2007

Computer science graduate project management system

Jianyuan Huang

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

Part of the Databases and Information Systems Commons

Recommended Citation
Huang, Jianyuan, "Computer science graduate project management system" (2007). Theses Digitization Project. 3250.
https://scholarworks.lib.csusb.edu/etd-project/3250

This Project is brought to you for free and open access by the John M. Pfau Library at CSUSB ScholarWorks. It has been accepted for inclusion in Theses Digitization Project by an authorized administrator of CSUSB ScholarWorks. For more information, please contact scholarworks@csusb.edu.
COMPUTER SCIENCE
GRADUATE PROJECT MANAGEMENT SYSTEM

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

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

by
Chien-Yuan Huang
June 2007
COMPUTER SCIENCE

GRADUATE PROJECT MANAGEMENT SYSTEM

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

by
Chien-Yuan Huang
June 2007

Approved by:

Dr. Keith Schubert, Chair,
Computer Science

Dr. Josephine G. Mendoza

Dr. David Turner

June 5, 2007
ABSTRACT

This project is a development and tracking system for graduate students in the Department of Computer Science of CSUSB. Students who are thinking about their project topics can get some ideas from either the faculty's research areas or from projects done by previous students. Students who are working on projects can publish their proposals, abstracts and final reports on this website. This project covers front-end web site development, back-end database design and security. This website provides secure access to information about ideas for project, status of on-going projects, and reports of finished projects. Faculty members can list their research areas, publications, ideas for graduate projects and monitor status of the projects of their advisees.
ACKNOWLEDGMENTS

First, I would like to express my special thanks to my advisor, Dr. Keith Schubert who assisted me in every aspect of this project.

My sincere thanks to my committee members, Dr. Josephine G. Mendoza and Dr. David Turner who have been very enthusiastic with my project, gave me valuable advice and encouraged me to complete the project.

Special thanks to the rest of the CSCI faculty, Dr. Arturo Concepcion, Dr. Owen Murphy, Dr. George Georgiou, Dr. Kay Zemoudeh, Dr. Richard Botting, and Dr. Yasha Karant who have given the knowledge and support to bring this project to fruition.

I am also grateful to my parents for their guidance and sacrifice and giving me a lot of encouragement and support to pursue my education.

Last but not the least, I thank my girl friend (Erica Chu) who is happy for me achieving my goal and dream.
TABLE OF CONTENTS

ABSTRACT .............................................. iii
ACKNOWLEDGMENTS ............................... iv
LIST OF TABLES ................................. vii
LIST OF FIGURES ................................. viii

CHAPTER ONE: INTRODUCTION

1.1 Introduction ................................. 1
1.2 Purpose of the Project ................. 1
1.3 Limitations .................................. 2
1.4 Organization of the Project .......... 2

CHAPTER TWO: SOFTWARE REQUIREMENTS SPECIFICATION

2.1 Introduction ................................. 3
   2.1.1 Scope .................................. 3
2.2 Overall Description ....................... 3
   2.2.1 Product Perspective ................. 4
   2.2.2 Product Functions .................... 7
   2.2.3 User Categorizes .................... 8
2.3 Specific Requirements ..................... 9
   2.3.1 External Interface Requirements ... 9
   2.3.2 Functional Requirements ............. 9
   2.3.3 Performance Requirements .......... 10
   2.3.4 Software System Requirements .... 10

CHAPTER THREE: DESIGN AND IMPLEMENTATION

3.1 System ..................................... 11
LIST OF TABLES

Table 1. Database Table Schema ................. 16
Table 2. Pseudo Code of Account Maintenance .... 20
Table 3. Pseudo Code to Search
         Project Information .................... 23
Table 4. Pseudo Code to Search
         Account Information .................... 25
Table 5. Pseudo Code to Maintain
         Project Information .................... 27
Table 6. Pseudo Code to Maintain Project
         Document Files .......................... 30
LIST OF FIGURES

Figure 1. The Use Case Diagram ................. 6
Figure 2. Class Diagram of Public Functions and JAVA Bean .................. 13
Figure 3. Entity Relationship Diagram ........... 15
Figure 4. UML Model for Account Maintenance . . . . 19
Figure 5. UML Model for Search Completed Project Information .............. 22
Figure 6. UML Model for Search Faculty Information ...................... 24
Figure 7. UML Model for Project Maintenance . . . . 26
Figure 8. UML Model for Update/Download Project Information .............. 29
Figure 9. Recommended Deployment ................. 32
CHAPTER ONE
INTRODUCTION

1.1 Introduction

Current faculty members and graduate students in the Department of Computer Science at California State University and others who are interested in graduate projects do not have a convenient way to access information about completed and on-going projects.

