2005

Advising module: Graduate application system for the Computer Science Graduate Program

Yen-Hsi Chiang

ADVISING MODULE: GRADUATE APPLICATION SYSTEM FOR THE 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
Yen-Hsi Chiang
December 2005

ADVISING MODULE: GRADUATE APPLICATION SYSTEM FOR THE COMPUTER SCIENCE GRADUATE PROGRAM

A Project

Presented to the

Faculty of

California State University,

San Bernardino

---

by

Yen-Hsi Chiang

December 2005

Approved by:

[redacted]

Dr. Josephine G. Mendoza, Chair, Computer Science

11-15-05
Date

[redacted]

Dr. Arturo I. Concepcion

[redacted]

Dr. Kerstin Voigt

[redacted]

Lorraine Frost, Interim Vice-President for IRT

ABSTRACT

The Advising Module: Graduate Application System is a Web-based application system that provides quality advice on coursework for prospective as well as continuing graduate students. It also serves as an improved tracking system for the graduate coordinator.

The Advising Module includes a wide variety of utilitarian options and system dynamics. It enhances the department's database design to interface with CSUSB Student Information System (SIS+) by automating the information entry and update process. In addition, it develops Oracle PL/SQL-based procedures and database triggers to implement crucial business rules.

This project implements prospective student, continuing student, faculty, graduate coordinator, and application administrator modules appropriate to the different users of the system. Authorized parties may obtain access to status evaluations, master's options, and permitted course waivers, course listings, personal data, various advisement forms, application usage statistics, and automatic data updating process reports. This project is designed to offer useful information and advisory service to all students.

ACKNOWLEDGMENTS

First, I wish to thank my project advisor, Dr. Josephine Mendoza, for her strong and persistent support. Her guidance enabled me to thoroughly and comprehensively complete the analysis, design, and implementation phases of this project.

In addition, I would like to express my gratitude to Dr. Arturo Concepcion and Dr. Kerstin Voigt, my project committee members, who provided insightful comments and suggestions.

Also, I wish to thank Lorraine M. Frost, Interim Vice-President for Information Resources and Technology, whose vision and intellectual insight contributed significantly to the accomplishment of this project.

Moreover, I would like to thank Charles Tabbut, Interim Director for Academic Computing Services, who encouraged me to explore the interface possibilities between CSUSB Student Information System (SIS+) and our departmental Oracle database.

I would further like to thank Lenora Venturina, Analyst/Programmer Lead, who has assisted me with the analysis of SIS+ data that this project required. Also, thank analyst/programmer Susan Walsh who coded the COBOL program that extracted the needed data.

Finally, I wish to thank Sunny Lin, analyst/programmer, for hosting the development of our Oracle server within the ACS network. Without these assistance from these professionals, I most certainly could not have completed this project.

TABLE OF CONTENTS

CHAPTER FIVE: DEPLOYMENT

CHAPTER SIX: CONCLUSIONS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 Background

Currently the Computer Science Department at California State University, San Bernardino, uses a stand-alone Microsoft Access 2000 Database system to manage its graduate students. Although the system has facilitated the work of the Graduate Coordinator and the Program Assistant, it has the following limitations:

1) Cannot be accessed by graduate students to update contact information (address, email, and telephone).
2) Cannot be accessed by faculty advisors to retrieve and update information about their student advisories.
3) Lacks scalability with respect to database as well as system design.
4) Student information already available from CSUSB Student Information System must be manually input again by the Program Assistant.

Thus the need for a system update occurs here - the Web-based Database Management System for the Computer Science Graduate Program (Web-based DBMS). This was undertaken as a Master's Project by Mr. Dung Vu in Spring 2004 under the direction of Dr. Josephine Mendoza. This system addresses limitations 1)-3) of the stand-alone MS Access system. [18]

Limitation four was not addressed by Dung's system; therefore, it is being addressed in this project.

