2006

Improvement and deployment of the web-based database management system for computer science graduate program

Li-Chuan Fang

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

Part of the Databases and Information Systems Commons

Recommended Citation
Fang, Li-Chuan, "Improvement and deployment of the web-based database management system for computer science graduate program" (2006). Theses Digitization Project. 2988.
https://scholarworks.lib.csusb.edu/etd-project/2988

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.
IMPROVEMENT AND DEPLOYMENT OF THE WEB-BASED
DATABASE MANAGEMENT SYSTEM FOR COMPUTER
SCIENCE GRADUATE PROGRAM

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
Li-Chuan Fang
June 2006
IMPROVEMENT AND DEPLOYMENT OF THE WEB-BASED DATABASE MANAGEMENT SYSTEM FOR COMPUTER SCIENCE GRADUATE PROGRAM

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

by Li-Chuan Fang
June 2006

Approved by:

Dr. Josephine G. Mendoza, Chair, Computer Science

Dr. Arturo I. Concepcion

Dr. Tong Lai Yu

Date 6-6-06
ABSTRACT

This project attempts to address the active and ongoing need for both faculty and student access to graduate student resources; moreover, it addresses the need for an integrated system which possess the capability for additions and utilitarian changes without excessively extensive revisions or modifications. There exists a need for vastly improved database views and report generating capacity. Therefore, the current project has required extensive computer design technique applications aimed toward actual delivery and implementation of the needed improvements.

In addition to the need for enhanced and expanded student and faculty access to relevant data, the improvements must include capability for prompt and accurate information updates; rapid access for advisor; increased scalability, or the ability to adequately serve multiple system users simultaneously; reduced need for manual input, thereby, reducing occurrence of errors; improved security protection; and capability for online advising as provided by the Graduate Coordinator. Overall, the scope and approach must be comprehensive, accurate, adaptable to intended program users, and practical in nature. Thus, extensive analysis and design planning were
necessary from the outset, encompassing a wide range of applications and specifications from actual use considerations to final reports. Perhaps, more than any other single consideration, the new system must be deployed and ready for use.
ACKNOWLEDGMENTS

First, I wish to thank my project advisor, Dr. Josephine Mendoza, for her consistent, strong and persistent support. With Dr. Mendoza’s guidance, I was able to completely and comprehensively analyze, design, and implement the critical phases of this project.

Moreover, I would also like to affirm my indebtedness to Dr. Arturo Concepcion, my project committee member, who assisted me throughout the crucial elements of the system design. I am grateful to Dr. Tong Lai Yu for his expertise and general suggestions.

My appreciation also extends to Mr. Dung Vu and Sunny Lin for their intensive technical support. I am most indebted to Ms. Tiffany Chiang for her consistent expression of ideas, thoughts, and opinions relative to the project.

Lastly, I am extremely grateful for the unrelenting support of several family members, including whole-hearted support from my husband, and particularly my mother and younger brother. Without encouragement from them, I almost certainly would not have progressed as rapidly as I have toward final completion of this project.
TABLE OF CONTENTS

ABSTRACT ................................................................. iii
ACKNOWLEDGMENTS ......................................................... v
LIST OF TABLES ........................................................... ix
LIST OF FIGURES .......................................................... x

CHAPTER ONE: INTRODUCTION
1.1 Background ............................................................. 1
1.2 Purpose of the Project ............................................... 3
1.3 Significance of the Project .......................................... 4

CHAPTER TWO: SOFTWARE REQUIREMENTS SPECIFICATION
2.1 Overall Description .................................................. 7
  2.1.1 Scope ............................................................... 7
  2.1.2 Definitions, Acronyms, and Abbreviations .................... 9
  2.1.3 Product Perspective ............................................. 11
  2.1.4 System Interfaces ............................................... 13
  2.1.5 User Interfaces .................................................. 13
  2.1.6 Database Interfaces ............................................. 15
2.2 Users ........................................................................ 15
2.3 Functions ................................................................. 17
  2.3.1 New Product Functions ......................................... 17
  2.3.2 Modifications to Existing Functions / Tables ............... 18
2.4 Constraints ............................................................... 26
2.5 System Characteristics ............................................... 26
  2.5.1 Reliability .......................................................... 26
2.5.2 Security .............................................. 27
2.5.3 Maintainability ................................. 27
2.5.4 Portability ........................................... 27

CHAPTER THREE: SOFTWARE DESIGN

3.1 Architectural Design .................................. 28
  3.1.1 System Design ................................. 28
  3.1.2 Database Design ................................. 28

CHAPTER FOUR: TEST AND MAINTENANCE

4.1 Unit Testing ........................................... 33
4.2 Test Scenarios ........................................ 35
  4.2.1 Main User Login ............................... 36
  4.2.2 Faculty / Graduate Coordinator / Program Assistant / Student: Change Password ...................... 38
  4.2.3 Faculty / Graduate Coordinator: Communications .............................................. 39
  4.2.4 Faculty / Graduate Coordinator / Program Assistant / Student: Change Home Page ...................... 47
  4.2.5 Faculty / Graduate Coordinator / Program Assistant: View Any Student Info ............................ 48
  4.2.6 Faculty / Graduate Coordinator: View Own Advisees Only ........................................ 51
  4.2.7 Graduate Coordinator: Academic Year Statistics ......................................................... 52
  4.2.8 Graduate Coordinator: Generate Reports - Pre-Defined Reports .............................. 52
  4.2.9 Graduate Coordinator: Generate Reports - Ad Hoc Reports ....................................... 56
CHAPTER FIVE: CONCLUSIONS

5.1 Project Accomplishments ...................... 81
5.2 Future Directions ............................... 83

5.2.1 Centralized Management of Documents .......... 83
5.2.2 Appointment Maker ........................... 83
5.2.3 Announcement ................................. 84
5.2.4 Frames versus Table Format ................... 84

APPENDIX A: DATA DICTIONARY ......................... 85
APPENDIX B: APPLICATION PROGRAM FILES ................ 109
REFERENCES ........................................ 119
LIST OF TABLES

Table 1. Comparisons between the Old System and New System ...................................... 18

Table 2. Yen-Hsi Chiang’s Table Structures - Revised Tables ........................................... 19

Table 3. Yen-Hsi Chiang’s Table Structures - New Data Tables ........................................ 22

Table 4. Yen-Hsi Chiang’s Table Structures - New Lookup Tables ...................................... 23

Table 5. Yen-Hsi Chiang’s Table Structures - New Exception Handling and Log Tables .......... 25

Table 6. Revision of Database Tables ............................................................... 29

Table 7. Newly Created Database Tables ............................................................. 30

Table 8. Unit Tests ..................................................................................... 33
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Use Case Diagram</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>Three-Tier Architecture Diagram</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>User Login</td>
<td>37</td>
</tr>
<tr>
<td>4</td>
<td>Change Password</td>
<td>38</td>
</tr>
<tr>
<td>5</td>
<td>Communications - All Advisees</td>
<td>39</td>
</tr>
<tr>
<td>6</td>
<td>Communications - Select Advisees</td>
<td>40</td>
</tr>
<tr>
<td>7</td>
<td>Communications - All Committees</td>
<td>41</td>
</tr>
<tr>
<td>8</td>
<td>Communications - Select Committees</td>
<td>42</td>
</tr>
<tr>
<td>9</td>
<td>Communications - Graduate Coordinator</td>
<td>43</td>
</tr>
<tr>
<td>10</td>
<td>Communications - Department of Chair</td>
<td>44</td>
</tr>
<tr>
<td>11</td>
<td>Communications - Dean of Graduate Studies</td>
<td>45</td>
</tr>
<tr>
<td>12</td>
<td>Communications - Program Assistant</td>
<td>46</td>
</tr>
<tr>
<td>13</td>
<td>Change Home Page</td>
<td>47</td>
</tr>
<tr>
<td>14</td>
<td>View Any Student Information</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>(Part A)</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>View Any Student Information</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>(Part B)</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>View Own Advisees Only</td>
<td>51</td>
</tr>
<tr>
<td>17</td>
<td>Academic Year Statistics</td>
<td>52</td>
</tr>
<tr>
<td>18</td>
<td>Generate Reports - Pre-Defined</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>Reports (Part A)</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Generate Reports - Pre-Defined</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>Reports (Part B)</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Generate Reports - Pre-Defined</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>Reports (Part C)</td>
<td></td>
</tr>
</tbody>
</table>
Figure 21. Generate Reports – Pre-Defined Reports (Part D) ........................................ 56
Figure 22. Generate Reports – Ad Hoc Reports (Input) ..................................................... 57
Figure 23. Generate Reports – Ad Hoc Reports (Output) .................................................... 58
Figure 24. Quarterly Reports .......................................................... 59
Figure 25. View Any Advisee ............................................................ 60
Figure 26. Update Student Information (Part A) ................. 61
Figure 27. Update Student Information (Part B) ................. 62
Figure 28. Update Student Information (Part C) ................. 63
Figure 29. Comprehensive Exam Information ................. 64
Figure 30. Insert New Faculty ........................................................ 66
Figure 31. Insert New Staff ............................................................ 67
Figure 32. Print Letters ............................................................... 68
Figure 33. Update/View Staff Information ..................................................... 69
Figure 34. Communications – All Committee Members .................................................... 70
Figure 35. Communications – Select Committee Members ..................................................... 71
Figure 36. Communications – Advisor Only ....................... 72
Figure 37. Communications – Graduate Coordinator ..................................................... 73
Figure 38. Communications – Department Chair ....................... 74
Figure 39. Communications – Dean of Graduate Studies ..................................................... 75
Figure 40. Communications – Program Assistant ....................... 76
Figure 41. Communications - Select Computer Science Faculty ......................... 77
Figure 42. Update Contact Information ...................... 78
Figure 43. View Student Information (Part A) ............ 79
Figure 44. View Student Information (Part B) .......... 80
CHAPTER ONE
INTRODUCTION

1.1 Background

Presently, the Department of Computer Science at California State University, San Bernardino utilizes a stand-alone system implementing Microsoft Access 2000 to manage the graduate student database. This has assisted the work of the Graduate Coordinator (GC) and the Program Assistant (PA). The system does feature an array of positive and highly utilitarian features. The system can generate both pre-defined reports and ad hoc reports, insert new student personal information, update existing student records, generate academic statistics, and provide weekly database backup. The reports display attendance status, academic standing as well as completed master projects and theses. The system also generates list of advanced to candidacy students who must be in continuous enrollment. The system is also able to generate GPA for all Computer Science courses taken by the students as well as prerequisites required for the students.

However, the current system suffers from certain limitations which need to be addressed. Significantly, the current system does not allow access by students nor by
faculty. Graduate students can not update contact information such as current address, telephone number, or e-mail address. Currently, this is done by filling out a hard copy -Personal Information Update Sheet. Also, advisors can not retrieve the information on their advisees. Moreover, the system lacks scalability. With the current stand-alone system, only one person can use the system at a time.

These limitations were addressed by Mr. Dung Vu in his Master Project - “The Web-Based Database Management System for the Computer Science Graduate Program.” Mr. Vu’s system used Oracle 9i, designed to facilitate web access to the database information by Computer Science Department faculty and graduate students. The improvements were demonstrated as a prototype and not deployed. Vu’s system added useful modules such as dynamic roles and functions; dynamic grouping; searching and task assignment. Students can update their personal information and view their status in the program, courses taken and grades earned. Although Vu’s system is a great improvement from the MS Access stand-alone system, it needs further improvement. Vu’s system still requires manual input of student grades by the PA. This deficiency was addressed by Ms. Yen-His Chiang in her Master Project, “Advising
Module: Graduate Application System for the Computer Science Graduate Program. Ms. Chiang provided a module that interfaced Vu’s system to SIS+ (CSUSB Student Information System). The courses taken and grades earned by students are extracted automatically from SIS+ and entered into the appropriate database tables.

1.2 Purpose of the Project

In Winter 2005 as part of my internship, I analyzed the deficiencies in the web-based system developed by Mr. Dung Vu. The purpose of this project is threefold (1) to remove the deficiencies, (2) to add modules to reflect the changes in the policies of the graduate program in Computer Science, and (3) to deploy the system to be used by the department.

For students admitted since Fall 2003 and opting to do the Master Project option will have to take and pass the oral exam first before being advanced to candidacy and allowed to register for CSCI 690 - Master Project. In Fall 2004, the University replaced social security numbers with student ids in identifying student records. Dung Vu’s system used social security numbers. My system will use the university-assigned student identification number.
Also in Fall 2004, the program added a new option - Comprehensive Exam.

The continuous enrollment module in the MS Access Database was not implemented in Vu’s system. This project will implement this module. The login module is enhanced by two features: (1) encrypts the passwords to maintain security, and (2) forces the users to change initial passwords at first login. This project uses JSP, Oracle Database Server, and Web-Application Server to provide

- views of the database by the coordinator, faculty and students;
- improved communication access for students, such as emails to professors, committee, coordinator and program assistant;
- enhanced report capability that includes ad hoc and quarterly reports in addition to standard reports.

1.3 Significance of the Project

The proposed improved system will be fully deployed for use by the CSCI Graduate Program.

The Graduate Coordinator will directly advise students online. This will help eliminate the necessity of retrieving student data from numerous printed pages.
Moreover, updating data is difficult when using only a paper format. Computerized data facilitates prompt updates, keeping information current, and maximizing efficiency and accuracy.

Students can update their personal information in connection with current addresses and email addresses. They can also view their academic information, learning about current status, and be informed about the next steps to follow within their academic program. Moreover, if there is any incorrect information, the student can readily notify the Graduate Coordinator about needed corrections.

For the Program Assistant, there will no longer be a need for typing in student personal or academic data. This serves to reduce inevitable typographical errors. Further, the new system permits prompt, accurate report generation, including use of ad hoc reports. Also, the system will provide for alert emails to students, so as to promptly advise students whose grade point averages have fallen below the required 3.0 GPA. The system will also alert students about any courses where they earned at least a C-grade. These represent two examples of alert emails provided, though there exists several others as well.
Faculty advisors will also benefit from the new system. They can quickly view their advisees’ academic records, including critical test scores, and these data may assist them in modifying instructional methods as needed without accessing SIS+.

The system can be presented to the CSUSB Graduate Council as a prototype for CSUSB Graduate Coordinator’s Database Management System for possible adoption.
2.1 Overall Description

2.1.1 Scope

During Winter 2005, I incorporated policy changes to the existing MS Access stand-alone system now in use by the Graduate Coordinator, and comprehensively examined the Web-Based Database Management System for the Computer Science Graduate Program. The intent was to implement changes useful to the coordinator, program assistant, students and faculty.

This project will provide the changes to correct deficiencies and develop practical and new application modules.

The changes are:

- Use the SID (Student Identification Number) rather than the Social Security Number as a student identifier.
- Revise the existing database table structure to accommodate the revised and new database application requirements.
• Implement the report-generation capability which was available in the MS access system but not implemented in Vu’s system.

The new modules are:

• Comprehensive Exam Module: Starting in Fall 2004, the comprehensive exam option is the third option available to CSCI graduate students.

• Oral Exam Before Advancement to Candidacy Module: Starting in Fall 2003, students choosing the project option will need to take and pass the oral exam before being advanced to candidacy and register for CS690, the Master’s Project. This module will provide guide questions and procedure that the students need to follow.

• Continuous Enrollment (CSCI698) Module: Students who have registered for the maximum number of units required for Project (5 units) or Thesis (9 units) or finished all courses and waiting to take Comprehensive Exam (1 unit) need to be in Continuous Enrollment. This module will
produce at the beginning of every quarter a listing of students who need to be in Continuous Enrollment.

2.1.2 Definitions, Acronyms, and Abbreviations

The following terms are defined as they apply to the project.

**Browser** - a program that accesses and displays files and other data available on the Internet and other network.

**Graduate Coordinator for Advising (GCA)** - a faculty member assigned to advise graduate students who have been accepted to the Master Program in Computer Science.

**Graphical User Interface (GUI)** - an interface for issuing commands to a computer utilizing a pointing device, (such as a mouse), that manipulates and activates graphical images on a monitor.

**Hypertext Markup Language (HTML)** - a markup language used to structure text and multimedia documents, to set up hypertext links between documents, and used extensively on the World Wide Web.

**IEEE** - Institute of Electrical and Electronics Engineers.

**JSP (JavaServer Pages)** - a technology that enables Web developers and designers to rapidly develop and
easily maintain, information-rich, dynamic Web pages that leverage existing business systems.