My project, "Computer Science Graduate Project Management", will provide information on ideas for project and documents of on-going and completed projects. Faculty members can list their research areas, ideas for graduate projects and monitor the progress of their advisees. Students who are developing projects can get ideas from either research projects of faculty or student projects that are already completed. Students can publish their project proposal, abstracts and final reports on this website.

1.2 Purpose of the Project

This project, "Computer Science Graduate Project Management System", implements a system which provides faculty members, graduate students and others to access appropriate project information through Internet from the
database. This helps in obtaining graduate project information, getting completed graduate project documents and monitoring ongoing graduate projects.

1.3 Limitations

This project is designed and implemented mainly to access project documents.

Several important functions have been developed such as account management, project information and documents update and project information search. Due to the complexity and time constraints, the discussion forum will be left as a future enhancement to the system.

1.4 Organization of the Project

This document is organized in five chapters: (1) Introduction, (2) Software Requirements Specification, (3) Design and Implementation, (4) Deployment and (5) Technology Highlights, Conclusion and Future Development. The appendix provides the program files used in this system.
CHAPTER TWO
SOFTWARE REQUIREMENTS SPECIFICATION

2.1 Introduction

The scope of the project will be described in Section 2.1.1. Section 2.2 provides an overall description of the product while Section 2.3 details the specific requirements.

2.1.1 Scope

The proposed web-based database system is designed to be used in the Master of Science program in the Computer Science Department. The existing graduate project documents are not maintained in a Computer Science Department database. This project uses MySQL 4.1 as a database engine and allows faculty, graduate students and others to have a secure Internet access to project information stored in the database.

2.2 Overall Description

The existing graduate project information for the graduate program in the Department of Computer Science is not maintained in a database.

This system is designed to open access to the database to the Computer Science Department faculty, graduate students and others via the Internet. The system is implemented using MySQL 4.1, and takes advantage of MySQL
4.1 features for fast performance, high reliability and ease of use. This system consists of five modules: (1) account maintenance, (2) search for completed project information, (3) search for faculty information, (4) project maintenance, (5) update/download of project information.

The system will be web-based, and will be accessible by all authorized users using any modern web browsers (e.g. IE, Netscape, Mozilla). The system provides high security that allows access by authorized users only. Communication between users at the client side and the system at the server side must be secure. User and project information must be protected, but must be convenient for access and update.

2.2.1 Product Perspective

The system will be implemented using JSP, JavaBean using Java 1.4 or higher version. Apache Tomcat 5.0 and MySQL 4.1 are required.

2.2.1.1 System Interfaces. The system with the Apache Tomcat Server with HTTP and Connector/J at the server side interacts with a user at the client side in the following manner:

1. User starts access to the database system by entering the URL in the browser.

2. System will authenticate and authorize the user
using a user ID and password. When a user is authenticated, the system will identify this user's corresponding role and direct the user to his or her home page. If authentication fails, the system will redirect the user to the login page again.

3. The authenticated and authorized user will have privileged access to all functions defined according to the user's role. Any attempt by the user to access an unauthorized page will be denied.

4. Based on the user's response, the system may access the MySQL database and return the result or message back to the user.

2.2.1.2 User Interfaces. All users may use any web browser on any platform to access from the client side. There is no restriction on the operating system used on the client side.

2.2.1.3 Software Interfaces. Software interfaces are provided with the Apache Tomcat 5.0 and MySQL 4.1. MySQL is a trademark of the MYSQL AB. Apache is a trademark of Apache Formatting Object Processor.

2.2.1.4 Communication Interfaces. Communication interfaces between the server side and the client side are implemented with Hypertext Transfer Protocol (HTTP).
Figure 1. The Use Case Diagram
2.2.2 Product Functions

There are 4 roles in this system. Figure 1 is the use case diagram showing the roles and their functions.

2.2.2.1 Login. The login function authenticates and authorizes a user with user ID and password. Depending on the user’s role, the user will be directed to his/her homepage.

2.2.2.2 Search/View Completed Projects. All roles can search and view information on completed projects. A list of previous completed projects with project title, student’s name, advisor and completed term will be provided. Users can access more detailed information on a certain project by clicking on the project details button.

2.2.2.3 View Faculty Information. All roles can view faculty information. This shows a list of faculty names and emails. Users can access more details on a particular faculty by clicking on the faculty details button.

2.2.2.4 View Project Advised. Faculty members can see a list (student name and project title) of projects by their advisees. Faculty members can access more detailed project information of a certain project by clicking on the project details button.
2.2.2.5 **View/Update Personal Information.** Faculty members, students and administrators can update their personal information on this page.