There exists another application that inspired this project: The "Graduate Advisory System Advising Component." This began as a class project in Fall 2003 under the direction of Dr. A. I. Concepcion. This system component worked with Graduate Advisory System (GRADS) to serve as an advising tool for prospective students, continuing students, and the Graduate Coordinator. [15] The revised Advising Component was developed in Summer 2004 under the direction of Dr. Concepcion and Dr. Mendoza. This replaced the class project database system from MySQL to Oracle9i. [5]

Both projects, the Web-Based DBMS and the GRADS Advising Component, serve a functional purpose for students and faculty by providing rich information and constructive technology tools. However, both projects suffer from information update constraints. Therefore, there is an urgent need to speed up the information update process to reduce manual data input. A solution proposed in this project is to interface the current system with CSUSB Student Information System (SIS+).

1.2 Significance

Currently CSUSB does not possess a systematic advising tool to help prospective and current graduate students effectively plan their graduate programs. All needed information is buried within five hundred pages of the school catalog. This limitation on information access especially frustrates out-of-state or foreign prospective students, since they cannot be physically present on campus to be counseled by a Program Advisor.

By providing the 24/7 advising service through the Web, this new processing module dramatically enhances information access for both the prospective and current students. Also, we increase the likelihood that both will make informed and meaningful decisions relative to overall degree requirements.

Besides these positive effects on improved information access, this new processing module also accelerates the data retrieval process. Furthermore, it reduces the need for paper storage for the Graduate coordinator by providing online versions of these forms: Preliminary Advancement to Candidacy, Advancement to

Candidacy, and the Graduate Advising Sheet. By providing the 24/7 advising service, all CSCI graduate students will greatly benefit from the valuable features that this new processing module offers.

1.3 Purpose

The Advising Module: Graduate Application System combines PHP: Hypertext Preprocessor with enterprise-class Oracle technologies within the Web application. The general purpose is to counsel both prospective and continuing graduate students with respect to course requirements. This module will also improve the ability of the Graduate Coordinator to monitor student progress, and to make modifications as needed. The variety of services this application offers includes:

- *Provide advising on the prerequisite requirement and required courses for prospective students.* The prospective student component will evaluate and recommend the prerequisite and required courses based on survey information and prior academic backgrounds.
- *Provide a study plan and status information for continuing students.* To serve continuing students, the continuing student component will

provide a study plan and allow students to check their grades and academic standing status on the Web.

- *Provide student information and statistics for faculty.* For the department faculty, the faculty component will provide an interface to enable viewing a student's overall performance, Master's option, project/thesis topic, and committee information.
- *Provide tracking results concerning students' requirements for the graduate coordinator.* For the graduate coordinator, the advisor component will provide an interface to evaluate and update students' academic status as well as prerequisite requirements and course recommendations. The Advisor component also provides the graduate coordinator online versions of these forms: Preliminary Advancement to Candidacy, Advancement to Candidacy, and Graduate Advising Sheet. This accelerates the data retrieval process, and eases the need for paper storage involving forms and documents.

- *Provide statistic usage reports and handy navigation tools for database administrators and developers.* For the database administrator, the DBA component will provide an interface to monitor application usage and the validation status of application-specific triggers, functions, and procedures. The DBA component also provides the developer a convenient tool for navigating the database structures in real time.

1.4 Organization of the Project Document

There are six chapters in this document: (1) Introduction, (2) Software Requirements Specifications, (3) Database Design and Implementation, (4) Application Implementation, (5) System Deployment and Maintenance, and (6) Conclusion. The appendices provide technical details on design, implementation, and methods of validation used within this system.

CHAPTER TWO

SOFTWARE REQUIREMENTS SPECIFICATION

2.1 Project Scope

This project will involve three main phases: (1) database development; (2) software development; and (3) system deployment.

The database development framework will be built upon three cornerstones. First, a focus on analysis and modification of the current database design to interface with CSUSB Student Information System (SIS+). Second, program the automation of data conversion and update between the department database and SIS+. Third, design and implement a plan to migrate the departmental MS Access database to Oracle database.

The software development framework will mainly focus on revising the GRADS Advising Component to implement a revised database design.

The system deployment framework will mainly focus on configuring Oracle production server to host department database. Also, this will include building an application production server to deliver the GRADS Advising Module to provide services for students, faculty, and graduate coordinator.

