-system maintains information regarding SON courses and pre-requisite courses and MyMap details.

4.3.3 Student System

This sub-system maintains information related to students i.e. student application information, application status, MyMap, grade etc.
Figure 4.2 ER diagram for SON courses and MyMap details.
Figure 4.3 ER diagram for student details.
4.4 IMPLEMENTATION DETAILS

This section describes few of the important implementation details of the SONDATA system. Detailed information regarding each user level operations was provided in Section 4.1.

Session Management:

HTTP (Hypertext Transfer Protocol) is a stateless; client’s browser establishes new TCP (Transmission Control Protocol) to server with either HTTP (GET or POST) web request. Web Server by design doesn’t rely on established TCP connection for more than one HTTP web request.

Session management is used to make HTTP’s stateless protocol support session state. Session states are maintained in HTTP cookies. HTTP cookie is a text string which contains multiple name-value pair information. The trivial information cookie contains is Unique Session Identifier sent back by server during very first initial authentication and connection. Now on, client browser would use unique session id to exchange information with server.

SONDATA system by default designed to support 60mins of idle session time i.e. session will expire after 60mins of no action from user on client’s browser. After which, user will have to re-login to the system.

Persistent Database Connection:

Database connection is expensive and can take proportionally long time to establish relative to actual operations performed on them. To avoid it, SONDATA system uses Persistent connections. Persistent connection is not closed once the sql scripts execution ends, so for next database operation connection will be re-used rather than establishing new one. This significantly reduces the overhead of creating a fresh connection every time, as unused connections are cached and ready to be reused. PHP’s mysqli extension is used for supporting persistence connection.

AJAX techniques:

SONDATA system uses AJAX methodology for web pages where a small part of web page is required to be updated. For e.g.: Update Course details feature lists all the courses in a single web page. So with every course update page would be reloaded i.e. say user needs to update say 2 courses than 2 page reloads would take place. To avoid the multiple page reload, course update details are implemented via AJAX technique.
4.5 GUI

SONDATA system provides a simple and easy to use GUI. The user interfaces were built using HTML and Cascading Style Sheet (CSS). The operation accessible by each user depends on the access level of logged in user. The remainder of this section shows some screenshots of user interfaces in Figure 4.4 through Figure 4.25.

Figure 4.4 Registration screen.

Figure 4.5 Login screen.
Figure 4.6 Password retrieval form.
Figure 4.7 Student welcome screen.
Figure 4.8 Student personal information screen.
Figure 4.9 Student emergency information screen.
Figure 4.10 Student high school information screen.
**Figure 4.11 Student pre-requisite courses information screen.**
Figure 4.12 Student military information screen.
Figure 4.13 Students statistical information screen.
Figure 4.14 Students application summary screen.
Figure 4.15 Students application summary screen (continued).
Figure 4.16 Admin welcome screen.
Figure 4.17 Admin application repository screen.
Figure 4.18 Admin application repository screen.
Figure 4.19 Admin set application period screen.
Figure 4.20 Admin view MyMap screen.
**Figure 4.21 Admin update MyMap screen.**

<table>
<thead>
<tr>
<th>Semester</th>
<th>Course Sem</th>
<th>Course No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester 1</td>
<td>1</td>
<td>NURS 200</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>NURS 202</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>NURS 206</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>NURS 208</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>NURS 397</td>
</tr>
<tr>
<td>Semester 2</td>
<td>2</td>
<td>NURS 300</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>NURS 302</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>NURS 304</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>NURS 397</td>
</tr>
<tr>
<td>Semester 3</td>
<td>3</td>
<td>NURS 307</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>NURS 354</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>NURS 356</td>
</tr>
</tbody>
</table>
Figure 4.22 Admin add courses to MyMap screen."
Figure 4.23 Admin add grade students screen.
**Figure 4.24 Admin projection screen.**

<table>
<thead>
<tr>
<th>Semester 1</th>
<th>Students Eligible</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Course No</strong></td>
<td><strong>Course Title</strong></td>
</tr>
<tr>
<td>NURS 202</td>
<td>Client Assessment</td>
</tr>
<tr>
<td>NURS 206</td>
<td>Fundamentals of Nursing Practice</td>
</tr>
<tr>
<td>NURS 208</td>
<td>Pathophysiology</td>
</tr>
<tr>
<td>NURS 397</td>
<td>Preparation for Clinical Practice</td>
</tr>
</tbody>
</table>

**Semester 2**

<table>
<thead>
<tr>
<th><strong>Course No</strong></th>
<th><strong>Course Title</strong></th>
<th>Students Eligible</th>
</tr>
</thead>
<tbody>
<tr>
<td>NURS 300</td>
<td>Nursing Care of the Acutely Ill Adult</td>
<td>2</td>
</tr>
<tr>
<td>NURS 302</td>
<td>Nurse-Client Relationships: Cultural and Mental Health Concepts</td>
<td>2</td>
</tr>
<tr>
<td>NURS 304</td>
<td>Clinical Pharmacology in Nursing Practice</td>
<td>2</td>
</tr>
</tbody>
</table>

**Semester 4**

<table>
<thead>
<tr>
<th><strong>Course No</strong></th>
<th><strong>Course Title</strong></th>
<th>Students Eligible</th>
</tr>
</thead>
<tbody>
<tr>
<td>NURS 410</td>
<td>Gerontological Nursing</td>
<td>1</td>
</tr>
<tr>
<td>NURS 416</td>
<td>Psychiatric-Mental Health Nursing</td>
<td>1</td>
</tr>
</tbody>
</table>
Figure 4.25 Admin demographic report screen.
CHAPTER 5

FUTURE ENHANCEMENTS

SONDATA system is designed using MVC model where model consists of PHP scripts. Thus to add new features to SONDATA, one requires to develop new PHP scripts for UI and business logic processing. Additionally, MySQL database schema can be extended to support new tables. Following are list of future works that could be considered:

- Support to integrate details related to Graduate level of students.
- Support for new students competition, which is currently carried out manually by SON faculty.
- Robust reporting tool:
  - Feature for Admin level user to select report data and store selected configuration.
  - Share reports between Admin users.
  - Save generated report in database.
- Automate Grading i.e. SIMS has functionality to export SON students data into excel sheet. Develop excel sheet parser to read data and populate appropriate tables.
REFERENCES

WORKS CITED


WORKS CONSULTED