2.2.2.6 **Edit/Delete Projects.** Faculty members can edit student names, ID, project title, term for the project, as well as committee information of his/her advisees. Faculty members can also delete projects of his/her advisees.

2.2.2.7 **Create New Projects.** Faculty members can create new projects. Faculty members can create new projects by entering the student’s name, ID, project title and the quarter term the student registered for the project.

2.2.2.8 **Add/Edit/Delete Project Information.** Students can update his/her project information. Students can update, download, upload, delete project abstract, proposal, and reports written in different formats.

2.2.2.9 **Create/Delete/Edit Accounts.** An administrator can create, delete or edit faculty, student or administrator accounts.

2.2.3 **User Categorizes**

The following applications are assumed for the various users defined in the system.
• Application Users
  Include graduate students, staff, faculty members of the Department of Computer Science, CSUSB and others who are interested in graduate projects.

• System Administrator
  Installs and administers the MySQL 4.1 database system and the Apache Tomcat 5 Server.

• Application Developer
  Needs to have knowledge and skills in web application (HTML, JSP, Java Bean, Java Script), SSL, MySQL 4.1 and SQL language.

2.3 Specific Requirements

2.3.1 External Interface Requirements

After a user starts the browser by entering the system’s URL, the users sees the login page. When a user is authenticated, he or she sees all functions defined for his/her role.

2.3.2 Functional Requirements

All functions of the system must provide correct results, and all data errors or inappropriate access will not cause the functions to crash.
2.3.3 Performance Requirements

The system will be able to serve multiple users. Faculty members, graduate students, staff in the Department of Computer Science and others can simultaneously access the Apache Tomcat server and the MySQL database server.

2.3.4 Software System Requirements

2.3.4.1 Reliability. The system should handle failure when the MySQL database server is down. The system must provide sufficient connection to the database system to maintain high availability.

2.3.4.2 Security. The system must be protected from any security risks. The MySQL database server, as well as the web server, must be protected. A user can only access authorized function pages. Transmission over the Internet must be encrypted.

2.3.4.3 Maintainability. The web and database system should be designed to cope with changes that will not require a significant work.

2.3.4.4 Portability. The system should be able to expand its service to different types of project or be used in different departments in the university or be installed on different platforms without significant change.
CHAPTER THREE
DESIGN AND IMPLEMENTATION

3.1 System

The web-based graduate project development system for the Computer Science graduate program consists of four main system components: (1) Authentication and Authorization, (2) Update and Retrieval of Information from the MySQL Database, (3) Processing, and (4) Connection and Transmission. These are discussed in detail in the following sections.

3.1.1 Authentication and Authorization

This component authenticates a user with a user ID and password at login. The passwords are encrypted before saving to the database by using MD5 hash. Based on the user's role, it will grant the user access for the authorized pages. When a user attempts to access a secure web page, this component will verify the access privilege required for the page. It will log the user out if the user does not have the appropriate privilege.

3.1.2 Update and Retrieval of Information from MySQL Database

This component retrieves and updates information from the MySQL database. It consists of statements in SQL language and is called by Java methods.
3.1.3 Processing

Processing components are carried out at the server side on the Apache Tomcat server and MySQL database server.

3.1.3.1 Processing On the Apache Tomcat Server. This component processes requests from a user browser and generates presentation results in HTML. This component consists of JSP pages. Depending on the processing logic, this component may call other system components, such as update or retrieve information from the MySQL database.

3.1.3.2 Processing On the MySQL Database Server. This component processes information on the MySQL Server. It consists of regular procedures in SQL language, which is called by other procedures.

3.1.4 Connection and Transmission

The connectivity component generates a connect pool between the Apache Tomcat Server and the MySQL Server. After a user completes an update or retrieval of information from the database, the connection will be closed and returned back to the connection pool. The transmission component facilitates communication between a user and the system over the Internet. The transmission is secured using encryption in a Secure Socket Layer.
3.2 Architecture

The system includes two important architecture components: web application and database. These components are described below.

![Class Diagram of Public Functions and JAVA Bean](image)

3.2.1 Web Components

This architecture component includes web pages, user-defined public functions, and Java Beans. Web pages are the
core part in handling a user’s request while Java Beans assist in processing and maintaining user information throughout a web session.

3.2.1.1 Web Pages. Web pages consist of JSP pages, HTML and JavaScript. They receive requests from the client side, process and generate HTML results. To complete the tasks, they may refer to other JSP pages, user-defined public functions, Java Beans, and MySQL statements or procedures. The appendix lists the 65 files for this website.