2.2 Product Perspective

2.2.1 Use-Case Diagram

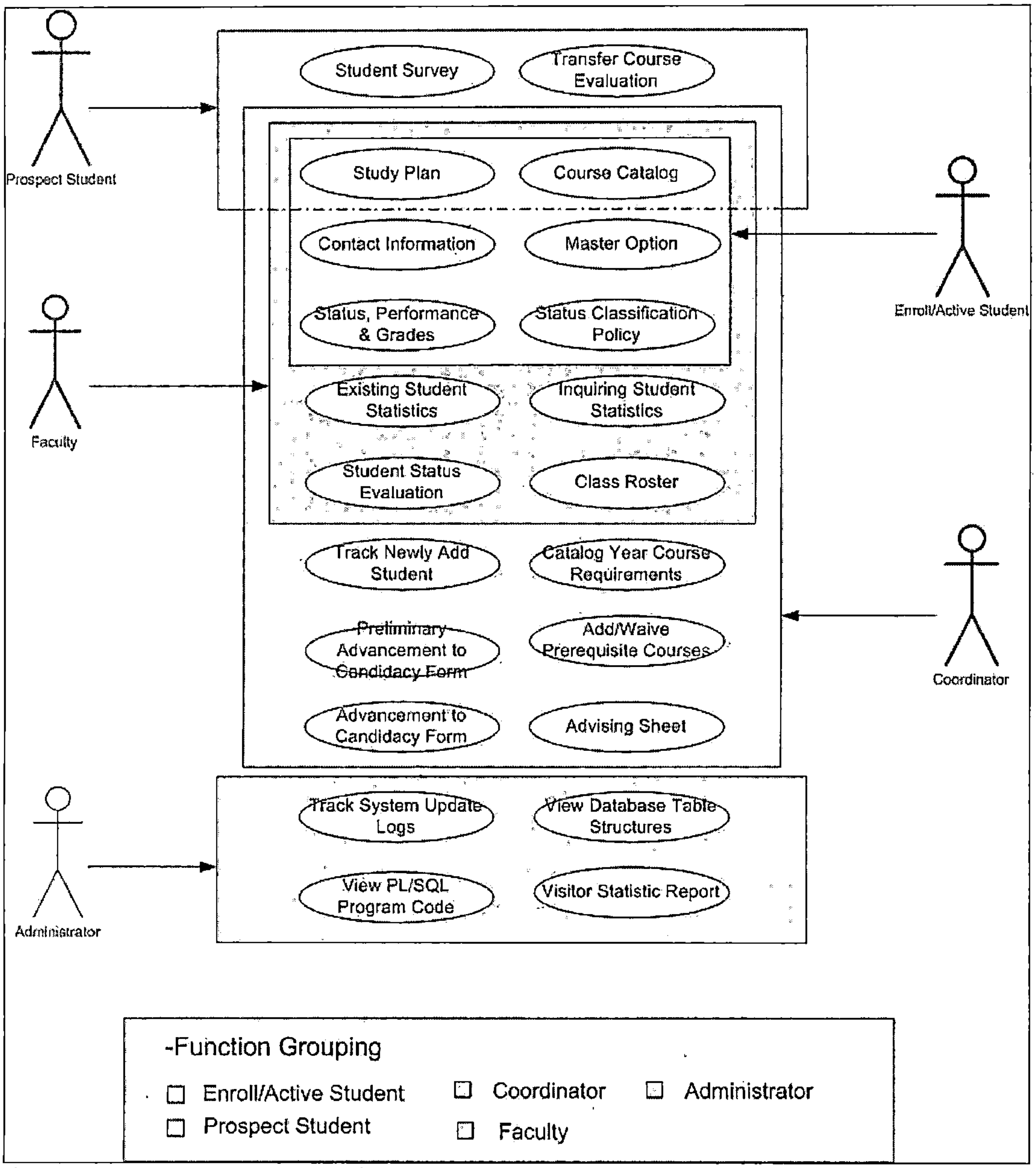


Figure 1. Use-Case Diagram for GRADS Advising Module

2.2.2 System Interfaces

First, users send requests to web server via LAN or Internet using Web browsers. After receiving requests, the web server initials a series OCI calls to the Oracle database server. The Oracle database server will process all DML statements and reply results to correspond sessions on the Web server. The Web server uses the DML results forming answers in proper HTML format and return to user whom commenced requests.