**Module** - a portion of a program that carries out a specific function and may be used alone or combined with other modules of the same program.

**NIC** - (Network Interface Card) an adapter circuit board installed in a computer to provide a physical connection to a network.

**OC4J** - the core J2EE runtime component of Oracle Application Server.

**Oracle** - a relational database management system developed by Oracle Corp.

**Program Assistant (PA)** - a staff member in the Computer Science Department who assists the GCA.

**SIS+** - Student Information System used by California State University, San Bernardino.

**SRS** (Software Requirement Specification) - a complete description of the behavior of the system to be developed.

**Status** - the current academic standing of a student who is enrolled in the university - Probation, Conditionally Classified, Classified, Preliminary Advanced to Candidacy, and Advanced to Candidacy.
XML (Extensible Markup Language) – a pared-down version of SGML (Standard Generalized Markup Language), designed especially for Web documents. It allows designers to create their own customized tags, enabling the definition, transmission, validation, and interpretation of data between applications and between organizations.

2.1.3 Product Perspective

Students can login, view personal information, update contact information, customize home page, and change password. Faculty can login, view own advisees, customize home page, and change password. In addition to the functions allowed for the faculty, the Graduate Coordinator can view not only own advisees but also any student or advisee of any faculty member. The Program Assistant can login, customize home page, change password, and update student information. The System Administrator can login, change password, and customize home page.
Figure 1. Use Case Diagram
2.1.4 System Interfaces

The Oracle Application Server with HTTP and Java Container J2EE (OC4J) at the server side interacts with a user at the client side.

![Three-Tier Architecture Diagram](image)

Figure 2. Three-Tier Architecture Diagram

2.1.5 User Interfaces

All user interfaces will be dynamic. They will allow the users to retrieve data from the database for updating or viewing. There are five roles allowed to log into this system – Graduate Coordinator, Faculty, Student, and Program Assistant, and System Administrator.

All users may use any web browser to interface with the client side. There is no restriction on the operating system used on the client side.
2.1.5.1 Faculty Component. This component will provide a faculty member the following functions: change password; communicate with all advisees, selected advisees, all committees, selected committees, graduate coordinator, department chair, dean of Graduate Studies, and program assistant; customize home page; view any student info; and view own advisees only.

2.1.5.2 Graduate Coordinator for Advising Component. This component will provide the Graduate Coordinator for Advising the following capabilities: view the academic year statistics; access the advising module; change password; communicate with all advisees, selected advisees, all committees, selected committees, Graduate Coordinator, Department Chair, Dean of Graduate Studies, and Program Assistant; customize home page; generate reports: pre-defined reports (reports on attendance status and reports on academic standing), ad hoc reports, and quarterly reports (courses taken with at least C-grade, student GPA below 3.0, students off probation, preliminary advanced to candidacy, less than maximum units for CSCI 690, less than maximum units for CSCI 699 and continuous enrollment); update student info; view any advisees; view any student info; view own advisees only
(since GCA is also a Faculty Advisor and a faculty member).

2.1.5.3 Program Assistant Component. This component will provide the Program Assistant the following tasks: change password; insert or update comprehensive exam information; customize home page; insert new faculty; insert new staff; insert new student; print letters; update/view student information; update/view staff information; and view any advisee.

2.1.6 Database Interfaces

This application will have the interfaces required to access the Oracle database server through JSP requests made from the web server. The server implementation will use SQL queries embedded in JSP to access the information from the Oracle database server.

2.2 Users

Users are classified as application users and system administrator. Functions common to all users include ability to login, change password customize home page and log out. However, each user will have specific functions as described below.
1. **Application Users** - include the graduate student, program assistant (PA), faculty member, and Graduate Coordinator for Advising (GCA).
   a. *Faculty Member*: views personal and academic information of any student as well as view own advisees.
   b. *Graduate Coordinator*: accesses the advising module, views academic year statistics, generates pre-defined reports, ad hoc reports, and quarterly reports, updates/views any student information, and views her/other advisees.
   c. *Program Assistant*: inserts/updates faculty or staff, updates/views any student information, views any advisees, and prints letters which are automatically emailed to the students by the system.
   d. *Graduate Student*: updates his or her contact information, and views own personal and academic information.

2. **System Administrator** - installs and manages the Oracle 9i Database and the Oracle 9i Application Server. In addition, this person must be knowledgeable in Linux installation and have the
hands-on experience in troubleshooting and database performance tuning, backup and recovery.

2.3 Functions

New product functions include comprehensive exam, oral exam before advancement to candidacy, and continuous enrollment. These functions were added because they were not implemented in Vu's system.

Integration of data with SIS+ has required changes in existing tables. Details of the changes done will be provided in Section 2.3.2.

2.3.1 New Product Functions

The new product functions include:

1. Comprehensive Exam -- allows the system to implement the third master option.

2. Oral Exam Before Advancement to Candidacy -- allows the system to handle a new policy. Students admitted since Fall 2003 and choosing the project option must take and pass the oral examination before being advanced to candidacy and allowed to register for the Project course, CSCI 690. Students are allowed to retake the oral exam, if they failed the first oral exam.
3. Continuous Enrollment (CSCI698) -- allows the system to track registration in CSCI 698 of graduate students who have already enrolled for the maximum units required for a project or thesis course but have not completed the project or thesis, have not done a project presentation or thesis defense, and have not passed the project or thesis document format check.

2.3.2 Modifications to Existing Functions / Tables

The integration of data with SIS+ required changes in the existing database tables as follows:

Table 1. Comparisons between the Old System and New System

<table>
<thead>
<tr>
<th></th>
<th>Old System</th>
<th>New System</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Field terms coded as Quarter Year pairs, i.e. CC#### in which CC stands for first two letters of the quarter (Fa - Fall, Wi - Winter, Sp - Spring, Su - Summer), and #### represents the year</td>
<td>3-digit SIS+ coding, yyT: yy represents last 2 digits of year and T stands for the quarter number (1 - Winter, 2 - Spring, 3 - Summer, 4 - Fall)</td>
<td></td>
</tr>
<tr>
<td>2. Two tables for Project and Thesis options only</td>
<td>Added third table for the Comprehensive Exam option</td>
<td></td>
</tr>
</tbody>
</table>
Table 2. Yen-Hsi Chiang’s Table Structures – Revised

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRADSTUDENTS</td>
<td>• Main table to store graduate student information</td>
<td>• Dropped columns: SSN, QTR_ADMIT, YR_ADMIT, QTR_CLASSIFIED, YR_CLASSIFIED,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CUR_QTR_START, CUR_YR_START, CUR_YR_END, QTR_LASTATD, YR_LASTATD, TOEFL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCORES, TOEFL_DATE, GRE_SUBJ, GRE_SUBJ_DATE, QTR_CANDIDACY, YR_CANDIDACY,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PRO_THESIS, TITLE, ORAL_EXAM, PASS, PRESENTATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Added columns: ADMIT_TERM, IS_CS, CLASSIFIED_TERM, CANDIDACY_TERM, CUR_START,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CUR_END, TERM_LASTATD, SIS_EMAIL, SIS_PHONE, NATIONALITY, EXIT_ACTION_ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Modified column names</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Old</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CSUSB BS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CA RES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ETHNICITY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PRO THESIS</td>
</tr>
<tr>
<td>STUDENTCOURSES</td>
<td>• Stores courses taken by a student</td>
<td>• Replace QUARTER_ID and YEAR columns with one new column COURSE_TERM VARCHAR2</td>
</tr>
<tr>
<td></td>
<td>• Each student has as many records as courses taken</td>
<td>(5) to represent when the course is taken</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Add a new column, DISCOUNT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GRADE CHAR (1), to indicate if this course will be counted in the GPA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>calculation</td>
</tr>
<tr>
<td>GRADES</td>
<td>• Defines letter grade and corresponding numerical grade and associated</td>
<td>• Replaces old system defined data values of GRADE ID (letter grade) and</td>
</tr>
<tr>
<td></td>
<td>description, e.g. letter grade “A” corresponds to numerical grade “4.0”.</td>
<td>SCORES (numerical grade) with SIS+ defined letter grade and numerical grade</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Adds a new column, GRADEID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DEFINITION VARCHAR2 (30), to store SIS+ Grade ID definition/description.</td>
</tr>
<tr>
<td>Table Name</td>
<td>Description</td>
<td>Modification</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ETHNICSORGN</td>
<td>- Defines ethnic origin of student</td>
<td>- Replaces old system defined data values of ETHNICITY_ID and ETHNICITY with SIS+ ETHNICITY code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Adds a new column, ETHNIC DEFINITION VARCHAR2 (250), to store SIS+ Ethnic code definition and description</td>
</tr>
<tr>
<td>COUNTRIES</td>
<td>- Defines country where international student comes from.</td>
<td>- Adds a new column, DEF_STANDING VARCHAR2 (500), to store long description or definition of country code/id.</td>
</tr>
<tr>
<td>COURSES</td>
<td>- Defines courses taught or counted in the graduate program including recommended, prerequisite, elective and core courses.</td>
<td>- Adds the new column, DESCRIPTION VARCHAR2 (1024), to store course description</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Adds the new column, GROUPID CHAR (1), to store course group ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Adds the new column, QTR_OFFER VARCHAR2 (4), to store course offer quarter (F for Fall, W for winter, S for Spring, and X for Summer)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Adds the new column, IS GRADE_STORE CHAR (1), to indicate whether it is a course whose grade will be stored in departmental database.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Adds the new column, IS_MS CHAR (1), to indicate whether it is a course which will be scheduled in the study plan.</td>
</tr>
<tr>
<td>QUARTER</td>
<td>- Defines quarters for courses taken.</td>
<td>- Change data definition for QUARTER_ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Winter</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Spring</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Summer</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Fall</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Table Name</td>
<td>Description</td>
<td>Modification</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ACADSTANDING</td>
<td>• Defines academic standing of students (probation, conditionally classified, classified, or advanced to candidacy)</td>
<td>• Adds a new column, DEF_STANDING VARCHAR2 (500), to store long description and definition of academic standing code.</td>
</tr>
<tr>
<td>ADMITSTATUS</td>
<td>• Defines admission status of students (probation conditionally classified, probation classified, conditionally classified, classified)</td>
<td>• Adds a new column, DEF_STANDING VARCHAR2 (500), to store long description or definition of admission status code.</td>
</tr>
<tr>
<td>CURRENTSTATUS</td>
<td>• Defines current status of students (Active, Incoming, Graduated, Inactive Attended, On Leave, Dismissed, Never Attended, Academic Probation, Withdraw)</td>
<td>• Adds a new column, DEF_CUR_STATUS VARCHAR2 (500), to store long description or definition of current status code.</td>
</tr>
</tbody>
</table>
Table 3. Yen-Hsi Chiang’s Table Structures — New Data

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIS_EXTRACT</td>
<td>Stores raw data extracted from SIS+</td>
</tr>
<tr>
<td>SIS_CLASS_ATTEND</td>
<td>Stores graduate student’s classified status (Conditionally Classified,</td>
</tr>
<tr>
<td></td>
<td>Classified, Advanced to Candidacy) and attending status (Attending,</td>
</tr>
<tr>
<td></td>
<td>Withdrawn, Dropped)</td>
</tr>
<tr>
<td>EXAM</td>
<td>Stores graduate student’s exam information, e.g. TOEFL, WREE.</td>
</tr>
<tr>
<td>COMEXAM</td>
<td>Stores graduate student’s master comprehensive exam information</td>
</tr>
<tr>
<td>PROJECT</td>
<td>Stores graduate student’s master project information</td>
</tr>
<tr>
<td>THESIS</td>
<td>Stores graduate student’s master thesis option information</td>
</tr>
</tbody>
</table>
Table 4. Yen-His Chiang's Table Structures - New Lookup Tables

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TERMS</td>
<td>Academic terms valid at CSUSB. Term is in the format “YYT” where “YY” represents the calendar year and “T” is the Term Suffix which represents the specific quarter term within the calendar year, Winter is 1, Spring is 2, Summer is 3, and Fall is 4</td>
</tr>
<tr>
<td>EXAMINFO</td>
<td>Information and rules regarding each specific type of exam, such as WREE, TOEFL</td>
</tr>
<tr>
<td>SIS_AA651_EXIT_ACTION</td>
<td>List of SIS+ raw data and associated definition regarding the action which brought about the termination of this matriculation, “GD” for “Graduation”, “IA” for “Leave of Absence”, “DD” for “Disenroll/Drop”</td>
</tr>
<tr>
<td>SIS_EXTRACT_LAYOUT</td>
<td>List of record layouts of SIS+ extract segments and data elements</td>
</tr>
<tr>
<td>SIS_RT047_ATTEND_STAT</td>
<td>List of SIS+ raw data and associated definition regarding student's attendance, “W” for “Withdrawn Regular”, “N” for “Not Attending”, and “1” for “Dropped Regular Retained”</td>
</tr>
<tr>
<td>SIS_RT080_ACAD_ACTION</td>
<td>List of SIS+ raw data and associated definition regarding the academic actions, “D” for “Disqualified”, “U” for “Probation”, “C” for “Continued on Probation”, and “M” for “Dismissal”</td>
</tr>
<tr>
<td>SIS_RT14C_CLASSIFICATION</td>
<td>List of SIS+ raw data and associated definition regarding to student’s classification, “51” for “Conditionally Classified”, “52” for “Classified”, and “53” for “Advanced to Candidacy”</td>
</tr>
<tr>
<td>SIS_RT310_GRADE_TYPE</td>
<td>List of SIS+ raw data and associated definition regarding the type of grading method, “CN” for “Credit/No Credit”, “CE” for “Credit by Exam”, “AU” for “Audit”, and “DC” for “Discount credit on repeats”</td>
</tr>
<tr>
<td>Table Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SIS_RT312GRADE_RULE</td>
<td>- List of SIS+ raw data and associated definition regarding the Grade Rule applicable, &quot;E&quot; for &quot;Extended Education&quot;, &quot;U&quot; for &quot;Undergraduate&quot;, and &quot;G&quot; for &quot;Graduate&quot;</td>
</tr>
<tr>
<td></td>
<td>- If a course section has a Grade Rule, this field will be initialized from the course section at the time of registration; otherwise the Grade Rule for the student academic program will be used</td>
</tr>
<tr>
<td>SIS_RT345CLASS_ENROLL</td>
<td>- List of SIS+ raw data and associated definition to reflect the student's status (itemize values) in the class section</td>
</tr>
<tr>
<td></td>
<td>- Active requests, attempts, permits, authorizations and drops are included</td>
</tr>
<tr>
<td></td>
<td>- Based on the date in the calendar table, a course dropped after the deadline for withdrawal grades will be changed to enrolled with a grade of W</td>
</tr>
</tbody>
</table>


Table 5. Yen-Hsi Chiang’s Table Structures - New Exception Handling and Log Tables

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRE_EXCP</td>
<td>An exception handling log table to track the failed transaction data on GRE exam</td>
</tr>
<tr>
<td>LOG_AUTOPROC</td>
<td>A log table that tracks execution of automation process. Records starting and ending time when each stored procedure was executed</td>
</tr>
<tr>
<td>LOG_CLASSCHANGE</td>
<td>A log table that tracks changes made on student’s classification status.</td>
</tr>
<tr>
<td>LOG_COURSEGRADE</td>
<td>A log table that tracks change made on student’s courses taken information of STUDENTCOURSES table</td>
</tr>
<tr>
<td>LOG_NEWSTD</td>
<td>A log table that tracks new student record inserted into GRADSTUDENTS table by the automation procedure</td>
</tr>
<tr>
<td>LOG_STATUSCHANGE</td>
<td>A log table that tracks any student’s status change in GRADSTUDENTS table by the automation procedure</td>
</tr>
<tr>
<td>LOG_GPA_Below3POINT0</td>
<td>A log table that stores those whose GPA is below 3.0.</td>
</tr>
<tr>
<td>NUKELOG_CTL</td>
<td>A lookup table which provides information for automation procedure to delete the old logs in assigned timely intervals</td>
</tr>
<tr>
<td>STDCOURSES_EXCP</td>
<td>An exception handling log table to track failed transactions on migrating student courses taken and grade</td>
</tr>
<tr>
<td>STDCOURSES_NOSTORE</td>
<td>A log table to track students’ non-stored (filtered out by procedure) courses and grade, such as PE class or non CSCI courses.</td>
</tr>
<tr>
<td>STDINFO_EXCP</td>
<td>An exception handling log table to track failed transactions on migrating graduate student’s bio/demo data.</td>
</tr>
<tr>
<td>STDREQ_EXCP</td>
<td>An exception handling log table to track failed transactions on inserting prerequisite requirements</td>
</tr>
<tr>
<td>Table Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TOEFL_EXCP</td>
<td>An exception handling log table to track failed transactions on migrating student's TOEFL exam</td>
</tr>
<tr>
<td>PROB_EXCP</td>
<td>An exception handling log table to track failed transactions on migrating student's probation information</td>
</tr>
<tr>
<td>COMMITTEE_EXCP</td>
<td>An exception handling log table to track failed transactions on migrating student's committee information</td>
</tr>
<tr>
<td>MASTEROPTION_EXCP</td>
<td>An exception handling log table to track failed transactions on migrating student's project or thesis information</td>
</tr>
<tr>
<td>USER_LOG</td>
<td>A log table that tracks application login activities, such as login time, IP, and the number of failed login attempts.</td>
</tr>
<tr>
<td>VISIT_LOG</td>
<td>A log table that tracks application usage by capturing visitor's activities, such as visited pages and visited time.</td>
</tr>
</tbody>
</table>

2.4 Constraints

This system is only as good as the data that has been migrated from MS Access Database and extracted from SIS+. This system depends on the correctness of the data from MS Access Database and SIS+.