Some JSP pages are embedded with JavaScript codes that provide convenient responses on client browsers.

3.2.2 MySQL Database

This component includes the MySQL database engine that manages four relational tables. This component is described in detail in the Design Details section.

3.2.3 Java Beans

The Java Bean used in this project is the Upload Bean. UploadBean is a JavaBean that allows uploading files from client-side web browser to server side. UploadBean can store uploaded files in a folder or a database supporting binary data. UploadBean can also limit the size of files to upload. The limit file size is set to 1 GB.
3.3 Design Details

The design to this Web-based database project consists of the designs for the MySQL database system, five web-based functionality modules and security.

3.3.1 MySQL Database System

The database system is designed to satisfy the requirements to store user and project data. The analysis of the information requirements resulted in the design of four entities for the relational model.

Figure 3. Entity Relationship Diagram
3.3.1.1 Tables. Following are the data dictionary definitions of tables used in this project. Their schema consists of an Identification (ID) field and a value field, which defines the value of the ID.

Table 1. Database Table Schema

<table>
<thead>
<tr>
<th>Table / Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STUDENT</td>
<td>TABLE</td>
<td>Stores student account information</td>
</tr>
<tr>
<td>STUDENT_ID</td>
<td>INT(9), PRIMARY KEY</td>
<td>Student ID number</td>
</tr>
<tr>
<td>PASSWORD</td>
<td>VARCHAR(32)</td>
<td>Login password</td>
</tr>
<tr>
<td>FIRST_NAME</td>
<td>VARCHAR(20)</td>
<td>First name</td>
</tr>
<tr>
<td>MID_NAME</td>
<td>VARCHAR(20)</td>
<td>Middle name</td>
</tr>
<tr>
<td>LAST_NAME</td>
<td>VARCHAR(20)</td>
<td>Last name</td>
</tr>
<tr>
<td>EMAIL</td>
<td>VARCHAR(30)</td>
<td>Email address</td>
</tr>
<tr>
<td>PHONE</td>
<td>VARCHAR(15)</td>
<td>Phone number</td>
</tr>
<tr>
<td>STATUS</td>
<td>VARCHAR(20)</td>
<td>Project status (Complete / Current)</td>
</tr>
<tr>
<td>STAFF</td>
<td>TABLE</td>
<td>Stores staff account information</td>
</tr>
<tr>
<td>STAFF_ID</td>
<td>INT(9), PRIMARY KEY</td>
<td>Staff ID number</td>
</tr>
<tr>
<td>PASSWORD</td>
<td>VARCHAR(32)</td>
<td>Login password</td>
</tr>
<tr>
<td>FIRST_NAME</td>
<td>VARCHAR(20)</td>
<td>First name</td>
</tr>
</tbody>
</table>

16
<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MID_NAME</td>
<td>VARCHAR(20)</td>
<td>Last name</td>
</tr>
<tr>
<td>LAST_NAME</td>
<td>VARCHAR(20)</td>
<td>Email address</td>
</tr>
<tr>
<td>EMAIL</td>
<td>VARCHAR(30)</td>
<td>Phone number</td>
</tr>
<tr>
<td>PHONE</td>
<td>VARCHAR(15)</td>
<td>Job title (System administrator, Support Assistant)</td>
</tr>
<tr>
<td>POSITION</td>
<td>VARCHAR(40)</td>
<td></td>
</tr>
</tbody>
</table>

**FACULTY**

- **FACULTY_ID**: INT(9), PRIMARY KEY
- **PASSWORD**: VARCHAR(32)
- **FIRST_NAME**: VARCHAR(20)
- **MID_NAME**: VARCHAR(20)
- **LAST_NAME**: VARCHAR(20)
- **EMAIL**: VARCHAR(30)
- **PHONE**: VARCHAR(15)
- **RESEARCHAREA**: VARCHAR(400)
- **ANNOUNCEMENT**: VARCHAR(400)
- **IDEAS**: VARCHAR(400)
- **OFFICE**: VARCHAR(10)

**PROJECT**

- **ID**: INT(9), PRIMARY KEY
- **FACULTY_ID**: INT

**FACULTY TABLE Stores faculty account information**

**PROJECT TABLE Stores basic project information**
3.3.2 Functionality Modules

The system features are categorized into five essential modules. Each module consists of a series of related functions and includes features and module functions. Each module may also include a UML data model and pseudo-codes. Please refer to the data dictionary for the definition of the tables, which are used in the implementation of the modules.
3.3.2.1 Account Maintenance Module. Account maintenance provides functions for staff to maintain user accounts. Other users will be update their own account information.