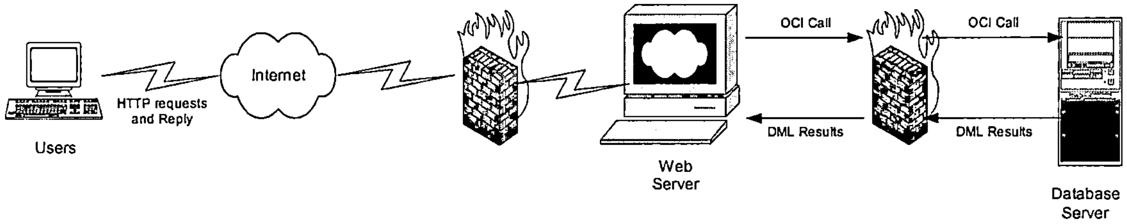


Figure 2. Network Design of GRADS Advising System

2.3 Users

For this Advising Module, the users are classified into five types:

a. *Prospective students*: Students who want to apply to the MSCS Program at CSUSB or have not decided to apply at CSUSB or not, or who are deciding whether to apply to CSUSB, or who want to know more about our program.

b. *Continuing Students*: Students who have been admitted to the MSCS program and are now seeking advice about what courses to take.

c. *Faculty*: A faculty member of the Computer Science Department.

d. *Coordinator:* The faculty member who has been designated as Graduate Coordinator for advising.

e. *Database Administrator/Developer:* IT personnel in charge of the database administration and development tasks.

2.4 Product Functions

All user interfaces will be dynamic. They will either allow the user to enter data into the database, or retrieve data from the database for viewing. A continuing student, faculty, coordinator, or database administrator/developer will be required to enter a Login ID and Password created by the Web-based Database Management System.

There are four roles allowed to login into this system:

Table 1. Roles and Privileges

Role	Privileges
Continuing Students	▪ View personal academic progress
Department Faculty	▪ View students' academic performance
Graduate Coordinator	▪ View and Update students' information and requirements
Database Administrator/ Developer	▪ View database structures and stored programs

Based on the role privileges associated with each login, the user will be taken to the corresponding role interface after a successful login: A Prospective Student Component, Existing Student Component, Coordinator/Faculty Components, or Database Administrator/Developer Component.

2.4.1 Prospective Student Component

This component may be broken down into several web pages for manageability, but the survey will ask the user for the following information:

- Bachelor's Degree type earned (Computer Science, Computer Information Science, Software Engineering, Computer Engineering, Other).
- Primary programming language(s) the student took as introductory programming course(s) (i.e., C, FORTRAN, COBOL, C++, or Java).

- Degree program option the student is interested in taking at CSUSB (i.e., thesis, project, or cumulative test).
- Number of classes a quarter the student desires to take.
- Has GRE been taken?
- Has Graduate Writing requirement been satisfied?

Once the prospective student has filled out the survey, he/she will be taken to the next survey page. This survey page will ask him/her questions regarding courses taken at another university and wish to be transferred into the CSUSB MSCS program. Once the prospective student submits the survey, his/her information will be saved into the departmental database for future recruitment analysis, and she/he will be provided with advising based on the previous answer on surveys.

2.4.2 Continuing Student Component

This component will provide a current student a number of functional links to access the system:

- Obtain a study plan
- View academic status, performance, and grades
- View Course Catalog using keyword search

- View personal contact information and send request for information update
- View master option (thesis, project, or Comprehended Exam) and send request for information update.

2.4.3 Faculty/Coordinator Component

This component will provide a faculty member and the Graduate Coordinator the following functional links to access information about continuing students as well as these features of the system.

- View a student's Study Plan
- View a student's contact information
- View a student's academic status, performance, and grades
- View statistical reports based on gender, country, and ethnicity of students

The coordinator options will include all the function above plus the following:

- Add/Waive Prerequisite courses for continuing students
- Generate Advising Sheet for continuing student
- Generate Student Preliminary Advancement to Candidacy Form

- Generate Student's Advancement to Candidacy Form
- Evaluate and edit a student's status
- Input/Edit Catalog Year Core/Prerequisite Course Requirements

An important feature for the coordinator concerns the notes for graduate advising. The Faculty/Advisor Component will allow the coordinator to enter advising information into the system where it will be maintained in the department's database.