2.5 System Characteristics

2.5.1 Reliability

The system should handle failure when the Oracle database is down. The system must provide sufficient connection to the database system to maintain high
availability. In this system, there are exception handlers to guarantee that the system is reliable.

2.5.2 Security

The system must be protected from any security risks. The Oracle Database as well as the Oracle Application Server must be protected. This system uses https protocol, which combines normal HTTP interaction over an encrypted secure socket layer. This ensures reasonable protection from eavesdroppers and man in the middle attacks.

2.5.3 Maintainability

The database and web application should be designed to deal with the changes and requirements without any significant modification. This application is implemented modularly so it is easy to maintain.

2.5.4 Portability

The system should be able to expand its service to different machines without any significant modification. Java and JSP are both portable to multiple operating systems and web servers. This system is portable since it is implemented in Java and JSP.
CHAPTER THREE
SOFTWARE DESIGN

3.1 Architectural Design

3.1.1 System Design

The Web-Based Database Management System uses two kinds of Oracle servers; one is the web application server and the other one is the database server. The Oracle Application Server, located behind a firewall, is the only computer that can connect directly to the database server. The Oracle Database Server is located in a private network and only allows the direct connection from the web server. This prevents hackers from cracking the database. For system deployment, the following are needed: (1) Firewall, (2) Oracle Application Server 10g, minimum 512 MB, and (3) Oracle Database Server 10g, minimum 512 MB.

3.1.2 Database Design

The new database design includes revision of several tables to accommodate changes in raw data sources and definitions as well as new database tables for storage of new data definitions. Moreover, specific new tables are created to collect activity information for automating processes in connection with data feed from SIS+.
Table 6. Revision of Database Tables

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROJECT</td>
<td>• Stores graduate student’s master project option information</td>
<td>• Add an additional new column, MODIFY_DONE_FLAG CHAR(1), to indicate whether the modification to the project document as recommended by the student’s committee has been done by the student</td>
</tr>
<tr>
<td>THESIS</td>
<td>• Stores graduate student’s master thesis option information</td>
<td>• Add an additional new column, MODIFY_DONE_FLAG CHAR(1), to indicate whether the modification to the thesis document as recommended by the student’s committee has been done by the student</td>
</tr>
</tbody>
</table>
Table 7. Newly Created Database Tables

<table>
<thead>
<tr>
<th>Table Name / Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADHOCLOG</td>
<td>Stores the SQL statements which the user creates in the ad hoc reports</td>
</tr>
<tr>
<td>USER_ID</td>
<td>Identification number to verify the authorized user</td>
</tr>
<tr>
<td>SQLNAME</td>
<td>Name of SQL statement which the user creates in the ad hoc reports</td>
</tr>
<tr>
<td>SQLSTRING</td>
<td>SQL statement which the user creates</td>
</tr>
<tr>
<td>TASKSCHEDULE</td>
<td>Defines frequencies for reported tasks</td>
</tr>
<tr>
<td>SCHEDULE_ID</td>
<td>Identification number of the schedule</td>
</tr>
<tr>
<td>SCHEDULE</td>
<td>Name of the schedule</td>
</tr>
<tr>
<td>SCHEDULE_DAYS</td>
<td>Duration of the schedule</td>
</tr>
<tr>
<td>ROLES</td>
<td>Defines roles of users (Graduate Coordinator, Faculty, Program Assistant, and Student)</td>
</tr>
<tr>
<td>ROLE_ID</td>
<td>Identification number of the role</td>
</tr>
<tr>
<td>ROLE</td>
<td>A set or group of privileges that can be granted to users</td>
</tr>
<tr>
<td>FUNCTIONS</td>
<td>Defines all registered functions to assign to roles and to users</td>
</tr>
<tr>
<td>FUNCTION_ID</td>
<td>Identification number of the function</td>
</tr>
<tr>
<td>FUNCTION</td>
<td>Name of the function</td>
</tr>
<tr>
<td>FUNCTIONPAGE</td>
<td>JSP page of the function</td>
</tr>
<tr>
<td>PRIVILEGE_CODE</td>
<td>Execution privilege token of the function</td>
</tr>
<tr>
<td>TASKSTATUS</td>
<td>Defines status of a task (denied, completed, failed, not yet evaluated, and uncompleted)</td>
</tr>
<tr>
<td>STATUS_ID</td>
<td>Identification number of the status</td>
</tr>
<tr>
<td>STATUS</td>
<td>Description of the status</td>
</tr>
<tr>
<td>ROLEFUNCTIONS</td>
<td>Stores functions assigned to each user role</td>
</tr>
<tr>
<td>ROLE_ID</td>
<td>Identification number of the role</td>
</tr>
<tr>
<td>FUNCTION_ID</td>
<td>Identification number of the function</td>
</tr>
<tr>
<td>TASKMANUAL</td>
<td>Defines all pre-defined tasks</td>
</tr>
<tr>
<td>TASK_CODE</td>
<td>Identification number of the task</td>
</tr>
<tr>
<td>TASK_ACTION</td>
<td>Brief description of the task</td>
</tr>
<tr>
<td>GUIDE</td>
<td>Guide to complete the task</td>
</tr>
<tr>
<td>Table Name / Column Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| JOBS JOB_ID JOB JOB_SCRIPT | • Defines pre-defined job for pre-defined tasks  
• Identification number of the job  
• Name of the job  
• Description of the job |
| TASKBOOK TASK_CODE JOB_ID JOB_ORDER | • Stores jobs in each pre-defined task  
• Identification number of the task  
• Identification number of the job  
• Sequence number of the job within the task |
| TASKREGSTR TASK_ID TASK_CODE TASK_OWNER TASK_ASSIGNEE DATE_ASSIGNED DATE_DUE DATE_COMPLETE DATE_EXPIRED NOTES STATUS REPORT COMMENTS | • Stores tasks assigned to each student, faculty, or staff member  
• Identification number of the task  
• Code of the task  
• Identification number of the staff who assigns the task  
• Identification number of the staff or student who needs to complete the task  
• Date when the task is assigned  
• Date when the task is expected to be completed  
• Date when the task is actually completed  
• Date when the task expires or is no longer valid  
• Brief note about the task  
• Status of the task - completed, failed, denied, uncompleted  
• Comment from the assignee about the task  
• Comment from the assignor about the task |
| COMPEXAMFACULTY QTRYR COURSE FACULTY COMPEXAM_ID | • Stores all information for the comprehensive exam  
• Quarter / year of the comprehensive exam  
• Name of the core course  
• Faculty name who made the exam questions  
• Identification number of the comprehensive exam |
| LOGINTRACE USER_ID ISEXPIRED CHANGEPWORDDATE LASTLOGIN | • Tracks the user login information  
• Identification number of the authorized user  
• Flag to track if the user logins for the first-time  
• Date when password is changed  
• Time of last-time login |
<table>
<thead>
<tr>
<th>Table Name / Column Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| JOB_CTL_TB, JOB_SEQ, RUN_DATE, PROC_NAME, COMMENTS | - Stores the system job information in the job scheduler  
- Identification number of the system job schedule  
- Date when the system job schedule is to be run  
- Name of the procedure for running the system job  
- Comments about the system job schedule |
| STUDENTEMAILLOG, STD_ID, SEND_DATE, ISSEND, COMMENTS, TERMCODE, REPORTCODE | - Tracks all information for emails sent to students  
- Identification number of the student  
- Date when email is sent to the student  
- Flag that indicates if the email has been sent  
- Content of the subject field of email  
- Quarter when the email is sent  
- Name of the report for which email is to be generated |
| DISMISS, STD_ID, DISMISS_TERM, DISMISS_REASON | - Tracks a student’s dismissal history  
- Identification number of the student  
- Quarter / year when the student is dismissed  
- Reason for dismissal |
| YEARS, YEAR | - Defines year range for an academic year  
- Starting year and ending year for the academic year |
| STUDENTMAILS, STD_ID, TERM, CATEGORY, CONTENT | - Stores the contents of the emails sent to students  
- Identification number of the student  
- Quarter / year when email is sent to the student  
- Type of the email  
- Body of the email |
CHAPTER FOUR

TEST AND MAINTENANCE

This chapter presents the quality assurance process.
There are two methods used for testing - unit tests and scenario tests.

4.1 Unit Testing

The unit testing provides a systematic approach for testing the user inputs and intended results of the system and improves the overall quality of the system. The following table shows the unit tests for the system.

Table 8. Unit Tests

<table>
<thead>
<tr>
<th>Page</th>
<th>Unit</th>
<th>Tests Performed</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login</td>
<td>User Input</td>
<td>• Test valid user</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Test invalid user</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Buttons/Links</td>
<td>• Ensure all buttons and links work as expected</td>
<td>Pass</td>
</tr>
<tr>
<td>Forgot Password</td>
<td>User Input</td>
<td>• Verify that all fields take expected input</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Test invalid data</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Buttons/Links</td>
<td>• Ensure all buttons and links work as expected</td>
<td>Pass</td>
</tr>
<tr>
<td>Change Password</td>
<td>User Input</td>
<td>• Verify that all fields take expected input</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Test invalid data</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Buttons/Links</td>
<td>• Ensure all buttons and links work as expected</td>
<td>Pass</td>
</tr>
<tr>
<td>Communications</td>
<td>Buttons/Links</td>
<td>• Ensure all buttons and links work as expected</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td>Menu</td>
<td>• Make sure all the menu items work appropriately</td>
<td>Pass</td>
</tr>
<tr>
<td>Customize Home Page</td>
<td>Buttons/Links</td>
<td>• Ensure all buttons and links work as expected</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td>Menu</td>
<td>• Make sure all the menu items work appropriately</td>
<td>Pass</td>
</tr>
<tr>
<td>Page</td>
<td>Unit</td>
<td>Tests Performed</td>
<td>Result</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>View Any Student Info</td>
<td>User Input</td>
<td>Verify that all fields take expected input</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Make sure all the menu items work appropriately</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ensure all buttons and links work as expected</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td>Menu</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Buttons/Links</td>
<td></td>
<td></td>
</tr>
<tr>
<td>View Own Advisees Only</td>
<td>Menu</td>
<td>Make sure all the menu items work appropriately</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td>Buttons/Links</td>
<td>Ensure all buttons and links work as expected</td>
<td>Pass</td>
</tr>
<tr>
<td>Academic Year Statistics</td>
<td>Menu</td>
<td>Make sure all the menu items work appropriately</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td>Buttons/Links</td>
<td>Ensure all buttons and links work as expected</td>
<td>Pass</td>
</tr>
<tr>
<td>Advising Link</td>
<td>Menu</td>
<td>Make sure all the menu items work appropriately</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td>Buttons/Links</td>
<td>Ensure all buttons and links work as expected</td>
<td>Pass</td>
</tr>
<tr>
<td>Generate Reports</td>
<td>Menu</td>
<td>Make sure all the menu items work appropriately</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td>Buttons/Links</td>
<td>Ensure all buttons and links work as expected</td>
<td>Pass</td>
</tr>
<tr>
<td>Quarterly Reports</td>
<td>Menu</td>
<td>Make sure all the menu items work appropriately</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td>Buttons/Links</td>
<td>Ensure all buttons and links work as expected</td>
<td>Pass</td>
</tr>
<tr>
<td>Update Student Info</td>
<td>User Input</td>
<td>Verify that all fields take expected input</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Make sure all the menu items work appropriately</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ensure all buttons and links work as expected</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td>Menu</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Buttons/Links</td>
<td></td>
<td></td>
</tr>
<tr>
<td>View Any Advisee</td>
<td>Menu</td>
<td>Make sure all the menu items work appropriately</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td>Buttons/Links</td>
<td>Ensure all buttons and links work as expected</td>
<td>Pass</td>
</tr>
<tr>
<td>Comp Exam Info</td>
<td>User Input</td>
<td>Verify that all fields take expected input</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Make sure all the menu items work appropriately</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ensure all buttons and links work as expected</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td>Menu</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Buttons/Links</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Insert New Faculty</td>
<td>User Input</td>
<td>Verify that all fields take expected input</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Make sure all the menu items work appropriately</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ensure all buttons and links work as expected</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td>Menu</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Buttons/Links</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Page</td>
<td>Unit</td>
<td>Tests Performed</td>
<td>Result</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------</td>
<td>--------------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>Insert New Staff</td>
<td>User Input</td>
<td>• Verify that all fields take expected input</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Test invalid data</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Menu</td>
<td>• Make sure all the menu items work appropriately</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td>Buttons/Links</td>
<td>• Ensure all buttons and links work as expected</td>
<td>Pass</td>
</tr>
<tr>
<td>Insert New Student</td>
<td>User Input</td>
<td>• Verify that all fields take expected input</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Test invalid data</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Menu</td>
<td>• Make sure all the menu items work appropriately</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td>Buttons/Links</td>
<td>• Ensure all buttons and links work as expected</td>
<td>Pass</td>
</tr>
<tr>
<td>Update/View Staff Info</td>
<td>User Input</td>
<td>• Verify that all fields take expected input</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Test invalid data</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Menu</td>
<td>• Make sure all the menu items work appropriately</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td>Buttons/Links</td>
<td>• Ensure all buttons and links work as expected</td>
<td>Pass</td>
</tr>
<tr>
<td>Update Contact Info</td>
<td>User Input</td>
<td>• Verify that all fields take expected input</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Test invalid data</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Menu</td>
<td>• Make sure all the menu items work appropriately</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td>Buttons/Links</td>
<td>• Ensure all buttons and links work as expected</td>
<td>Pass</td>
</tr>
<tr>
<td>View Student Info</td>
<td>Menu</td>
<td>• Make sure all the menu items work appropriately</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td>Buttons/Links</td>
<td>• Ensure all buttons and links work as expected</td>
<td>Pass</td>
</tr>
</tbody>
</table>

4.2 Test Scenarios

Test scenarios are useful to determine if functional requirements have been met within the user interface. In the following sections, a scenario will be given, then screen shots will display the results.
4.2.1 Main User Login

This page presents the starting point for the system. A student, faculty, Graduate Coordinator, and Program Assistant will be required to enter a user-id and password initially created by the Web-Based Database Management System System Administrator.
User Input:
• User ID
• Password

Output:
1. Success -
   Verify the user id and password, then redirect to home page of appropriate user role:
   • Faculty
   • Student
   • Graduate Coordinator
   • Program Assistant

2. Failure -
   • Incorrect user id/password
   • Forget password
   Type in the user id and click on “Send” button

Figure 3. User Login
4.2.2 Faculty / Graduate Coordinator / Program Assistant / Student: Change Password

This function allows users to change their passwords as frequently as desired.

User Input:
- Type the user’s new password in both “New Password” and “Confirm New Password” input boxes
- Click on the “Submit” button

Output:
1. Success -
   Your User ID / Password has been changed Successfully!

2. Failure -
   - New password and confirm password are the same

Figure 4. Change Password
4.2.3 Faculty / Graduate Coordinator: Communications

This module allows the faculty and/or Graduate Coordinator to email different groups. The functionality for each of the particular groups will be provided in detail in the following sections.