Staff will be able to access student, faculty or staff tables.

Account maintenance allows staff to view, create, edit, delete any student, faculty or staff account.

![Figure 4. UML Model for Account Maintenance](image-url)
Functions. The Account Maintenance consists of the following functions:

1. View user account. Administrators can display user account detail.

2. Create new account. Administrators can add new student, faculty or staff accounts.

3. Delete account. Administrator can remove student, faculty or staff accounts.

4. Update account. Administrator can modify student, faculty or staff accounts, other users can update their own account information.

Table 2. Pseudo Code of Account Maintenance

<table>
<thead>
<tr>
<th>Pseudo Code to Create/Insert Account Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get parameters for insert query from request object</td>
</tr>
<tr>
<td>If any parameter value is blank then</td>
</tr>
<tr>
<td>send an alert message and go back to input page</td>
</tr>
<tr>
<td>elseif parameter &quot;password&quot; does not match</td>
</tr>
<tr>
<td>send an alert message and go back to input page</td>
</tr>
<tr>
<td>Endif</td>
</tr>
<tr>
<td>Construct SQL query statement</td>
</tr>
<tr>
<td>If execution of the SQL statement fails then</td>
</tr>
<tr>
<td>send an alert and return</td>
</tr>
<tr>
<td>Pseudo Code to Edit Account Information</td>
</tr>
<tr>
<td>-----------------------------------------</td>
</tr>
<tr>
<td>Get parameters for update query from request object</td>
</tr>
<tr>
<td>If any parameter value is blank then</td>
</tr>
<tr>
<td>send an alert message and go back to edit page</td>
</tr>
<tr>
<td>elseif parameter &quot;password&quot; does not match</td>
</tr>
<tr>
<td>send an alert message and go back to edit page</td>
</tr>
<tr>
<td>Endif</td>
</tr>
<tr>
<td>Construct SQL query statement</td>
</tr>
<tr>
<td>If execution of the SQL statement fails then</td>
</tr>
<tr>
<td>send an alert and return</td>
</tr>
<tr>
<td>else then</td>
</tr>
<tr>
<td>update the record to STUDENT/FACULTY/STAFF table</td>
</tr>
<tr>
<td>Endif</td>
</tr>
<tr>
<td>Go back to database page</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pseudo Code to Delete Account Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get parameters for delete query from request object</td>
</tr>
<tr>
<td>Construct SQL query statement</td>
</tr>
<tr>
<td>If execution of the SQL statement fails then</td>
</tr>
<tr>
<td>send an alert and return</td>
</tr>
<tr>
<td>else then</td>
</tr>
<tr>
<td>delete the record from STUDENT/FACULTY/STAFF table</td>
</tr>
</tbody>
</table>
3.3.2.2 Search Completed Project Information. This module provides a list of completed master projects. Users can search for certain projects by entering keywords of the project title, student’s last name or part of project abstract. Users can also see project detail by clicking on the detail button.

Figure 5. UML Model for Search Completed Project Information
Functions. The Search Completed Project Information consists of the following functions:

1. Search/List Completed Projects: Users can search for projects satisfying the criteria entered or see the entire projects list.

2. View Project Detail: After getting a project list, users can view a specific project’s detail by clicking on its detail button.

PROJECT, STUDENT and FACULTY tables have the following relationships: FACULTY has 1-M relationship with PROJECT and STUDENT has 0-1 relationship with PROJECT.

Table 3. Pseudo Code to Search Project Information

<table>
<thead>
<tr>
<th>Pseudo Code to Search Project Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get parameters for select query from request object</td>
</tr>
<tr>
<td>Construct SQL query statement</td>
</tr>
<tr>
<td>If execution of the SQL statement fails then</td>
</tr>
<tr>
<td>send an alert and return</td>
</tr>
<tr>
<td>else then</td>
</tr>
<tr>
<td>select requested record(s) from PROJECT table</td>
</tr>
<tr>
<td>Endif</td>
</tr>
<tr>
<td>Go back to project database page</td>
</tr>
</tbody>
</table>

23
3.3.2.3 Search Faculty Information. This module provides a list of faculty members. Users can see a particular faculty member's ideas for project, detail information about the faculty member, research area, announcement and number of project(s) currently supervised.