2.4.4 Database Administrator/Developer Component

This component will provide the database administrator/developer a number of options to:

- List database tables, columns, associated comments and constraints
- List stored program units and associated valid status
- View source codes of stored program units
- View log tables
- Configure processing parameters
- Generate ad-hoc application usage reports

CHAPTER THREE

DATABASE DESIGN AND IMPLEMENTATION

This chapter discusses two important phases in database design requirements relating to data collection and analysis as well as to database implementation.

The following activities within database analysis and implementation include:

- Identifying attributes that would be obtained from the SIS+.
- Verifying which available SIS+ data are consistent with information needs, and which need further conversion.
- Deciding which data elements must be migrated from source databases to the target database.
- Programming the automatic data conversion/upload process between SIS+ extraction files and the CSCI department Oracle database.
- Verifying existing PL/SQL programs as well as identifying the need for new stored programs. This will enable the handling of business policies due to modifications within current database objects.

- Design a migration plan to migrate the department's MS Access stand-alone database to the new Oracle database.

3.1 Data Migration Mythology and Design Challenge

Data Migration is a broad term, used differently by different people. I use the term "data migration" here to refer to the entire process of moving data from one system (source) to another (target). Data migration will originate from the current legacy system (MS Access database) and from the CSUSB campus main data source system (SIS+).

Based on Koletzke and Dorsey, in their text Oracle Designer Handbook [10], there exist two major types of projects where data migration is necessary:

- Standard legacy system redesign projects in which the legacy system is discarded when the new system is implemented (i.e. OLTP system).
- New system design projects in which the new system accepts data from and/or passes data to another system on an ongoing basis (i.e. data warehouse system).

In this project, the data migration activity of the departmental Access database can be categorized as the data migration of a standard legacy system redesign. On the other hand, the data migration process from SIS+ to our new system is more similar to building a data warehouse system. Therefore, this operation falls within the second category.

Different types of migration may require different migrating tools and strategies, but they share the same migration phases and tasks. Included here is a list of migration phases and tasks I have covered within this project:

Table 2. Migration Phases

Phrase	Deliverable	Major Tasks Involved
Migration-Strategy	Data Migration Scope & Strategy Document	▪ Determining the specific focus on the overall project. ▪ Identifying the number of source systems & interfaces. ▪ Analyzing which actual data should migrate.
Migration-Pre-Analysis	Migration Plan, Test Plan	▪ Perform legacy data analysis and additional comprehensive data cleansing research. ▪ Generate a preliminary, detailed migration plan. ▪ Develop a migration Test Plan to lay out the measures that will be used to ensure the migration is successful. ▪ Transport the legacy data from its current environment into an Oracle staging area (a database account which contains tables that

Phrase	Deliverable	Major Tasks Involved
		essentially replicate the data structures of the legacy system). ▪ Change the data types to accommodate different database management systems. ▪ Begin the data elements selection process (proceeding through the list of data elements from each & every source data structure, and deciding whether or not each one must migrate.
Migration-Analysis	Identify fields to be mapped	▪ Make a checklist of the legacy data elements which we know must migrate. ▪ Incorporate each data element identified as a candidate for migration into the emerging data model of the new system.
Migration-Design	Data mappings and Conversion code	▪ Map the legacy data elements into tables and columns. ▪ Code migration scripts using PL/SQL.
Migration-Build, Test & Revise	Debugging the conversion code	▪ Execute the code components utilizing the mapping process. ▪ Answer the questions: How many records did we expect this script to create? Has the data been loaded into the correct fields? Has the data been formatted correctly? ▪ Identify other historical data elements to be migrated that were not apparent during the Analysis/Design phases. ▪ Adjust the data model as well as the migration scripts.

Phrase	Deliverable	Major Tasks Involved
Migration-Implementation	Production-ready data	▪ Validate all of the mappings and successfully implement a series of scripts that have been thoroughly tested. ▪ Move the debugged code from the development system to the production system. ▪ Load data to the new system with the goal of being production-ready.
Migration-Maintenance		▪ Schedule automatic process job to reload and refresh the new system in timely intervals. ▪ Validate scripts.

The most challenging parts of the design phase I encountered center around the fact that data migration occurs from multiple sources and converge into one source, and the fact that the specific data definitions and structures of the older and newer systems differ. These differences create the need for intensive analysis tasks to identify the pros and cons relative to decisions on mapping, conversion, and transformation strategy.