4.2.3.1 All Advisees. This function allows the faculty or Graduate Coordinator to send email to all his or her current advisees.

User Input:
- Click on the “All Advisees” hyperlink

A pop-up page where FROM field is automatically filled with the sender’s (faculty or graduate coordinator) email address; TO field is automatically filled with email addresses of pertinent advisees; and SUBJECT and CONTENTS fields will be filled out by the sender.

Figure 5. Communications - All Advisees
4.2.3.2 Select Advisees. This function allows the faculty or Graduate Coordinator to send email to advisees selected from a list of current advisees.

User Input:
- Click on the "Select Advisees" hyperlink

Output:
- A pop-up page where FROM field is automatically filled with the sender’s (faculty or graduate coordinator) email address; TO field presents a list of advisees that the sender needs to select from; and SUBJECT and CONTENTS fields will be filled out by the sender

Figure 6. Communications - Select Advisees
4.2.3.3 All Committees. This function allows faculty or Graduate Coordinator to send email to all project or thesis committee members of his / her advisees.

User Input:
- Click on the “All Committees” hyperlink

Output:
- A pop-up page where FROM field is automatically filled with the sender’s (faculty or graduate coordinator) email address; TO field is automatically filled with email addresses of pertinent advisees; and SUBJECT and CONTENTS fields will be filled out by the sender.

Figure 7. Communications – All Committees
4.2.3.4 Select Committees. This function allows faculty or the Graduate Coordinator to send email to certain committee members selected from a list.

User Input:
- Click on the "Select Committees" hyperlink

Output:
- A pop-up page where FROM field is automatically filled with the sender’s (faculty or graduate coordinator) email address; TO field presents a list of advisees that the sender needs to select from; and SUBJECT and CONTENTS fields will be filled out by the sender

Figure 8. Communications - Select Committees
4.2.3.5 Graduate Coordinator. This function permits faculty members to send email to the Graduate Coordinator.

User Input:
- Click on the "Graduate Coordinator" hyperlink

Output:
- A pop-up page where FROM field is automatically filled with the sender's (faculty) email address; TO field is automatically filled with email address of the Graduate Coordinator; and SUBJECT and CONTENTS fields will be filled out by the sender

Figure 9. Communications - Graduate Coordinator
4.2.3.6 Department Chair. This function allows the faculty or Graduate Coordinator to send email to the Department Chair.

User Input:
• Click on the "Department of Chair" hyperlink

Output:
• A pop-up page where FROM field is automatically filled with the sender's (faculty, Graduate Coordinator) email address; TO field is automatically filled with email address of the Department Chair; and SUBJECT and CONTENTS fields will be filled out by the sender

Figure 10. Communications - Department of Chair
4.2.3.8 Dean of Graduate Studies. This function allows the faculty or Graduate Coordinator to send email to the Dean of the Graduate Studies.

User Input:
- Click on the “Dean of Graduate Studies” hyperlink

Output:
- A pop-up page where FROM field is automatically filled with the sender’s (faculty, Graduate Coordinator) email address; TO field is automatically filled with email address of the Dean of the Graduate Studies; and SUBJECT and CONTENTS fields will be filled out by the sender

Figure 11. Communications - Dean of Graduate Studies
4.2.3.9 Program Assistant. This function allows the faculty or Graduate Coordinator to send email to the Program Assistant.

User Input:
- Click on the "Program Assistant" hyperlink

Output:
- A pop-up page where FROM field is automatically filled with the sender's (faculty, Graduate Coordinator) email address; TO field is automatically filled with email address of the Program Assistant; and SUBJECT and CONTENTS fields will be filled out by the sender

Figure 12. Communications - Program Assistant
4.2.4 Faculty / Graduate Coordinator / Program Assistant / Student: Change Home Page

This function allows a user to change his/her home page as often as she or he wishes.

User Input:
• Choose one of the pages (Change Password, Communications, Customize Home Page, View Any Student Info, and View Own Advisees Only) from the drop-down menu
• Click the “Confirm” button

Output:
• When the user clicks on reload/refresh button in the browser or clicks on the “Click Here” hyper link, the system automatically refreshes the screen to the new home page

Figure 13. Change Home Page
This function allows the faculty, Graduate Coordinator, or Program Assistant to view any student information.
User Input:
- Choose the student name in the drop-down menu
- Type in the student’s last name and click on the “Last Name” radio button
- Type in the student’s first name and click on the “First Name” radio button to initiate the search string then click on the “Search and View” button

Output:
- On the top - Student’s personal information is displayed
- On the bottom - a summary of the student’s academic information

1. Click on “Course & Grades” A screen displays the core courses, elective courses and prerequisite courses the student has taken and the grades earned

Figure 14. View Any Student Information (Part A)
2. Click on "Prerequisite Exams" A screen displays the prerequisite exams (such as GRE, TOEFL, and WREE) the student has taken and the scores earned.

3. Click on "Probation History" A screen displays the student's probation history (starting date, starting term, ending date, ending term, and reason for being on probation).

4. Click on "Master Option" A screen displays the student's master option information.

5. Click on "Grad Check" A screen displays the student's grad check information.

---

Figure 15. View Any Student Information (Part B)
4.2.6 Faculty / Graduate Coordinator: View Own Advisees Only

This function allows the faculty or Graduate Coordinator to view information about own advisees only.

User Input:
- None

Output:
- On the top - The information (name, term advanced to candidacy, master option, and project/thesis title) about current advisees is displayed
- Click on “View Graduated Advisees” on the top A screen displays information for this faculty’s advisees who have already graduated
- Click on “View Graduated Advisees” on the top A screen with the information about graduated advisees (such as name, term advanced to candidacy, master option, and project/thesis title)
- On the bottom - The information (name, term advanced to candidacy, master option, and project/thesis title) about current advisees where the faculty is a committee member
- Click on “View Graduated Advisees” on the bottom A screen displays information about advisees who have graduated and where the faculty is a committee member

Figure 16. View Own Advisees Only
4.2.7 Graduate Coordinator: Academic Year Statistics

This function allows the Graduate Coordinator to view statistics on students with respect to admission, and academic standing (classified, advanced to candidacy, on probation, dismissed, and graduated).

User Input:
- Choose the academic year from the drop-down list
- Click the “Submit” button

Output:
- The statistical report - showing the number of students admitted, classified, advanced to candidacy, on probation, graduated and dismissed. The total is further broken down by gender and whether the student is an international student or not.

Figure 17. Academic Year Statistics

4.2.8 Graduate Coordinator: Generate Reports - Pre-Defined Reports

This function allows the Graduate Coordinator to generate pre-defined reports. A pre-defined report is a
report that must be produced on a regular basis, at least once a quarter.

User Input:
- For "Reports on Attendance Status"
  Check on one of the radio buttons in (Active, Inactive Attended, Never Attended, Incoming, On Probation, Dismissed, and Alumni)
- For "Reports on Academic Standing"
  Check on one of the radio buttons (Advanced to Candidacy, Classified, Conditionally Classified, and Probationally Conditionally Classified)
- Click the "Submit" button

Output:
Reports on Attendance Status -
- displays the student id, name, gender, admit status, and term admitted in ascending student id order; to
- change the sort field for display click on any column title

Figure 18. Generate Reports - Pre-Defined Reports (Part A)
### 2. Inactive

**Graduate Program Management System**

**Attendance Status Report - Inactive Students**

<table>
<thead>
<tr>
<th>Student ID</th>
<th>Name</th>
<th>Status</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>12345678</td>
<td>John Doe</td>
<td>Inactive</td>
<td>01/01/2023</td>
</tr>
</tbody>
</table>

### 3. Never Attended

**Graduate Program Management System**

**Attendance Status Report - Never Attended Students**

<table>
<thead>
<tr>
<th>Student ID</th>
<th>Name</th>
<th>Status</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>98765432</td>
<td>Jane Smith</td>
<td>Never</td>
<td>01/01/2022</td>
</tr>
</tbody>
</table>

### 4. Incoming

**Graduate Program Management System**

**Attendance Status Report - Incoming Students**

<table>
<thead>
<tr>
<th>Student ID</th>
<th>Name</th>
<th>Status</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>87654321</td>
<td>Mike Johnson</td>
<td>Incoming</td>
<td>01/01/2021</td>
</tr>
</tbody>
</table>

### 5. On Probation

**Graduate Program Management System**

**Attendance Status Report - On Probation Students**

<table>
<thead>
<tr>
<th>Student ID</th>
<th>Name</th>
<th>Status</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>65432109</td>
<td>Lisa Morgan</td>
<td>On Probation</td>
<td>01/01/2020</td>
</tr>
</tbody>
</table>

### 6. Dismissed

**Graduate Program Management System**

**Attendance Status Report - Dismissed Students**

<table>
<thead>
<tr>
<th>Student ID</th>
<th>Name</th>
<th>Status</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>09876543</td>
<td>Sam Brown</td>
<td>Dismissed</td>
<td>01/01/2019</td>
</tr>
</tbody>
</table>

Figure 19. Generate Reports - Pre-Defined Reports (Part B)
7. Alumni

- Reports on Academic Standing
  1. Advanced to Candidacy - displays the student id, name, gender, term advanced to candidacy, advisor, and committee members in ascending student id order; to change the sort field for display click on any column title that is underlined.
  2. Classified Status - displays the student id, name, gender, term classified, term admitted and term status in ascending student id order; to change the sort field for display click on any column title that is underlined.
  3. Conditionally Classified Status - displays the student id, name, gender, term classified, term admitted and admit status in ascending student id order; to change the sort field for display click on any column title that is underlined.

Figure 20. Generate Reports - Pre-Defined Reports (Part C)
4. Probationally Conditionally Classified Status - displays the student id, name, gender, term classified, term admitted and admit status in ascending student id order; to change the sort field for display click on any column title that is underlined.

Figure 21. Generate Reports - Pre-Defined Reports (Part D)

4.2.9 Graduate Coordinator: Generate Reports – Ad Hoc Reports

This function allows the Graduate Coordinator to generate an ad hoc report. An ad hoc report is a formatted output created as needed and usually produced by an after-the-fact reporting system designed for the purpose. Once an ad hoc report is created it can be saved and recalled for later use.
User Input:
New Reports –
1. Check on the radio button New Reports in the "Ad Hoc Reports" section to create a new ad hoc report.
2. Click the "Submit" button.
3. Fill in the title of the report.
4. From the drop-down list, select the fields to be displayed in the report.
5. Select which of the display fields will be used to sort the report. By the default, sort is in ascending order. If descending order is desired, then check the descending box field.

Figure 22. Generate Reports – Ad Hoc Reports (Input)
Output:
- A page displays the student ID, student name, gender, and the terms when the student is admitted, classified, and advanced to candidacy.

Saved Reports:
1. Choose the name of the report from the drop-down list. The SQL statement that pertains to the saved report is displayed.
2. Click the "Submit" button.

Output:
- A page displays an ad hoc report.

Figure 23. Generate Reports – Ad Hoc Reports (Output)

4.2.10 Graduate Coordinator: Quarterly Reports

This function allows the Graduate Coordinator to generate the following quarterly reports—Students with C- grade in CSCI courses; Students with GPA below 3.0.
Students who are off the probation List; Students who are Preliminary Advanced to Candidacy; Students who have taken less than max 5 units for CSCI690, less than max 9 units for CSCI699; and Students who should be in continuous enrollment.

User Input:
1. Choose the term from the left-hand side drop-down list
2. Choose the quarterly report name from the right-hand side drop-down list

Output:
3. A page displays the student id, name, email address, telephone number, course name and quarter/year when a C- was earned.

Figure 24. Quarterly Reports
4.2.11 Graduate Coordinator / Program Assistant:
View Any Advisee

This function allows the Graduate Coordinator to view information concerning any advisee under any specific faculty advisor.

User Input:
1. Choose the advisor's name from the left-hand side drop-down list
2. Choose the advisee from the right-hand side drop-down list (current advisees, graduated advisees, current advisees in committee, graduated advisees in committee)

Output:
A page lists the current advisees providing student's name, quarter/year advanced to candidacy, master option and the title of the project or thesis

Figure 25. View Any Advisee
4.2.12 Graduate Coordinator / Program Assistant: Update Student Info

This function allows the Graduate Coordinator or Program Assistant to update a student’s information.

User Input:
1. Choose the student’s name from the drop-down list
Or
Type in the student’s last name and click on the “Last Name” radio button
Or
Type in the student’s first name and click on the “First Name” radio button to initiate the search string
2. Click on the “Search and View” button

Output:
Click “Summary”
To update student’s term admitted, term re-admitted, term classified, term advanced to candidacy, term last attended, admission status, academic standing, current status, master option, date and scores for GRE (Verbal, Quantitative, Analytical)

Figure 26. Update Student Information (Part A)
Click on “Personal”
To update student’s personal information such as current USA address, phone number and email address

Click on “Exams”
To update student’s score and date taken for GRE, TOEFL and WREE exams

Click on “Courses Taken”
The only thing that can be updated is if this course needs to be discounted since the student received an unsatisfactory score in the first attempt. Check the “Discount” box if this course has been approved by Dean of Graduate Studies for a discount.

Click on “Committee”
To update the student’s advisor and committee members

Figure 27. Update Student Information (Part B)
Click on "Probation History"
To update student's probation history such as starting date, starting term, ending date, ending term and reason why the student was put on probation.

Click on "Master Option"
To update student's master option information.

Click on "Grad Check"
To update the date the student submitted for format check, date of binding receipt, and date when grad check was filed.

Figure 28. Update Student Information (Part C)

4.2.13 Program Assistant: Comp Exam Info
This function allows the Program Assistant to insert or update information on the comprehensive examination: quarter/year it is given, the faculty designated to make
the exam for each of the core courses (CSCI600, CSCI610, CSCI 630, CSCI655 and CSCI660).

User Input:
To add the new information -
1. Choose the term, course name, and faculty name from the lower drop-down lists
2. Click the "Add" button

To update the existing information -
Choose the term, course name and faculty name from the upper drop-down lists

To delete the existing information -
Click the "Remove" button

Output:
Displays the comprehensive examination for the quarter/year indicated

Figure 29. Comprehensive Exam Information
4.2.14 Program Assistant: Insert New Faculty

This function allows the Program Assistant to insert information on new faculty who have been hired or for persons who are not from CSUSB but who will be the student’s advisor or will be in the student’s committee.
User Input:
1. Enter the new faculty’s title, last name, first name, middle name, email, office phone, position, and status. If the individual is a CSCI faculty member, place a check mark next to the box asking whether the person is CSCI faculty. If not, type in the new faculty affiliation, department, and institution. This is required for persons who will be in student’s committee.
2. Click the “Add” or “Cancel” button.

Output:
Add - A screen with the successful message.
Cancel - Return to the previous page.

Figure 30. Insert New Faculty
4.2.15 Program Assistant: Insert New Staff

This function allows the Program Assistant to insert information on a staff person who has been hired.

User Input:
1. Enter the last name, first name, middle name, email, office phone, position, status of the staff person hired
2. Click the "Add" or "Cancel" button

Output:
Add —
A screen with a successful message

Cancel —
Return the previous page

Figure 31. Insert New Staff
4.2.16 Program Assistant: Print Letters

This function allows the Program Assistant to print the letters that need to be sent to the students.

<table>
<thead>
<tr>
<th>User Input:</th>
<th>Output:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Choose the student’s name from the drop-down list</td>
<td>3. A pop-up page displaying the content of the letter from the clicked hyperlink. The Program Assistant must make sure that a department letter head is on the printer paper bin.</td>
</tr>
<tr>
<td>2. Click on the hyperlink</td>
<td>Dear Oi-Bun, Our records indicate that you received a C- in CSCI660 Operating Systems Spring 2002 A C- is not acceptable in the MS CSCI program. You will need to retain the course(s) and get a better grade. As soon as you get a better grade, you will need to request to discount this grade. You will be allowed to request a discount only for one course. Please contact me if you are ready to request a grade discount. Sincerely, Josephine Mendoza, Ph.D. Graduate Coordinator for Advising mscs <a href="mailto:advising@csusb.edu">advising@csusb.edu</a> (909) 537-5326.</td>
</tr>
</tbody>
</table>
This function allows the Program Assistant to update or view faculty or staff information.