<table>
<thead>
<tr>
<th>PROJECT</th>
<th>FACULTY</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID : INT(9)</td>
<td>FACULTY ID NUMBER</td>
</tr>
<tr>
<td>FACULTY_ID : INT(9)</td>
<td>FACULTY ID NUMBER</td>
</tr>
<tr>
<td>COMMITTEE_1_ID : VARCHAR(25)</td>
<td>COMMITTEEONE.ID</td>
</tr>
<tr>
<td>COMMITTEE_2_ID : VARCHAR(25)</td>
<td>COMMITTETWOD.ID</td>
</tr>
<tr>
<td>COMMITTEE_3_ID : VARCHAR(25)</td>
<td>COMMITTEE THREE.ID</td>
</tr>
<tr>
<td>BEGIN_YEAR : INT(4)</td>
<td>PROJECT BEGIN YEAR</td>
</tr>
<tr>
<td>BEGIN_TERM : VARCHAR(6)</td>
<td>PROJECT BEGIN TERM</td>
</tr>
<tr>
<td>FINISH_YEAR : INT(4)</td>
<td>PROJECT FINISHED YEAR</td>
</tr>
<tr>
<td>FINISH_TERM : VARCHAR(6)</td>
<td>PROJECT FINISHED TERM</td>
</tr>
<tr>
<td>STATUS : VARCHAR(25)</td>
<td>PROJECT STATUS</td>
</tr>
<tr>
<td>(CURRENT/COMPLETED)</td>
<td></td>
</tr>
<tr>
<td>TITLE : VARCHAR(25)</td>
<td>PROJECT TITLE</td>
</tr>
<tr>
<td>ABSTRACT : VARCHAR(200)</td>
<td>PROJECT ABSTRACT</td>
</tr>
<tr>
<td>&lt;&lt;FK=4D</td>
<td>&lt;&lt;PK&gt;&gt;FACULTY_ID</td>
</tr>
</tbody>
</table>

Figure 6. UML Model for Search Faculty Information

Functions. The Search Faculty Information consists of the following functions:

1. List of Faculty Members: Users can get a complete list of faculty members.

2. View Faculty Detail: After getting a faculty list, users can view details for a specific faculty by clicking on the detail button.

FACULTY has 1-M relationship with PROJECT.
Table 4. Pseudo Code to Search Account Information

<table>
<thead>
<tr>
<th>Pseudo Code to Search Account Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get parameters for select query from request object</td>
</tr>
<tr>
<td>Construct SQL query statement</td>
</tr>
<tr>
<td>If execution of the SQL statement fails then</td>
</tr>
<tr>
<td>send an alert and return</td>
</tr>
<tr>
<td>else then</td>
</tr>
<tr>
<td>select requested record(s) from STUDENT/FACULTY/STAFF table</td>
</tr>
<tr>
<td>Endif</td>
</tr>
<tr>
<td>Go back to account database page</td>
</tr>
</tbody>
</table>

3.3.2.4 Project Maintenance. Project Maintenance assists a faculty to create, monitor, change status or delete projects of their advisees.

A student may update his/her project information.

Functions. Project Maintenance consists of the following functions:

1. List of Projects: Generate a list of project(s) currently supervised by a certain faculty.
2. Edit Project: Edit project title, begin/finish year/term and progress status.
3. Create/Delete Projects: Create and delete student projects.

4. Update Project Abstract: Update the project abstracts.

FACULTY has 1-M relationship with PROJECT.

Figure 7. UML Model for Project Maintenance
Table 5. Pseudo Code to Maintain Project Information

Pseudo Code to Create/Insert Project Information

<table>
<thead>
<tr>
<th>Get parameters for insert query from request object</th>
</tr>
</thead>
<tbody>
<tr>
<td>If any parameter value is blank then</td>
</tr>
<tr>
<td>send an alert message and go back to input page</td>
</tr>
<tr>
<td>Endif</td>
</tr>
<tr>
<td>Construct SQL query statement</td>
</tr>
<tr>
<td>If execution of the SQL statement fails then</td>
</tr>
<tr>
<td>send an alert and return</td>
</tr>
<tr>
<td>else then</td>
</tr>
<tr>
<td>insert a new record to PROJECT table</td>
</tr>
<tr>
<td>Endif</td>
</tr>
<tr>
<td>Go back to project list page</td>
</tr>
</tbody>
</table>