Numerous SIS+ data definitions and translated data values have been incorporated into the new system within the final decision process. These changes necessitate an additional intensive revision process regarding underlying queries and codes within the application. As Koletzke states about the data migration process [Koletzke 9], "This is a task that is usually greatly underestimated, if not completely ignored. Usually, the scope and difficulty of data migration is impossible to assess until the migration itself is almost complete."

3.2 Student Information System Data Extraction

SIS+ data extraction covers 13 data definitions, 38 attributes per student, and a total of 8822 rows of data. The data can be grouped as follows:

- Data definitions
- Students' Bio/Demographic Data
- Students' terms of admittance and classified status for each term enrolled
- GRE and TOEFL exams info
- Courses taken and grade history
- Course enrollment/registration info

The table below includes a variety of relevant extracted data items.

Table 3. Student Information System Extraction Layout and Description

Extraction Layout	Description
BIO-010-SEGMENT	▪ Extraction Segment identifier
BS-SID	▪ Student Identification Number
BS-FIRST	▪ First name

Extraction Layout	Description
BS-MIDDLE	▪ Middle name
BS-LAST	▪ Last name
BS-BIRTH-DATE	▪ Date of birth
BS-GENDER	▪ Gender
BS-ETHNIC	▪ The student's ethnic origin.
BS-EMAIL	▪ Email Address
BS-PHONE	▪ Contact phone number
BS-STREET	▪ Street Address
BS-CITY	▪ City
BS-STATE	▪ State
BS-ZIP	▪ Zip code
BS-COUNTRY	▪ Country Name
BS-INTERNATIONAL	▪ Y, if is an international student
BS-CA-RES	▪ Y, if is a California resident
BS-CSUSB-BS-DEG	▪ Y, if obtained a BS degree from CSUSB
ENTRY-TERM	▪ SIS+ term code ▪ Represent student's admit quarter and year
BS-EXIT-TERM	▪ SIS+ term code ▪ Represent the expecting graduate quarter and year for this student
BS-EXIT-ACTION	▪ SIS+ action code ▪ Represent the action which brought about the termination of this matriculation.
BS-CIT-CNTRY	▪ Student's citizenship country code
BIO-020-SEGMENT	▪ Extraction Segment identifier
TS-SID	▪ Student Identification Number
TS-TEST	▪ Test/Exam name (TOEFL/GRE)
TS-DATE	▪ Test/Exam taken date
TS-SCORE-1	▪ Test/Exam score for TOEFL, GRE VERBAL score
TS-SCORE-2	▪ GRE Quantitative score

Extraction Layout	Description
TS-SCORE-3	▪ GRE Analytical score
BIO-030-SEGMENT	▪ Extraction Segment identifier
TM-SID	▪ Student Identification Number
TM-TERM	▪ SIS+ term code to represent the quarter and year for this student
TM-CLASS	▪ Student Identification Number
TM-ACAD-ACTION	▪ Values indicating either the type of program in which the student is enrolled or the class year of the student.
TM-ATTEND-STAT	▪ A flag indicating attendance
BIO-040-SEGMENT	▪ Extraction Segment identifier
CS-SID	▪ Student Identification Number
CS-TERM	▪ SIS+ term code ▪ Represent the course taken quarter and year
CS-COURSE	▪ Course ID and Session ID
CS-GRADE	▪ The grade assigned to the student for this course.
CS-UNITS	▪ Course units
GRADE-RULE	▪ 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-TYPE	▪ 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.
COURSE-STATUS	▪ The values reflect the student's status in the section. ▪ Active requests, attempts, permits, authorizations and drops are included.

3.3 Microsoft Database Migration

3.3.1 Migration Tool

Migrating from one database to another is often a high-risk, expensive, and time-consuming process. However, in this project I selected the Oracle Migration Workbench as a migration tool. This reduces the time and risks involved in migrating the MS Access database to the Oracle platform.

The Oracle Migration Workbench is an intuitive, wizard-driven tool. It consists of two primary components: *Migration Workbench* and *Workbench Plug-ins (see Figure 3). These components* work together to perform migration in three major steps: capturing the source database, creating the Oracle Model, and migrating the source database. Figure 4 and Figure 5 below show how to use the Oracle Workbench Capture Wizard to capture the source database into the Workbench Repository. Figure 6 shows how to use the Oracle Workbench Migration Wizard to migrate the source database to the destination Oracle database intermediate staging area.