User Input:
1. Choose the faculty or staff member's name from the drop-down list
   Or
   Type in the faculty or staff member's last name and click on the "Last Name" radio button
   Or
   Type in the faculty or staff member's first name and click on the "First Name" radio button to initiate the search string
2. Click on the "Search and View" button

Output:
3. A page displays the information on the faculty or staff

Figure 33. Update/View Staff Information
4.2.18 Student: Communications

This module allows the student to email different groups. The functionality for each of the particular groups will be provided in detail in the following sections.

4.2.18.1 All Committee Members. This function allows the student to send emails to his or her committee members.

User Input:
- Click on the “All Committee Members” hyperlink

Output:
- A pop-up page where FROM field is automatically filled with the sender’s (student) email address; TO field is automatically filled with email addresses of pertinent committee members; and SUBJECT and CONTENTS fields will be filled out by the sender

Figure 34. Communications - All Committee Members
4.2.18.2 Select Committee Members. This function allows the student to send email to certain committee members selected from the list.

User Input:
• Click on the "Select Committee Members" hyperlink

Output:
• A pop-up page where FROM field is automatically filled with the sender's (student) email address; TO field presents a list of committee member that the sender needs to select from; and SUBJECT and CONTENTS fields will be filled out by the sender.

Figure 35. Communications - Select Committee Members
4.2.18.3 Advisor Only. This function allows the student to send an email to his or her advisor.

User Input:
- Click on the "Advisor Only" hyperlink

Output:
- A pop-up page where FROM field is automatically filled with the sender’s (student) email address; TO field is automatically filled with the email address of the advisor; and SUBJECT and CONTENTS fields will be filled out by the sender

Figure 36. Communications - Advisor Only
4.2.18.4 Graduate Coordinator. This function allows the student to send an email to the Graduate Coordinator.

User Input:
- Click on the "Graduate Coordinator" hyperlink

Output:
- A pop-up page where FROM field is automatically filled with the sender's (student) email address; TO field is automatically filled with email address of the Graduate Coordinator; and SUBJECT and CONTENTS fields will be filled out by the sender

Figure 37. Communications - Graduate Coordinator
4.2.18.5 Department Chair. This function allows the student to send an email to the Department Chair.

User Input:
- Click on the "Department Chair" hyperlink

Output:
- A pop-up page where FROM field is automatically filled with the sender's (faculty or graduate coordinator) email address; TO field is automatically filled with email address of the Department Chair; and SUBJECT and CONTENTS fields will be filled out by the sender

Figure 38. Communications - Department Chair
4.2.18.6 Dean of Graduate Studies. This function allows the student to send an email to the Dean of Graduate Studies.

User Input:
- Click on the "Dean of Graduate Studies" hyperlink

Output:
- A pop-up page where FROM field is automatically filled with the sender's (faculty or graduate coordinator) email address; TO field is automatically filled with email address of the Dean of Graduate Studies; and SUBJECT and CONTENTS fields will be filled out by the sender

Figure 39. Communications - Dean of Graduate Studies
4.2.18.7 Program Assistant. This function allows the student to send an email to the Program Assistant.

User Input:
- Click on the "Program Assistant" hyperlink

Output:
- A pop-up page where FROM field is automatically filled with the sender's (faculty or graduate coordinator) email address; TO field is automatically filled with email address of the Program Assistant; and SUBJECT and CONTENTS fields will be filled out by the sender

Figure 40. Communications - Program Assistant
4.2.18.8 Select CSCI Faculty / Instructors. This function allows the student to send emails to selected CSCI faculty or instructors.

User Input:
- Click on the "Select CSCI Faculty / Instructors" hyperlink

Output:
- A pop-up page where the FROM field is automatically filled with the sender's (student) email address; TO field presents a list of CSCI faculty that the sender needs to select from; and SUBJECT and CONTENTS fields will be filled out by the sender

Figure 41. Communications - Select Computer Science Faculty
This function permits students to update their email address, telephone numbers, or current address.

User Input:
1. Enter the email address, phone number, current address, city, state, zip
2. Click the "Update" or "Cancel" button

Output:
A page displaying the personal information
Update - A screen displays the updated personal information
Cancel - A screen displays the old personal information

Figure 42. Update Contact Information
4.2.20 View Student Info

This function allows the student to view his or her personal and academic information.

User Input:
None
Output:
On the top -
Student's personal information, such as last name, first name, middle name, date of birth, ethnicity, phone number, email, current address, and permanent address.

On the bottom -
Student's academic information.
Click on "Summary" to view a screen with a student's term admitted, term classified, term advanced to candidacy, term last attended, admission status, current status, academic standing, master option, GRE date and score.

Click on "Course & Grades" to view a screen with the courses (including core course, elective courses, and prerequisite courses) which the student has taken and the grade which the student has earned.

Figure 43. View Student Information (Part A)
Click on “Prerequisite Exams”
A screen with the prerequisite exams (such as GRE, TOEFL, and WREE) which the student has taken and the scores that he/she earned

Click on “Probation History”
A screen with the student’s probation history (staring date, starting time, ending date, ending term, and probation reason)

Click on “Master Option”
A screen with the student’s master option information

Click on “Grad Check”
A screen with the student’s graduate checking information (such as date of format check, date of binding receipt, and date of grad check)

Figure 44. View Student Information (Part B)
CHAPTER FIVE

CONCLUSIONS

5.1 Project Accomplishments

This project, the Web-Based Database System for the Computer Science Graduate Program, has been designed and implemented by correcting the deficiencies found in Graduate Database Management System developed by Dung Vu and added new functionalities to address the changes in the Computer Science graduate program policies and California State University, San Bernardino requirements. Appropriate data structures and up-to-date technologies have been incorporated in this project: vector, SQL language, JavaServer Pages, Java Bean, a relational database model, and XML technology.

The application of the Oracle technology in storing as well as in re-constructing SQL queries helps facilitates reporting, useful in functions such as the generation of ad-hoc reports. The database management system developed and deployed in this project has been made robust, reliable and protected from potential breaches by the incorporation of security techniques: password enforcement, encrypted passwords, transmission encryption with SSL (https), placement of the database
within a private network and behind a firewall, use of authentication and authorization functions. Execution privilege of a function no longer relies upon hard-coding or fixed roles; instead, a vector data structure is used at program run time for authorization verification.

The use of modules makes the system flexible, easy to maintain and accommodate changes and new requirements with minimal programming workaround. New framework architecture complements system functions simply by registering new functions and assigning these to appropriate roles. The use of Java and JSP has made the system portable to multiple operating systems.

Most of the processing for this system has been automated. The system can monitor pre-determined and designated events like monitoring of GPA and status update through the use of SQL triggers and procedures.

This system can now be accessed 24 x 7 by not only the Graduate Coordinator and Program Assistant but by Computer Science faculty and graduate students. Graduate students can check their status in the program and communicate with their advisor and committee members, Graduate Coordinator, Program Assistant, other faculty, department chair and the dean of Graduate Studies without knowing the email addresses. The interfacing with SIS+ has
removed the manual input of student course grades by the Program Assistant.

5.2 Future Directions

There are features which could be added to the database system, but which, due to the scope of my master project, have not been implemented. These nice-to-have modules are listed below.

5.2.1 Centralized Management of Documents

The system will allow graduate students to upload their project or thesis documents, thus permitting the sharing of such documents with faculty, PA, and GCA utilizing a centralized database. The system can readily manage system-generated documents. While this functionality provides for ready access by administrative users, this centralization of the document management system also serves as an electronic secondary archive for student documents.

5.2.2 Appointment Maker

The system will enable the scheduling of appointments for students to meet with specific faculty members; in particular their advisor and committee members as well as the Graduate Coordinator for Advising.
5.2.3 Announcement

Announcements intended for faculty, students, and specific special interest groups will be facilitated by a posting system.

5.2.4 Frames versus Table Format

Currently, my project uses frames within the web page structure rather than tables. Both formats possess certain advantages and disadvantages, and an analysis of these aspects can help modify the next version.

Frames allow access to data without the need for scrolling up or down in order to view functions on the left side of the relevant screens. However, frames do limit the potential size of the user interface due to screen size. In contrast, table use does not have screen size limitation. However, users must utilize scrolling up and down to view needed functions. It is my belief that tables represent a better choice, as screen size is enhanced even though some scrolling is necessary. Improved size represents a more central consideration, as scrolling can be easily accomplished.
APPENDIX A

DATA DICTIONARY
A.1 Look-Up Tables

1. _ACADSTANDING_. Define academic standing of students -- probation, conditionally classified, classified, or advanced to candidacy.

   STANDING_ID [VARCHAR2 (10), PK]: Academic standing code
   STANDING [VARCHAR2 (30)]: Description of academic standing code -- probation, conditionally classified, classified, or advanced to candidacy.
   DEF_STANDING [VARCHAR2 (500)]: Definition of academic standing.

2. _ADMITSTATUS_. Define admission status of students -- classified, conditionally classified, probation classified, or probation conditionally classified.

   ADMIT_ID [VARCHAR2 (10), PK]: Admission status code
   ADMIT_STATUS [VARCHAR2 (30)]: Description of admission status code -- classified, conditionally classified, probation classified, or probation conditionally classified.
   DEF_ADMIT_STATUS [VARCHAR2 (500)]: Definition of admission status.

3. _COUNTRIES_. Define countries where international students come from.

   COUNTRY_ID [CHAR (4), PK]: SIS+ country Code.
   COUNTRY [VARCHAR2 (50)]: SIS+ country name, which represent international students' countries.

4. _COURSES_. Define courses taught in the graduate program including recommended, prerequisite, elective and core courses.

   COURSE_ID [VARCHAR2 (15), PK]: Course identification number
   COURSE_NAME [VARCHAR2 (50)]: Course name.
   COURSETYPE [CHAR (1), FK]: Course type code (for core, elective, prerequisite, recommend, and other course), has referential integrity constraint with COURSETYPES table.
   UNITS [CHAR (1)]: Course units.
   DESCRIPTION [VARCHAR2 (1024)]: Course Description.
   GROUPID [CHAR (1)]: Course group ID.
   QTR_OFFER [VARCHAR2 (4)]: Course offer quarter. F for Fall, W for Winter, S for Spring, and X for Summer.
IS_GRADE_STORE [CHAR (1)]: A (Y-N) flag field to indicate whether it is a course which grade will be stored in departmental database.

IS_MS [CHAR (1)]: A (Y-N) flag field to indicate whether it is a course which will be scheduled in study plan.

5. COURSETYPES. Define courses taught in the graduate program including prerequisite, core, elective, recommend and all other courses.

COURSETYPE_ID [CHAR (1), PK]: Course type code to define course type -- P for Prerequisite course, C for Core course, E for Elective course, R for recommend course, O for all others.

COURSETYPE [VARCHAR2 (15)]: Description of course type -- Prerequisite course, Core course, Elective course, Recommend course, and other course.


CUR_STATUS_ID [VARCHAR2 (10), PK]: Current status code.


DEF_CUR_STATUS [VARCHAR2 (500)]: Definition of current status of student.

7. CY/Core. Store core courses requirement for each academic catalog year.

YR_CATALOG [VARCHAR2 (4), PK]: Academic catalog year, a composite primary key.

COURSE_ID [VARCHAR2 (4), PK, FK]: Course identification (ID) number, a composite primary key has referential integrity constraint with COURSES table.

8. CY_PREQ. Store prerequisite courses requirement for each academic catalog year.

YR_CATALOG [VARCHAR2 (4), PK]: Academic catalog year, a composite primary key.

COURSE_ID [VARCHAR2 (4), PK, FK]: Course identification (ID) number, a composite primary key has referential integrity constraint with COURSES table.

IS_ALL_REQ [VARCHAR2 (1)]: A (Y-N) flag field to indicate whether it is a prerequisite course which all students are required to take.
IS_COUNT [VARCHAR2 (1)]: A (Y-N) flag field to indicate the units of this course whether can be counted into graduate units.

9. ETHNICSORGN. Define ethnic origin of a student.

ETHNICITY_ID [VARCHAR2 (10), PK]: A code representing the racial or ethnic background of the individual. The individual is to be included in the ethnic group to which he or she appears to belong, is regarded by the community as belonging or categorized him or herself as belonging.

ETHNICITY [VARCHAR2 (60)]: Ethnic origin group.

ETHNIC_DEFINITION [VARCHAR2 (250)]: Ethnic code associated definition and description.

10. EXAMINFO. Store information and rules regarding each specific type of exam, such as WREE, TOEFL, GRE.

EXAM_NAME [VARCHAR2 (10), PK]: Exam name
DESCRIPTION [VARCHAR2 (1000)]: EXAM description
PASS_SCORE [VARCHAR2 (1000)]: Passing Score and rules.

11. GRADES. Define grades and corresponding scores.

GRADE_ID [VARCHAR2 (3), PK]: A code representing the grade assigned to the student for this course.
SCORES [NUMBER (3,1)]: Grade ID corresponding scores.
GRADEID_DEFINITION [VARCHAR2 (30)]: Grade ID definition.

12. GRAD_UNITS_REQ. Store graduate units’ requirement for each academic catalog year.

YR_CATALOG [VARCHAR2 (4), PK]: Academic catalog year.
TOTAL_U_REQ [NUMBER (2)]: Academic catalog year.
TOTAL_500_ALLOWED [NUMBER (2)]: Maximum counted units of 500-leve courses.

13. PREQ_CONSTRAIN. Store prerequisite constraints on prerequisite course

CONSTRAIN_SEQ [VARCHAR2 (5)]: Sequence number, generated by selecting SEQUENCE PREQCONSTRAIN_SEQ, start with 10000 and increment by 10.

COURSE_ID [VARCHAR2 (15), PK, FK]: Course identification number which a composite primary key has referential integrity constraint with COURSES table.
PREQ_COURSEID  [VARCHAR2 (15), PK, FK]: prerequisite constraint course identification number which a composite primary key has referential integrity constraint with COURSES table.

14. QUARTER. Defines quarters (fall, winter, spring, summer) when courses are taken. In semester system there are only fall and spring values.

QUARTER_ID  [CHAR (1), PK]: Last digit of term code -- 1 for Winter, 2 for Spring, 3 for Summer, 4 for Fall.

QUARTER  [VARCHAR2 (10)]: Winter, Spring, Summer, and Fall.

15. ROLES. Define roles of users. Currently the system has five roles: Student, Faculty, Coordinator, MS Program Assistant, Administrator and System.

ROLE_ID  [CHAR (1), PK]: ID to define user's role.

ROLE  [VARCHAR2 (30)]: Student, Faculty, Coordinator, MS Program Assistant, Administrator and System.

16. SIS_AA651_EXIT_ACTION. SIS+ Extract -- store the action which brought about the termination of this matriculation. This field defaults to the Exit Action defined on the Exit Action Table for either degree or non-degree programs.

EXIT_ACTION_ID  [VARCHAR2 (10), PK]: SIS+ exit action code.

EXIT_ACTION  [VARCHAR2 (80)]: Description of SIS+ exit action code.

DEF_EXIT_ACTION  [VARCHAR2 (500)]: Definition of SIS+ exit action code.

17. SIS_EXTRACT_LAYOUT. Store record layouts for SIS+ extract segments data layout.

18. SIS_RT047_ATTEND_STAT. SIS+ Extract -- student classification code and description.

CLASSIFICATION_ID  [VARCHAR2 (10), PK]: SIS+ classification code

CLASSIFICATION  [VARCHAR2 (80)]: Description of SIS+ classification code.

DEF_CLASSIFICATION  [VARCHAR2 (500)]: Definition of SIS+ classification code.

19. SIS_RT080_ACAD_ACTION. SIS+ Extract -- Any academic action that has been taken on this student.

ACAD_ACTION_ID  [VARCHAR2 (10), PK]: SIS+ academic action code.

UNOFFICIAL TRANSLAT  [VARCHAR2 (80)]: Description of SIS+ academic action code on UNOFFICIAL TRANSCRIPT TRANSLATION.

DEF_UNOFFICIAL TRANSLAT  [VARCHAR2 (500)]: Definition of SIS+ academic action code on UNOFFICIAL TRANSCRIPT TRANSLATION.
OFFICIAL_TRANSLAT  [VARCHAR2 (80)]: Description of SIS+ academic action code on OFFICIAL TRANSCRIPT TRANSLATION.

DEF_OFFICIAL_TRANSLAT  [VARCHAR2 (500)]: Definition of SIS+ academic action code on OFFICIAL TRANSCRIPT TRANSLATION.

20.  SIS_RT14C_CLASSIFICATION. SIS+ Extract -- student classification code and description.

CLASSIFICATION_ID [VARCHAR2 (10), PK]: SIS+ classification code.