Pseudo Code to Edit Project Information

<table>
<thead>
<tr>
<th>Get parameters for update query from request object</th>
</tr>
</thead>
<tbody>
<tr>
<td>If any parameter value is blank then</td>
</tr>
<tr>
<td>send an alert message and go back to edit page</td>
</tr>
<tr>
<td>Endif</td>
</tr>
<tr>
<td>Construct SQL query statement</td>
</tr>
<tr>
<td>If execution of the SQL statement fails then</td>
</tr>
<tr>
<td>send an alert and return</td>
</tr>
<tr>
<td>else then</td>
</tr>
<tr>
<td>update the record to PROJECT table</td>
</tr>
<tr>
<td>Endif</td>
</tr>
<tr>
<td>Go back to project list page</td>
</tr>
</tbody>
</table>
Pseudo Code to Delete Project Information

| Get parameters for delete query from request object |
| Construct SQL query statement |
| If execution of the SQL statement fails then |
| send an alert and return |
| else then |
| delete the record to PROJECT table |
| Endif |
| Go back to project list page |

3.3.2.5 Update/Download Project Information.

Update/Download Project Information assists students to
- update their project documentation in PDF, DOC, RTF, TEX, TXT, EPS and PS formats;
- upload/update abstract, proposal and final report in PDF, DOC, RTF, TEX, TXT, EPS and PS formats; and
- delete previously uploaded project documentation.
Figure 8. UML Model for Update/Download Project Information

Functions. Update/Download/Delete Project Information consists of the following functions:


2. Delete Project File: Delete a previously uploaded proposal, abstract and final report.
Table 6. Pseudo Code to Maintain Project Document Files

<table>
<thead>
<tr>
<th>Pseudo Code to Upload Project Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get parameters for UploadBean query from request object</td>
</tr>
<tr>
<td>If any parameter value is blank then</td>
</tr>
<tr>
<td>send an alert message and go back to input page</td>
</tr>
<tr>
<td>Endif</td>
</tr>
<tr>
<td>If execution of the UploadBean fails then</td>
</tr>
<tr>
<td>send an alert and return</td>
</tr>
<tr>
<td>else then</td>
</tr>
<tr>
<td>upload/update the file</td>
</tr>
<tr>
<td>Endif</td>
</tr>
<tr>
<td>Go back to student project page</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pseudo Code to Delete Project Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get parameters for delete query from request object</td>
</tr>
<tr>
<td>If execution of the file deletion fails then</td>
</tr>
<tr>
<td>send an alert and return</td>
</tr>
<tr>
<td>else then</td>
</tr>
<tr>
<td>delete the file</td>
</tr>
<tr>
<td>Endif</td>
</tr>
<tr>
<td>Go back to student project page</td>
</tr>
</tbody>
</table>
3.3.3 Security

The transmission between user-end and the web server, and the authentication of different users are the security issues being enforced in this project. The following measures are used to ensure security of sensitive information.

3.3.3.1 Encryption of Internet Transmission with SSL.
All JSP web pages are processed by the Apache Tomcat server, which is powered by the Apache Server and Open SSL. Communication between server and client sides are forced to transmit under SSL transmission protocol. Any attempts to access unsecured pages are redirected to secure pages.

3.3.3.2 User/Role and Password Enforced Authentication.
User authentication and authorization are implemented by user ID / password. When a user logs in, the system will verify the user ID and password against account tables. After a user is authenticated, the system will authorize the user to access to all functions appropriate for his or her role.

When a user attempts to execute a page, a verifying component will check if the user's privilege container has the privilege token required for the page. If the user does not have the proper privilege token, the verifying logic will deny access and redirect the request to a login page.
CHAPTER FOUR
DEPLOYMENT

4.1 System Requirements

The MySQL database server is located in a private network. The front end is the Apache Tomcat server with SSL. Window Advanced Server 2000 or Linux 9 can be used. However, Linux is recommended for its secure and performance features. Following is a recommendation for system deployment:

1. Firewall
2. Apache Tomcat server / SSL, minimum 64MB
3. MySQL database 4.1.14, minimum 32MB

Figure 9. Recommended Deployment
4.2 Installation

4.2.1 Installation of Servers

Apache Tomcat is installed as root and a group of user accounts are created for Tomcat to run. Apache server is configured to have an automatic startup when the server boots up, and shutdown properly before the server is powered off to avoid lost data. Detailed configuration for security is discussed in the Security Section.

4.2.2 Installation of Web Component

The project components are packaged into gpds.zip. Unzip the file in the /usr/local/jakarta-tomcat/webapps/ROOT/gpds folder. /usr/local/jakarta-tomcat is the folder where Apache Tomcat server is installed.

4.2.3 Setup Initiation Tables

There is a file named create_table.sql in the gpds folder. It is the file to create the initialization tables and the initialization administration account. Load the text file into MySQL then the initialization tables and the administration account will be created.
CHAPTER FIVE
CONCLUSION AND FUTURE DIRECTIONS

5.1 Technology Highlights and Conclusion

The Computer Science Graduate Project Management System implements data structure, SQL language, HTML, Java Server Pages, Java Bean, and relational database model to complete the whole application. Furthermore, several mechanisms like SSL and MD5 encryption are implemented to secure access from the Internet to the database system.

Using the relational database model with MySQL database, the information is modeled and constructed from basic and simple tables using binary and ternary relationships. As a result, the database system can cope with new requirements without drastic changes in programming.

Several security mechanisms have been implemented such as transmission encryption with SSL, password-enforced authentication and authorization, and encryption of SQL procedures.

The Computer Science Graduate Project Management System has been designed and developed for use in the Computer Science Department, and may be used by other departments in the university.
5.2 Extensions

Due to the scope of my master's project, the following feasible features are not yet implemented but are worthwhile to be added into the web-based database system. To implement these features, some additional tables need to be added into the database schema.

Progress Monitor: The system allows faculty to be able to track all progress of all projects. Faculty may monitor not only the documentation but also the progress of the programs. For example, a link to the project's webpage may be added.

Presentation Scheduler: When a student has finished his/her project, the system may be able assist to schedule a time/place for project presentation.

Discussion Board/Forum: The system will provide a forum for users to discuss project-related topics. Students may discuss project related problems on this discussion board.

File Download Interface: The interface for users to download project files may be improved to show the availability of each file type. Users can know which file is available for download before trying to download.
APPENDIX
Following are over 65 JSP, HTML, Java Script and files developed for the Database Management System. Naming convention is based on their functionality. They are arranged within their functions and in alphabetic order.

1. **Authentication & Authorization**
   - login.jsp
   - logout.jsp
   - sql8.jsp

2. **Completed Projects Search**
   - cppj_details.jsp
   - finished_pj.jsp

3. **Faculty Information**
   - ft_details_p.jsp
   - list_faculty_info.jsp

4. **Student Home Page**
   - student.jsp

5. **Student Project Page**
   - pj_details.jsp
   - sd_mpj.jsp

6. **Student Update Project Abstract**
   - updateabstractc.jsp
   - updateabstractd.jsp

7. **Student Upload Project File**
uploadrename3.jsp

8. **Student Delete Project File**
   deletefile.jsp
   deletefilec.jsp

9. **Student Edit Personal Information**
   sd_my_edit.jsp
   sd_my_editc.jsp
   sd_my_editd.jsp

10. **Faculty Home Page**
    faculty.jsp

11. **Faculty View Directing Projects**
    ft_vpj.jsp

12. **Faculty Create New Project**
    ft_pjcrt.jsp
    ft_pjcrtc.jsp
    ft_pjcrtcd.jsp

13. **Faculty View Project Detail**
    ft_pjdl.jsp

14. **Faculty Edit Project**
    ft_pjedit.jsp
    ft_pjeditc.jsp
    ft_pjeditd.jsp

15. **Faculty Delete Project**
    ft_pjdelc.jsp
16. Faculty View Personal Information
   ft_ifdtl.jsp
   ft_mif.jsp

17. Faculty Edit Personal Information
   ft_ifedit.jsp
   ft_ifeditc.jsp
   ft_ifeditd.jsp

18. Staff Home Page
   admin.jsp

19. Faculty Account List
   faculty_db.jsp

20. Create Faculty Account
   add_a_ft.jsp
   fa_db_add.jsp
   fa_db_add_con.jsp

21. Faculty Account Detail
   ft_details.jsp

22. Edit Faculty Account
   edit_faculty.jsp
   fa_db_edit.jsp
   fa_db_edit_con.jsp

23. Delete Faculty Account
   del_faculty.jsp
del_faculty_con.jsp

24. Student Account List
    student_db.jsp

25. Create Student Account
    sd_create.jsp
    sd_createc.jsp
    sd_created.jsp

26. Student Account Detail
    sd_detail.jsp

27. Edit Student Account
    sd_edit.jsp
    sd_editc.jsp
    sd_editd.jsp

28. Delete Student Account
    sd_delc.jsp
    sd_deld.jsp

29. Staff Account List
    staff_db.jsp

30. Create Staff Account
    add_a_sf.jsp
    sf_db_add.jsp
    sf_db_add_con.jsp

31. Staff Account Detail
    staff_details.jsp
32. Edit Staff Account
   edit_staff.jsp
   sf_db_edit.jsp
   sf_db_edit_con.jsp

33. Delete Staff Account
   del_staff.jsp
   del_staff_con.jsp
BIBLIOGRAPHY


http://www.w3.org/TR/REC-html40/interact/scripts.html

http://tomcat.apache.org/tomcat-5.5-doc/index.html