CLASSIFICATION  [VARCHAR2 (80)]: Description of SIS+ classification code.

DEF_CLASSIFICATION  [VARCHAR2 (500)]: Definition of SIS+ classification code.

21.  SIS_RT310_GRADE_TYPE. SIS+ Extract -- The type of grading method selected by the student for this course or assigned by the registrar’s staff to handle special grading considerations. These values are institution-related.

GRADE  [VARCHAR2 (10), PK]: SIS+ code of grading method type.

TYPE  [VARCHAR2 (80)]: Description of type of grading method.

DEF_GRADE  [VARCHAR2 (200)]: Definition of SIS+ type of grading method.

22.  SIS_RT312_GRADE_RULE SIS+ Extract -- The Grade Rule applicable to this SPE. If a course section has a Grade Rule, this field will be initialized from the course section at the time of registration; otherwise the Grade Rule for the student academic program will be used.

GRADE  [VARCHAR2 (10)]: SIS+ grad rule code.

RULE  [VARCHAR2 (30)]: Description of SIS+ grad rule.

23.  SIS_RT345_CLASS. SIS+ Extract -- The values reflect the student’s status in the section. Active requests, attempts, permits, authorizations and drops are included. All activity will be retained in the audit trail. No permanent record (transcript) notation will be made of drops unless they occur during the withdrawal grade period. Based on the date in the calendar table, a course dropped after the deadline for withdrawal grades will be changed to enrolled with a grade of W.

STATUS_ID  [VARCHAR2 (10)]: SIS+ class enrollment status code.

REASON  [VARCHAR2 (30)]: Description of SIS+ class enrollment status code.
24. **TERMS.** Define 3 digits SIS+ term code in ‘YYQ’ format which represents year and quarter.

**TERMCODE** [VARCHAR2 (5), PK]: Term code, first two digits represent year and last digit represents quarter: 1-Winter, 2-Spring, 3-Summer, 4-Fall.

**DESCRIPTION** [VARCHAR2 (20)]: Description of term code

25. **YR_LOOK UP.** Store 4 digits year value for application drop-down list reference.

**YR** [CHAR (4), PK]: Academic catalog year

26. **ADHOCLOG.** Stores the SQL statements which the user creates in the ad hoc reports

**USER_ID** [VARCHAR2(10)]: Identification number to verify the authorized user

**SQLNAME** [VARCHAR2(25)]: Name of SQL statement which the user creates in the ad hoc reports

**SQLSTRING** [VARCHAR2(1000)]: SQL statement which the user creates

27. **ADV_COMMENT.** Store advising comment on each course of Advancement to Candidacy Graduate Approved Program of Study Form.

**COMMENT_SEQ** [VARCHAR2 (10), PK]: Sequence number, generated by selecting SEQUENCE ADVCOMMENT_SEQ, start with 10000 and increment by 1.

**STD_ID** [CHAR (9), FK]: Student Identification Number which has referential integrity constraint with GRADSTUDENTS table.

**COURSE_ID** [VARCHAR2 (15), FK]: Course identification (ID) number, has referential integrity constraint with COURSES table.

**STAFF_ID** [CHAR (9), FK]: Staff Identification Number which has referential integrity constraint with STAFF table.

**COMMENT_DATE** [DATE, NOT NULL]: Comment’s insertion date.

**COMMENTS** [VARCHAR2 (500)]: Content of comment

28. **ADV NOTE.** Store advising notes/comment of Graduate Advising Sheet. Each student has as many records as his/her advisor has written.

**STD_ID** [CHAR (9), FK]: Student Identification Number which has referential integrity constraint with GRADSTUDENTS table.
STAFF_ID [CHAR (9), FK]: Staff Identification Number which has referential integrity constraint with STAFF table.

QTR_ADV [CHAR (1)]: Quarter of advising note made.

YR_ADV [VARCHAR2 (4)]: Year of advising note made.

NOTE_DATE [DATE]: Note's insertion date.

NOTE [VARCHAR2 (500)]: Content of note/comment

29. COMEXAM. Store master comprehensive exam track information.

STD_ID [CHAR (9), PK, FK]: Student Identification Number which has referential integrity constraint with GRADSTUDENTS table.

COM_DATE [DATE]: Comprehensive exam date.

IS_COM_PASS [CHAR (1)]: Y if pass; N if fails. Data value derives from total score.

TOTALSCORE [NUMBER (5,2)]: Comprehensive exam total score. Data value derives from summing up each subject score.

CS600 [NUMBER (5,2)]: CSCI 600 exam score.

Q600_FAC [CHAR (9)]: Staff-ID of faculty, who provides CSCI 600 exam questions which has referential integrity constraint with STAFF table.

CS610 [NUMBER (5,2)]: CSCI 610 exam score.

Q610_FAC [CHAR (9)]: Staff-ID of faculty, who provides CSCI 610 exam questions which has referential integrity constraint with STAFF table.

CS630 [NUMBER (5,2)]: CSCI 630 exam score.

Q630_FAC [CHAR (9)]: Staff-ID of faculty, who provides CSCI 630 exam question which has referential integrity constraint with STAFF table.

CS655 [NUMBER (5,2)]: CSCI 655 exam score.

Q655_FAC [CHAR (9)]: Staff-ID of faculty, who provides CSCI 655 exam questions which has referential integrity constraint with STAFF table.

CS660 [NUMBER (5,2)]: CSCI 660 exam score.
Q660_FAC [CHAR (9)]: Staff-ID of faculty, who provides CSCI 660 exam questions which has referential integrity constraint with STAFF table

ACTIONS [VARCHAR2 (2000)]: Note if not-pass in exams (fail)

30. COMMITTEE. Store student’s advisor or graduate committee members. Each record represents a faculty member for a student. Thus, if a student has one advisor and two committee members, he or she will have a total of three records.

STD_ID [CHAR (9), PK, FK]: Student Identification Number which has referential integrity constraint with GRADSTUDENTS table

STAFF_ID [CHAR (9), PK, FK] Staff-ID of the student’s faculty which has referential integrity constraint with STAFF table

(STD_ID, STAFF_ID) constitutes the primary key of the table

ISADVISOR [CHAR (1)]: Y if the faculty is the advisor, blank or N if the faculty is a committee member.

30. COMPEXAMFACULTY. Stores all information for the comprehensive exam

QTRYR [VARCHAR2(5)]: Quarter / year of the comprehensive exam

COURSE [VARCHAR2(15)]: Name of the core course

FACULTY [CHAR(9)]: Faculty name who made the exam questions

COMPEXAM_ID [NUMBER(5)]: Identification number of the comprehensive exam

31. DISMISS. Tracks a student’s dismissal history

STD_ID [VARCHAR2(10)]: Identification number of the student

DISMISS_TERM [VARCHAR2(5)]: Quarter / year when the student is dismissed

DISMISS_REASON [VARCHAR2(1000)]: Reason for dismissal

32. EXAM. Store graduate student’s exam associated information.

EXAM_IDSEQ [VARCHAR2 (10), PK]: Sequence number, generated by selecting SEQUENCE EXAM_SEQ, start with 10000 and increment by 1.

STD_ID [CHAR (9), FK]: Student Identification Number which has referential integrity constraint with GRADSTUDENTS table

EXAM_NAME [DATE]: Exam name – WREE, TOEFL

SCORE [NUMBER (5,2)]: Exam score

EXAM_DATE [DATE]: Exam date
33. FUNCTIONS. Defines all registered functions to assign to roles and to users

FUNCTION_ID [NUMBER(3)]: Identification number of the function
FUNCTION [VARCHAR2(40)]: Name of the function
FUNCTIONPAGE [VARCHAR2(30)]: JSP page of the function
PRIVILEGE_CODE [VARCHAR2(10)]: Execution privilege token of the function

34. GRADSTUDENTS. This is a main table to store student information. Using the relational model most fields are foreign keys with referential integrity constraint with lookup tables. This dramatically reduces the record size and input errors when foreign key fields must satisfy referential integrity constraints.

STD_ID [CHAR (9)]: Student Identification Number
LAST [VARCHAR2 (30)]: Student last name
FIRST [VARCHAR2 (30)]: Student first name
MIDDLE [VARCHAR2 (15)]: Student middle name
DOB [DATE]: Date of birth
GENDER [CHAR (1)]: Gender – F for female, M for male
ETHNICITY_ID [VARCHAR2 (10), FK]: SIS+ ethnicity code which has referential integrity constraint with ETHNICSORGN table
SIS_EMAIL [VARCHAR2 (45)]: E-mail address extracted from SIS+
EMAIL [VARCHAR2 (45)]: E-mail address which input or updated by department or students themselves
SIS_PHONE [VARCHAR2 (13)]: Phone number extracted from SIS+
PHONE [VARCHAR2 (13)]: Phone number which input or updated by department or students themselves
INTNAL [CHAR (1)]: Y if the student is an international student
IS_CA_RES [CHAR (1)]: Y if the student is a California resident
IS_BS_CSUSB [CHAR (1)]: Y if the student obtained a BS degree from CSUSB
IS_CS [CHAR (1)]: Y if the student’s undergraduate major is computer science
NATIONALITY [CHAR (4)]: SIS + country code which has referential integrity constraint on COUNTRIES table
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RES_ADDRS</td>
<td>[VARCHAR2 (100)]: Street address where the student is residing</td>
</tr>
<tr>
<td>CITY</td>
<td>[VARCHAR2 (20)]: City</td>
</tr>
<tr>
<td>STATE</td>
<td>[CHAR (2)]: State abbreviation</td>
</tr>
<tr>
<td>ZIP</td>
<td>[VARCHAR2 (9)]: Zip code</td>
</tr>
<tr>
<td>COUNTRY</td>
<td>[VARCHAR2 (50)]: Country name</td>
</tr>
<tr>
<td>PERM_ADDRS</td>
<td>[VARCHAR2 (200)]: Whole address extracted from SIS+</td>
</tr>
<tr>
<td>ADMIT_TERM</td>
<td>[VARCHAR2 (5)]: Term code which represents admitted quarter and year</td>
</tr>
<tr>
<td>ADMIT_STATUS</td>
<td>[VARCHAR2 (10), FK]: Admission status which has referential integrity constraint with ACADSTANDING table</td>
</tr>
<tr>
<td>ACAD_STAND</td>
<td>[VARCHAR2 (10), FK]: Academic standing status which has referential integrity constraint with ACADSTANDING table</td>
</tr>
<tr>
<td>CUR_STATUS</td>
<td>[VARCHAR2 (10), FK]: Current Status which has referential integrity constraint with CURRENTSTATUS table</td>
</tr>
<tr>
<td>CUR_START</td>
<td>[VARCHAR2 (5)]: Term code which represents the year and quarter the current status started</td>
</tr>
<tr>
<td>CUR_END</td>
<td>[VARCHAR2 (5)]: Term code which represents the year and quarter when current status will end</td>
</tr>
<tr>
<td>TERM_LASTATD</td>
<td>[VARCHAR2 (5)]: Term code which represents the last year and quarter attended</td>
</tr>
<tr>
<td>CLASSIFIED_TERM</td>
<td>[VARCHAR (5)]: Term code which represents classified quarter and year</td>
</tr>
<tr>
<td>CANDIDACY_TERM</td>
<td>[VARCHAR (5)]: Term code which represents the year and quarters that student is advanced to candidacy</td>
</tr>
<tr>
<td>TOEFL_WAIVED</td>
<td>[CHAR (1)]: Y if the TOEFL test requirement has been waived</td>
</tr>
<tr>
<td>GRE_VERB</td>
<td>[NUMBER (5,2)]: GRE verbal score</td>
</tr>
<tr>
<td>GRE_QUANT</td>
<td>[NUMBER (5,2)]: GRE quantitative score</td>
</tr>
<tr>
<td>GRE_ANAL</td>
<td>[NUMBER (5,2)]: GRE analytical score</td>
</tr>
<tr>
<td>GRE_DATE</td>
<td>[DATE]: Date GRE was taken</td>
</tr>
<tr>
<td>MASTER_OPTION</td>
<td>[CHAR (1)]: Type of graduation track option – P for project, T thesis, C for comprehensive exam.</td>
</tr>
</tbody>
</table>
EXIT_ACTION_ID [VARCHAR2 (10)]: SIS+ action code which brought about the termination of this matriculation. This field defaults to the Exit Action defined on the Exit Action Table for either degree or non-degree.

NOTES [VARCHAR2 (1000)]: Notes / comments

35. INQ_STD. Store survey information of inquiring students who wish to pursue a MS in Computer Science at CSUSB.

EMAIL [VARCHAR2 (45)]: E-mail address

FNAME [VARCHAR2 (30)]: First name

LNAME [VARCHAR2 (30)]: Last name

ADDRS [VARCHAR2 (100)]: Street address where the inquiring is residing

CITY [VARCHAR2 (20)]: City name

STATE [VARCHAR2 (2)]: US state abbreviation

ZIP [VARCHAR2 (9)]: Zip code

GENDER [VARCHAR2 (1)]: Gender – F/M for female/male

BS_TYPE [VARCHAR2 (35)]: Baccalaureate degree

GRETAKE [VARCHAR2 (1)]: Y if the inquiring student already took the GRE.

GPAQUALIFY [VARCHAR2 (1)]: Y if the inquiring student’s GPA is greater or equal to 3.0.

MSCSTRACK [VARCHAR2 (20)]: The available tracks that the CS department offers – project, theses, comprehensive exam.

WREETAKE [VARCHAR2 (1)]: Y if the inquiring student has taken WREE (Writing Requirement Exemption Exam).

CUMP_LANG [VARCHAR2 (10)]: The primary programming language that inquiring student learned from his/her introductory programming courses.

INTNAL [VARCHAR2 (1)]: Y if the inquiring student is an international student.

TRANS [VARCHAR2 (1)]: Y if the inquiring student is a transferring student.
QTR_START  [CHAR (1)]: Quarter that inquiring student plans on beginning the program which has referential integrity constraint with QUARTER table.

YR_START  [VARCHAR2 (4)]: Year that inquiring student plans on taking courses.

COURSE_PQ  [VARCHAR2 (2)]: The number of courses that inquiring student plans on taking per quarter.

SURVEYDATE  [DATE]: Survey date

COUNTRY  [VARCHAR2 (50)]: Inquiring student's nationality.

36. _JOBS_. Defines pre-defined job for pre-defined tasks

JOB_ID  [CHAR(5)]: Identification number of the job

JOB  [VARCHAR2(20)]: Name of the job

JOB_SCRIPT  [VARCHAR2(150)]: Description of the job

37. _JOB_CTL_TB_. Stores the system job information in the job scheduler

JOB_SEQ  [VARCHAR2(10)]: Identification number of the system job schedule

RUN_DATE  [DATE]: Date when the system job schedule is to be run

PROC_NAME  [VARCHAR2(50)]: Name of the procedure for running the system job

COMMENTS  [VARCHAR2(100)]: Comments about the system job schedule

39. _LOGINTRACE_. Tracks the user login information

USER_ID  [VARCHAR2(50)]: Identification number of the authorized user

ISEXPIRED  [CHAR(1)]: Flag to track if the user logins for the first-time

CHANGEPWORDDATE  [DATE]: Date when password is changed

LASTLOGIN  [VARCHAR2(20)]: Time of last-time login

40. _MAJOR_. Defines student's undergraduate education major

MAJOR_ID  [VARCHAR2 (4), PK]: Undergraduate education major identification number

MAJOR_NAME  [VARCHAR2 (30)]: Description of major
41. **PREVEDUCATION.** Defines student’s undergraduate education information

- **STD_ID** [CHAR (9), PK]: Student Identification Number which has referential integrity constraint with GRADSTUDENTS table.
- **YEAR** [VARCHAR2 (4)]: Undergraduate year.
- **EDU_LEVEL** [CHAR (1)]: Education Level.
- **SCHOOL** [VARCHAR2 (50)]: Undergraduate school.
- **MAJOR_ID** [VARCHAR2 (4), FK]: Undergraduate education major identification number which has referential integrity constraint with MAJOR table.

42. **PROBATION.** Store student’s probation information.

- **STD_ID** [CHAR (9), PK, FK]: Student Identification Number which has referential integrity constraint with GRADSTUDENTS table.
- **PROB_START_TERM** [VARCHAR2 (5)]: Probation starts term.
- **PROB_END_TERM** [VARCHAR2 (5)]: Probation ends term.
- **PROB_REASON** [VARCHAR2 (500)]: Store the reason for probation.

43. **PROJECT.** Store master project track information.

- **STD_ID** [CHAR (9), PK, FK]: Student Identification Number which has referential integrity constraint with GRADSTUDENTS table.
- **TITLE** [VARCHAR2 (300)]: Project title.
- **ORAL1_DATE** [DATE]: First oral exam date.
- **IS_ORAL1_PASS** [CHAR (1)]: P if pass; F if fails.
- **ORAL2_DATE** [DATE]: Second oral exam date.
- **IS_ORAL2_PASS** [CHAR (1)]: P if pass; F if fails.
- **CMTE_RECOM** [VARCHAR2 (2000)]: Committee member’s recommendation, if fail (F) in Oral exam.
- **PRESENT_DATE** [DATE]: Project presentation date.
- **PJ_RATING** [CHAR (1)]: P for no-modification, M for modification, F for poorly organized (fail) in document.
- **DOC_REV_COMNT** [VARCHAR2 (2000)]: Document revision comment.
44. ROLEFUNCTIONS. Stores functions assigned to each user role

- **ROLE_ID** [CHAR(1)]: Identification number of the role
- **FUNCTION_ID** [NUMBER(3)]: Identification number of the function

45. SIS_CLASS_ATTEND.

46. STAFF. Stores staff information including faculty, staff or other employees, who are involved in the graduate program.

- **STAFF_ID** [CHAR (9), PK]: Staff identification (ID) number
- **TITLE** [VARCHAR2 (20)]: title – Professor, Associate Professor, Chair, President
- **LAST** [VARCHAR2 (20)]: Last name
- **FIRST** [VARCHAR2 (20)]: First name
- **MIDDLE** [VARCHAR2 (15)]: Middle name
- **ISFACULTY** [CHAR (1)]: Y if the staff member is faculty
- **EMAIL** [VARCHAR2 (40)]: E-mail address
- **PHONE** [VARCHAR2 (20)]: Telephone number including area code
- **POSITION** [VARCHAR2 (40)]: MS program Assistant or Graduate Coordinator

47. STD_PREQ. Store evaluation result of students’ required and waived prerequisite courses.

- **STD_ID** [CHAR (9), PK]: Student Identification number, a composite primary key has referential integrity constraint with GRADSTUDENTS table.
- **COURSE_ID** [VARCHAR2 (15), PK]: Course identification (ID) number, a composite primary key has referential integrity constraint with COURSES table.

(STD_ID, COURSE_ID) constitutes the primary key of the table

- **ISWAIVED** [VARCHAR2 (1)]: Y if the prerequisite course is waived by advisor.
- **UPDATE_DATE** [DATE]: Record’s insertion or last updating date.
48. **STD_RECOM.** Store advisor’s recommend courses for student.

- **STD_ID** [CHAR (9), PK, FK]: Student Identification number, a composite primary key has referential integrity constraint with GRADSTUDENTS table.
- **COURSE_ID** [VARCHAR2 (15), PK, FK]: Recommend Course ID, a composite primary key has referential integrity constraint with COURSES table.

(STD_ID, COURSE_ID) constitutes the primary key of the table

**UPDATE_DATE** [DATE]: Recommend course last update date.

49. **STUDENTCOURSES.** Stores courses taken by a student. Each student has as many records as courses he or she has taken.

- **STD_ID** [CHAR (9), PK, FK]: Student Identification Number which has referential integrity constraint with GRADSTUDENTS table.
- **COURSE_ID** [VARCHAR2 (15), PK, FK]: Course identification number which has referential integrity constraint with COURSES table.
- **COURSE_TERM** [VARCHAR2 (5), PK]: Term code which represents the course taken year and quarter.

(STD_ID, COURSE_ID, COURSE_TERM) constitutes the primary key of the table

- **COURSETYPE** [CHAR (1), FK]: Course type code (store core, elective, prerequisite course, recommend only), which has referential integrity constraint with COURSETYPES table.
- **GRADE_ID** [CHAR (3), FK]: Grade for the course taken, has referential integrity constraint with GRADES table.
- **DISCOUNT_GRADE** [CHAR (1)]: Y for discount grade which will not be counted into GPA & course taken.

50. **STUDENTEMAILLOG.** Tracks all information for emails sent to students

- **STD_ID** [VARCHAR2(10)]: Identification number of the student
- **SEND_DATE** [DATE]: Date when email is sent to the student
- **ISSEND** [CHAR(1)]: Flag that indicates if the email has been sent
- **COMMENTS** [VARCHAR2(100)]: Content of the subject field of email
- **TERMCODE** [VARCHAR2(5)]: Quarter when the email is sent
REPORTCODE [VARCHAR2(10)]: Name of the report for which email is to be generated

51. STUDENTMAILS. Stores the contents of the emails sent to students

- STD_ID [VARCHAR2(10)]: Identification number of the student
- TERM [VARCHAR2(5)]: Quarter/year when email is sent to the student
- CATEGORY [VARCHAR2(100)]: Type of the email
- CONTENT [BLOB]: Body of the email

52. TASKSCHEDULE. Defines frequencies for reported tasks

- SCHEDULE_ID [CHAR(1)]: Identification number of the schedule
- SCHEDULE [VARCHAR2(30)]: Name of the schedule
- SCHEDULE_DAYS [NUMBER(3)]: Duration of the schedule

53. TASKSTATUS. Defines status of a task (denied, completed, failed, not yet evaluated, and uncompleted)

- STATUS_ID [CHAR(1)]: Identification number of the status
- STATUS [VARCHAR2(30)]: Description of the status

54. TASKMANUAL. Defines all pre-defined tasks

- TASK_CODE [CHAR(3)]: Identification number of the task
- TASK_ACTION [VARCHAR2(50)]: Brief description of the task
- GUIDE [VARCHAR2(300)]: Guide to complete the task

55. TASKBOOK. Stores jobs in each pre-defined task

- TASK_CODE [CHAR(3)]: Identification number of the task
- JOB_ID [CHAR(5)]: Identification number of the job
- JOB_ORDER [NUMBER(2)]: Sequence number of the job within the task

56. TASKREGSTR. Stores tasks assigned to each student, faculty, or staff member

- TASK_ID [NUMBER(6)]: Identification number of the task
- TASK_CODE [CHAR(3)]: Code of the task
- TASK_OWNER [VARCHAR2(9)]: Identification number of the staff who assigns the task
57. THESES. Store master thesis track information.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD_ID</td>
<td>[CHAR (9), PK, FK]: Student Identification Number which has referential integrity constraint with GRADSTUDENTS table</td>
</tr>
<tr>
<td>TITLE</td>
<td>[VARCHAR2 (300)]: Thesis title</td>
</tr>
<tr>
<td>PRESENT_DATE</td>
<td>[DATE]: Thesis presentation date</td>
</tr>
<tr>
<td>THESIS_RATING</td>
<td>[CHAR (1)]: P for no-modification, M for modification, F for poorly organized (fail) in document</td>
</tr>
<tr>
<td>DOC_REV_COMNT</td>
<td>[VARCHAR2 (2000)]: Document revision comment</td>
</tr>
</tbody>
</table>

58. USERS. Store user account information for faculty, staff, and students.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER_ID</td>
<td>[VARCHAR2 (10), PK]: User identification number</td>
</tr>
<tr>
<td>PWORD</td>
<td>[VARCHAR2 (15)]: Password for authentication</td>
</tr>
<tr>
<td>ROLE_ID</td>
<td>[CHAR (1), FK]: User’s role which has referential integrity constraint with ROLES table.</td>
</tr>
<tr>
<td>STAFF_STD_ID</td>
<td>[CHAR (9)]: Staff identification number of staff or student identification number of Student</td>
</tr>
<tr>
<td>ISSTAFF</td>
<td>[CHAR (1)]: S if the user is a staff member</td>
</tr>
<tr>
<td>FIRSTPAGE</td>
<td>[NUMBER (3)]: User’s customized home page</td>
</tr>
</tbody>
</table>
59. YEARS. Defines year range for an academic year

YEAR [VARCHAR2(15)]: Starting year and ending year for the academic year

A.3 Automation Processing Intermediate tables, process log tables, and Exception Handling Tables

60. COMMITTEE_EXCP. An exception handling log table of MS Access database migration stores failing insertion records on student’s committee information.

STD_ID [CHAR (9)]: Student identification number
STAFF_ID [CHAR (9)]: Staff identification number of staff

61. GRE_EXCP. Exception handling log table to store fail loading records of student GRE exam taken information, which extracted from SIS+ for automation process procedure to insert new record or update existing record in GRADSTUDENTS table.

STD_ID [CHAR (9)]: Student Identification Number
GRE_VERB [NUMBER (5,2)]: GRE verbal score
GRE_QUANT [NUMBER (5,2)]: GRE quantitative score
GRE_ANAL [NUMBER (5,2)]: GRE analytical score
GRE_DATE [DATE]: Date GRE was taken
LOG_DATE [DATE]: Data fail loading

62. LOG_AUTOPROC. Log table which help to keep track automation process time log of stored procedures.

AUTOPROC_ID_SEQ [VARCHAR2 (5)]:
PROCESS_START [TIMESTAMP]: Automation process start time log
PROCESS_END [TIMESTAMP]: Automation process finish time log
PROCESS_NAME [VARCHAR2 (30)]: Automation process name, it may be a stored procedure name, log, or data dump table name

63. LOG_CLASSCHANGE. Log table which help to keep track the changes in classification which update by stored procedure of automation process.

STD_ID [CHAR (9)]: Student Identification Number
CLASSIFICATION_ID [VARCHAR2 (10)]: Indicate the type of classification -- Classified or Adv. to Candidacy.
START_TERM [VARCHAR2 (10)]: Store the term begin with this classification
64. **LOG COURSEGRADE.** Log table which help to keep track any student course taken information change in STUDENTCOURSES table which insert or update by stored procedure of automation process.

- **STD_ID** [CHAR (9)]: Student Identification Number
- **COURSE_ID** [VARCHAR2 (15)]: Course Identification Number
- **TERM** [VARCHAR2 (5)]: Course taken term
- **ACTION** [CHAR (1)]: Insert or update
- **OLD_GRADE** [CHAR (2)]: Old letter grade, if any
- **NEW_GRADE** [CHAR (2)]: New letter grade, if any
- **LOG_DATE** [DATE]: Data modification date

65. **LOG NEWSTD.** Log table which help to keep track any new student record which has been insert into GRADSTUDENTS table by automation process procedure.

- **NEW_STDID** [CHAR (9)]: Student Identification Number
- **LOG_DATE** [DATE]: New student record (data) insertion date

66. **LOG STATUSCHANGE.** Log table which help to keep track any student status change in GRADSTUDENTS table which update by stored procedure of automation process.

- **STD_ID** [CHAR (9)]: Student Identification Number
- **STATUS_TYPE** [VARCHAR2 (10)]: Indicate which type (field) of status has been modified -- ADMIT/ACAD/CUR
- **OLD_STATUS** [VARCHAR2 (10)]: Store old status data value
- **NEW_STATUS** [VARCHAR2 (10)]: Store new status data value
- **LOG_DATE** [DATE]: Data modification date

67. **MASTEROPTION_EXCP.** An exception handling log table stores failed migration records of student’s PROJECT, THESIS information.

- **STD_ID** [CHAR (9)]: Student Identification Number
- **MASTER_OPTION** [VARCHAR2 (3)]: Type of graduation track option – P for project, T thesis, C for comprehensive exam.
- **TITLE** [VARCHAR2 (200)]: Student’s project or thesis title.
- **PRESENT_DATE** [DATE]: Student’s project or thesis presentation date.
COMMENTS [VARCHAR2 (1000)]: Document or presentation comments.

LOG_DATE [DATE]: Data fail loading date

68. NUKelog_ctltb. Store PLSQL automation associated update logs keep days.

LOGNAME [VARCHAR2 (30), PK]: Log name
KEEPDAY [VARCHAR2 (5)]: Days log keeps

69. Prob_excp. An exception handling log table stores failed migration records of student's probation information.

STD_ID [CHAR (9)]: Student Identification Number

PROB_START_TERM [VARCHAR2 (10)]: Probation starts term

PROB_END_TERM [VARCHAR2 (10)]: Probation ends term

LOG_DATE [DATE]: Data fail loading date

70. SIS_extract. Intermediate table stores information extracted from SIS+ for automation process procedure to insert or update existing record in GRADSTUDENTS, STUDENTCOURSES, and EXAM tables.

IDENTIFIER [VARCHAR2 (3)]: SIS+ extracted BIO data segment identifier

STD_ID [VARCHAR2 (9)]: Student Identification Number

COL3 [VARCHAR2 (50)]: Generic column
COL4 [VARCHAR2 (50)]: Generic column
COL5 [VARCHAR2 (50)]: Generic column
COL6 [VARCHAR2 (50)]: Generic column
COL7 [VARCHAR2 (50)]: Generic column
COL8 [VARCHAR2 (50)]: Generic column
COL9 [VARCHAR2 (50)]: Generic column
COL10 [VARCHAR2 (50)]: Generic column
COL11 [VARCHAR2 (50)]: Generic column
COL12 [VARCHAR2 (50)]: Generic column
COL13 [VARCHAR2 (50)]: Generic column
COL14 [VARCHAR2 (50)]: Generic column
<table>
<thead>
<tr>
<th>COL15</th>
<th>[VARCHAR2 (50)]: Generic column</th>
</tr>
</thead>
<tbody>
<tr>
<td>COL16</td>
<td>[VARCHAR2 (10)]: Generic column</td>
</tr>
<tr>
<td>COL17</td>
<td>[VARCHAR2 (10)]: Generic column</td>
</tr>
<tr>
<td>COL18</td>
<td>[VARCHAR2 (10)]: Generic column</td>
</tr>
<tr>
<td>COL19</td>
<td>[VARCHAR2 (10)]: Generic column</td>
</tr>
<tr>
<td>COL20</td>
<td>[VARCHAR2 (10)]: Generic column</td>
</tr>
<tr>
<td>COL21</td>
<td>[VARCHAR2 (10)]: Generic column</td>
</tr>
<tr>
<td>COL22</td>
<td>[VARCHAR2 (10)]: Generic column</td>
</tr>
</tbody>
</table>

71. **STDCOURSES_EXCP.** Exception handling log table stores fail loading records of student course taken information.

<table>
<thead>
<tr>
<th>STD_ID</th>
<th>[CHAR (9)]: Student Identification Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>COURSE_ID</td>
<td>[VARCHAR2 (15)]: Course Identification Number</td>
</tr>
<tr>
<td>COURSE_TERM</td>
<td>[VARCHAR2 (15)]: Course taken term</td>
</tr>
<tr>
<td>GRADE_ID</td>
<td>[VARCHAR2 (15)]: Letter grade</td>
</tr>
<tr>
<td>UNITS</td>
<td>[VARCHAR2 (15)]: Course units</td>
</tr>
<tr>
<td>LOG_DATE</td>
<td>[DATE]: Data fail loading date</td>
</tr>
</tbody>
</table>

72. **STDCOURSES_NOSTORE.** A log table stores student’s non-stored course taken information.

<table>
<thead>
<tr>
<th>STD_ID</th>
<th>[CHAR (9)]: Student Identification Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>COURSE_ID</td>
<td>[VARCHAR2 (15)]: Course Identification Number</td>
</tr>
<tr>
<td>COURSE_TERM</td>
<td>[VARCHAR2 (15)]: Course taken term</td>
</tr>
<tr>
<td>GRADE_ID</td>
<td>[VARCHAR2 (15)]: Letter grade</td>
</tr>
<tr>
<td>UNITS</td>
<td>[VARCHAR2 (15)]: Course units</td>
</tr>
<tr>
<td>LOG_DATE</td>
<td>[DATE]: Data loading date</td>
</tr>
</tbody>
</table>

73. **STDINFO_EXCP.** An exception handling log table stores fail loading records of graduate student information which extracted from SIS+ for automation process procedure to insert new student record or update existing record in GRADSTUDENTS table.

| STD_ID        | [CHAR (9)]: Student Identification Number       |
LAST [VARCHAR2 (30)]: Student last name
FIRST [VARCHAR2 (30)]: Student first name
MIDDLE [VARCHAR2 (15)]: Student middle name
DOB [DATE]: Date of birth
GENDER [CHAR (1)]: Gender – F for female, M for male
ETHNICITY [VARCHAR2 (10)]: Ethnicity code
SIS_EMAIL [VARCHAR2 (30)]: E-mail address
SIS_PHONE [VARCHAR2 (13)]: Phone number
INTNAL [CHAR (1)]: Y if the student is an international student
CA_RES [CHAR (1)]: Y if the student is a California resident
IS_BS_CSUSB [CHAR (1)]: Y if the student obtained a BS degree from CSUSB
PERM_ADDR [VARCHAR2 (200)]: Address extracted from SIS+, includes street address, city, state, zip
COUNTRY [VARCHAR2 (50)]: Country, part of PERM_ADDR
ADMIT_TERM [VARCHAR2 (5)]: Student admits term

74. _STDPREQ_EXCP_. An exception handling log table stores fail insertion of new student prerequisite requirement information.

STD_ID [CHAR (9)]: Student Identification Number
COURSE_ID [VARCHAR2 (15)]: Course Identification Number
LOG_DATE [DATE]: Data fail loading date

75. _TOEFL_EXCP_. An exception handling log table stores fail loading records of international student TOEFL exam taken information, which extracted from SIS+ for automation process procedure to insert new record or update existing record in EXAM table.

STD_ID [CHAR (9)]: Student Identification Number
TOEFL_SCORES [NUMBER (5,2)]: TOEFL exam score
TOEFL_DATE [DATE]: TOEFL exam date
LOG_DATE [DATE]: Data fail loading date
76. **VISIT_LOG.** Store application visitor’s activity information

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VISIT_SEQ</td>
<td>[VARCHAR2 (8)]: A sequence number which generates from VISITLOG_SEQ to distinct each row</td>
</tr>
<tr>
<td>VISIT_DATETIME</td>
<td>[DATE]: Visitor’s visiting date and time</td>
</tr>
<tr>
<td>VISITOR_IP</td>
<td>[VARCHAR (15)]: Visitor’s IP address</td>
</tr>
<tr>
<td>PAGE_NAME</td>
<td>[VARCHAR2 (100)]: Visited page (file) name</td>
</tr>
</tbody>
</table>

77. **USER_LOG.** Store application usage information

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGIN_SEQ</td>
<td>[VARCHAR2 (8)]: A sequence number which generates from USERLOG_SEQ to distinct each row</td>
</tr>
<tr>
<td>USERNAME</td>
<td>[VARCHAR2 (10)]: User’s login username</td>
</tr>
<tr>
<td>LOGIN_DATETIME</td>
<td>[DATE]: User’s login date and time</td>
</tr>
<tr>
<td>LOGIN_IP</td>
<td>[VARCHAR (15)]: user’s IP address</td>
</tr>
<tr>
<td>IS_SUCCESS</td>
<td>[VARCHAR (1)]: A flag to indicate user’s login process whether success</td>
</tr>
</tbody>
</table>
APPENDIX B

APPLICATION PROGRAM FILES
### B.1 Login

<table>
<thead>
<tr>
<th>Function Name</th>
<th>(LOGIN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associate File</td>
<td>login.jsp; loginProcess.jsp;</td>
</tr>
<tr>
<td>Include File</td>
<td>checkSecurePage.jsp; noCache.jsp;</td>
</tr>
</tbody>
</table>

### B.2 Forget Password

<table>
<thead>
<tr>
<th>Function Name</th>
<th>FORGET PASSWORD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associate File</td>
<td>forgetPWD.jsp; forgetPWDProcess.jsp; forgetPWDConfirm.jsp;</td>
</tr>
<tr>
<td>Include File</td>
<td>checkSecurePage.jsp; publicFunction.jsp; /Connections/gradConnectBean.jsp; noCache.jsp;</td>
</tr>
</tbody>
</table>

### B.3 Logout

<table>
<thead>
<tr>
<th>Function Name</th>
<th>LOGOUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associate File</td>
<td>bye.jsp;</td>
</tr>
<tr>
<td>Include File</td>
<td>noCache.jsp; /Connections/gradConnectBean.jsp;</td>
</tr>
</tbody>
</table>
### B.4 Graduate Coordinator/Faculty/Program Assistant/Student: Change Password

<table>
<thead>
<tr>
<th>Function Name</th>
<th>CHANGE PASSWORD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associate File</td>
<td>changeIDPassword.jsp; changeIDPasswordError.htm; changeIDPasswordError2.htm; changeIDPasswordDone.htm;</td>
</tr>
<tr>
<td>Include File</td>
<td>checkSecurePage.jsp; publicFunction.jsp; /Connections/gradConnectBean.jsp; noCache.jsp;</td>
</tr>
</tbody>
</table>

### B.5 Graduate Coordinator/Faculty/Program Assistant/Student: Customize Home Page

<table>
<thead>
<tr>
<th>Function Name</th>
<th>CUSTOMIZE HOME PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associate File</td>
<td>custFirstPage.jsp; custFirstPageAbort.jsp; custFirstPageDone.jsp;</td>
</tr>
<tr>
<td>Include File</td>
<td>checkSecurePage.jsp; publicFunction.jsp; /Connections/gradConnectBean.jsp; noCache.jsp;</td>
</tr>
</tbody>
</table>
### B.6 Graduate Coordinator/Faculty/Program Assistant: View Any Student Information

<table>
<thead>
<tr>
<th>Function Name</th>
<th>VIEW ANY STUDENT INFO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associate File</td>
<td>viewStudentInfo.jsp;</td>
</tr>
<tr>
<td></td>
<td>viewDetail.jsp;</td>
</tr>
<tr>
<td></td>
<td>viewStudentList.jsp;</td>
</tr>
<tr>
<td></td>
<td>viewDetailPersonalI.jsp;</td>
</tr>
<tr>
<td></td>
<td>viewDetailAdmission.jsp;</td>
</tr>
<tr>
<td></td>
<td>viewDetailMenu.jsp;</td>
</tr>
<tr>
<td></td>
<td>viewDetailCourses.jsp;</td>
</tr>
<tr>
<td></td>
<td>viewDetailTOEFL.jsp;</td>
</tr>
<tr>
<td></td>
<td>viewDetailProbation.jsp;</td>
</tr>
<tr>
<td></td>
<td>viewDetailMasterOption.jsp;</td>
</tr>
<tr>
<td></td>
<td>viewDetailGradCheck.jsp;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Include File</th>
<th>checkSecurePage.jsp;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>publicFunction.jsp;</td>
</tr>
<tr>
<td></td>
<td>/Connections/gradConnectBean.jsp;</td>
</tr>
<tr>
<td></td>
<td>noCache.jsp;</td>
</tr>
</tbody>
</table>

### B.7 Graduate Coordinator/Faculty: View Own Advisees

<table>
<thead>
<tr>
<th>Function Name</th>
<th>VIEW OWN ADVISEES ONLY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associate File</td>
<td>viewAdvisees.jsp</td>
</tr>
<tr>
<td></td>
<td>viewAdviseesTopCurrent.jsp</td>
</tr>
<tr>
<td></td>
<td>viewAdviseesBotCurrent.jsp</td>
</tr>
<tr>
<td></td>
<td>viewAdviseesTopGraduated.jsp</td>
</tr>
<tr>
<td></td>
<td>viewAdviseesBotGraduated.jsp</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Include File</th>
<th>checkSecurePage.jsp;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>publicFunction.jsp;</td>
</tr>
<tr>
<td></td>
<td>/Connections/gradConnectBean.jsp;</td>
</tr>
<tr>
<td></td>
<td>noCache.jsp;</td>
</tr>
</tbody>
</table>
### B.8 Graduate Coordinator/Program Assistant: View Any Advisees

<table>
<thead>
<tr>
<th>Function Name</th>
<th>VIEW ANY ADVISEES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associate File</td>
<td>viewAnyAdvisees.jsp</td>
</tr>
<tr>
<td></td>
<td>viewAnyAdviseesTop.jsp</td>
</tr>
<tr>
<td></td>
<td>viewAnyAdviseesBot.jsp</td>
</tr>
<tr>
<td></td>
<td>viewAnyAdviseesCurrentBody.jsp</td>
</tr>
<tr>
<td></td>
<td>viewAnyAdviseesGraduatedBody.jsp</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Include File</th>
<th>checkSecurePage.jsp;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>publicFunction.jsp;</td>
</tr>
<tr>
<td></td>
<td>/Connections/gradConnectBean.jsp;</td>
</tr>
<tr>
<td></td>
<td>noCache.jsp;</td>
</tr>
</tbody>
</table>

### B.9 Graduate Coordinator/Faculty: Communications

<table>
<thead>
<tr>
<th>Function Name</th>
<th>COMMUNICATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associate File</td>
<td>facultyCommunication.jsp</td>
</tr>
<tr>
<td></td>
<td>facultyCommunication1.jsp</td>
</tr>
<tr>
<td></td>
<td>facultyCommunication2.jsp</td>
</tr>
<tr>
<td></td>
<td>facultyCommunicationMail1.jsp</td>
</tr>
<tr>
<td></td>
<td>facultyCommunicationMail2.jsp</td>
</tr>
<tr>
<td></td>
<td>facultyCommunicationMail3.jsp</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Include File</th>
<th>checkSecurePage.jsp;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>publicFunction.jsp;</td>
</tr>
<tr>
<td></td>
<td>/Connections/gradConnectBean.jsp;</td>
</tr>
<tr>
<td></td>
<td>noCache.jsp;</td>
</tr>
</tbody>
</table>
B.10 Graduate Coordinator/Program Assistant: Update Student Information

<table>
<thead>
<tr>
<th>Function Name</th>
<th>UNDATE STUDENT INFO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Include File</td>
<td>checkSecurePage.jsp; publicFunction.jsp; /Connections/gradConnectBean.jsp; noCache.jsp;</td>
</tr>
</tbody>
</table>

B.11 Graduate Coordinator: Academic Year Statistics

<table>
<thead>
<tr>
<th>Function Name</th>
<th>ACADEMIC YEAR STATISTICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associate File</td>
<td>academicStatistics.jsp academicStatistics1.jsp</td>
</tr>
<tr>
<td>Include File</td>
<td>checkSecurePage.jsp; publicFunction.jsp; /Connections/gradConnectBean.jsp; noCache.jsp;</td>
</tr>
</tbody>
</table>
### B.12 Graduate Coordinator: Advising Link

<table>
<thead>
<tr>
<th>Function Name</th>
<th>ADVISING LINK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associate File</td>
<td>advisingPHP.jsp</td>
</tr>
</tbody>
</table>
| Include File           | checkSecurePage.jsp;  
                        | publicFunction.jsp;    
                        | /Connections/gradConnectBean.jsp;  
                        | noCache.jsp;               |

### B.13 Graduate Coordinator: Generate Reports

<table>
<thead>
<tr>
<th>Function Name</th>
<th>GENERATE REPORTS</th>
</tr>
</thead>
</table>
| Associate File         | produceReports.jsp;  
                        | produceReportsProcess.jsp;  
                        | attendStatusReports.jsp;  
                        | candidacyReport.jsp;  
                        | classifiedReport.jsp;  
                        | condClassifiedReport.jsp;  
                        | probCondClassifiedReport.jsp;  
                        | adHocReport.jsp;  
                        | adHocProcess.jsp;  
                        | adHocSaveProcess.jsp;  
                        | reusedAdHocReport.jsp;   |
| Include File           | checkSecurePage.jsp;  
                        | publicFunction.jsp;    
                        | /Connections/gradConnectBean.jsp;  
                        | noCache.jsp;               |

### B.14 Graduate Coordinator: Quarterly Reports

<table>
<thead>
<tr>
<th>Function Name</th>
<th>QUARTERLY REPORTS</th>
</tr>
</thead>
</table>
| Associate File         | quarterlyReports.jsp;  
                        | quarterlyReportsProcess.jsp;   |
| Include File           | checkSecurePage.jsp;  
                        | publicFunction.jsp;  
                        | /Connections/gradConnectBean.jsp;  
                        | noCache.jsp;               |
**B.15 Program Assistant: Comprehensive Exam Information**

<table>
<thead>
<tr>
<th>Function Name</th>
<th>COMP EXAM INFO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associate File</td>
<td>compExamInfo.jsp; compExamInfoProcess.jsp;</td>
</tr>
<tr>
<td>Include File</td>
<td>checkSecurePage.jsp; publicFunction.jsp;</td>
</tr>
<tr>
<td></td>
<td>/Connections/gradConnectBean.jsp; noCache.jsp;</td>
</tr>
</tbody>
</table>

**B.16 Program Assistant: Insert New Faculty**

<table>
<thead>
<tr>
<th>Function Name</th>
<th>INSERT NEW FACULTY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associate File</td>
<td>newFaculty.jsp; insertNewFacultyProcess.jsp;</td>
</tr>
<tr>
<td></td>
<td>newFacultyConfirm.jsp;</td>
</tr>
<tr>
<td>Include File</td>
<td>checkSecurePage.jsp; publicFunction.jsp;</td>
</tr>
<tr>
<td></td>
<td>/Connections/gradConnectBean.jsp; noCache.jsp;</td>
</tr>
</tbody>
</table>

**B.17 Program Assistant: Insert New Staff**

<table>
<thead>
<tr>
<th>Function Name</th>
<th>INSERT NEW STAFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associate File</td>
<td>newStaff.jsp; insertNewStaffProcess.jsp;</td>
</tr>
<tr>
<td></td>
<td>newStaffConfirm.jsp</td>
</tr>
<tr>
<td>Include File</td>
<td>checkSecurePage.jsp; publicFunction.jsp;</td>
</tr>
<tr>
<td></td>
<td>/Connections/gradConnectBean.jsp; noCache.jsp;</td>
</tr>
</tbody>
</table>
## B.18 Program Assistant: Print Letters

<table>
<thead>
<tr>
<th>Function Name</th>
<th>PRINT LETTERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associate File</td>
<td>printLetters.jsp</td>
</tr>
<tr>
<td>Include File</td>
<td>checkSecurePage.jsp; publicFunction.jsp; /Connections/gradConnectBean.jsp; noCache.jsp;</td>
</tr>
</tbody>
</table>

## B.19 Program Assistant: Update/View Staff Information

<table>
<thead>
<tr>
<th>Function Name</th>
<th>UPDATE/VIEW STAFF INFO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associate File</td>
<td>staffInfo.jsp; staffInfoProcess.jsp; staffInfoList.jsp; updateStaffInfoProcess.jsp</td>
</tr>
<tr>
<td>Include File</td>
<td>checkSecurePage.jsp; publicFunction.jsp; /Connections/gradConnectBean.jsp; noCache.jsp;</td>
</tr>
</tbody>
</table>

## B.20 Student: Communications

<table>
<thead>
<tr>
<th>Function Name</th>
<th>COMMUNICATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associate File</td>
<td>studentCommunication.jsp; studentCommunication1.jsp; studentCommunication2.jsp; studentCommunicationMail1.jsp; studentCommunicationMail2.jsp; studentCommunicationMail3.jsp;</td>
</tr>
<tr>
<td>Include File</td>
<td>checkSecurePage.jsp; publicFunction.jsp; /Connections/gradConnectBean.jsp; noCache.jsp;</td>
</tr>
</tbody>
</table>
### B.21 Student: Update Contact Information

<table>
<thead>
<tr>
<th>Function Name</th>
<th>UPDATE CONTACT INFO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associate File</td>
<td>updateContactInfo.jsp; updateContactInfoForm.jsp; updateContactInfoProcess.jsp;</td>
</tr>
<tr>
<td>Include File</td>
<td>checkSecurePage.jsp; publicFunction.jsp; /Connections/gradConnectBean.jsp; noCache.jsp;</td>
</tr>
</tbody>
</table>

### B.22 Student: View Student Information

<table>
<thead>
<tr>
<th>Function Name</th>
<th>VIEW STUDENT INFO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associate File</td>
<td>viewDetailStudent.jsp; viewDetailPersonal2.jsp; viewDetailAdmission2.jsp; viewDetailMenu.htm; viewDetailCourses2.jsp; viewDetailTOEFL2.jsp; viewDetailProbation2.jsp; viewDetailMasterOption2.jsp; viewDetailGradCheck2.jsp;</td>
</tr>
<tr>
<td>Include File</td>
<td>checkSecurePage.jsp; publicFunction.jsp; /Connections/gradConnectBean.jsp; noCache.jsp;</td>
</tr>
</tbody>
</table>
REFERENCES