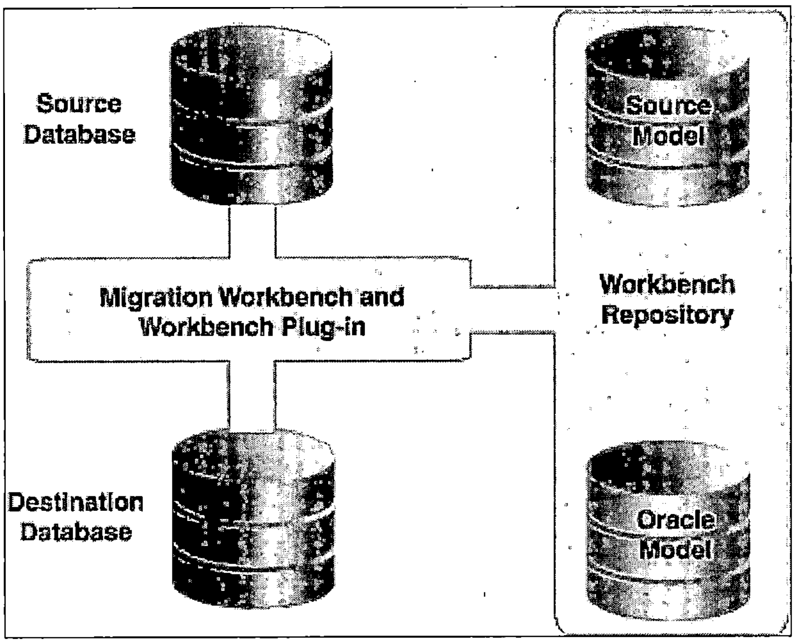
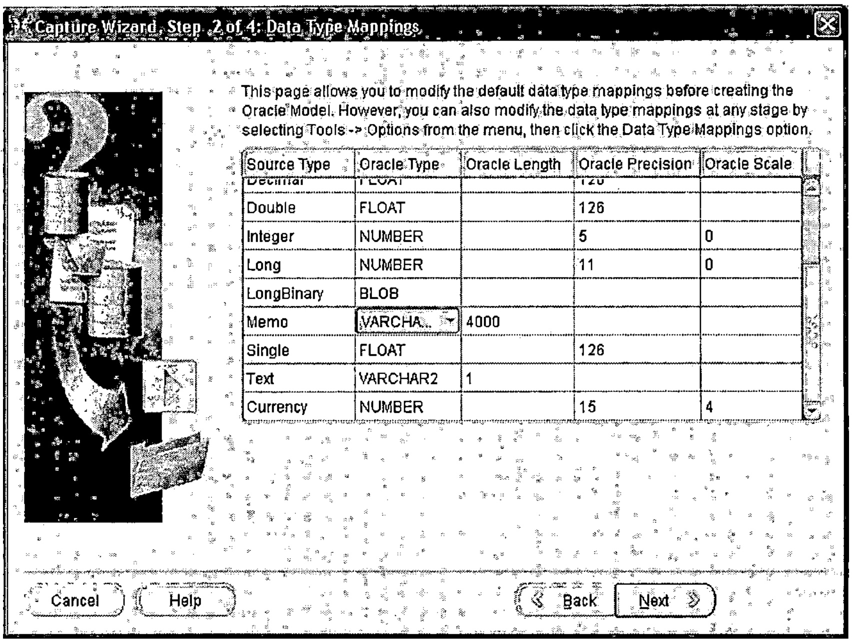


Figure 3. Oracle Migration Workbench



This page allows you to modify the default data type mappings before creating the Oracle Model. However, you can also modify the data type mappings at any stage by selecting Tools -> Options from the menu, then click the Data Type Mappings option.

Source Type	Oracle Type	Oracle Length	Oracle Precision	Oracle Scale
Decimal	FLOAT		126	
Double	FLOAT		126	
Integer	NUMBER		5	0
Long	NUMBER		11	0
LongBinary	BLOB			
Memo	VARCHA..	4000		
Single	FLOAT		126	
Text	VARCHAR2	1		
Currency	NUMBER		15	4

Figure 4. Capture Wizard: Data Type Mappings

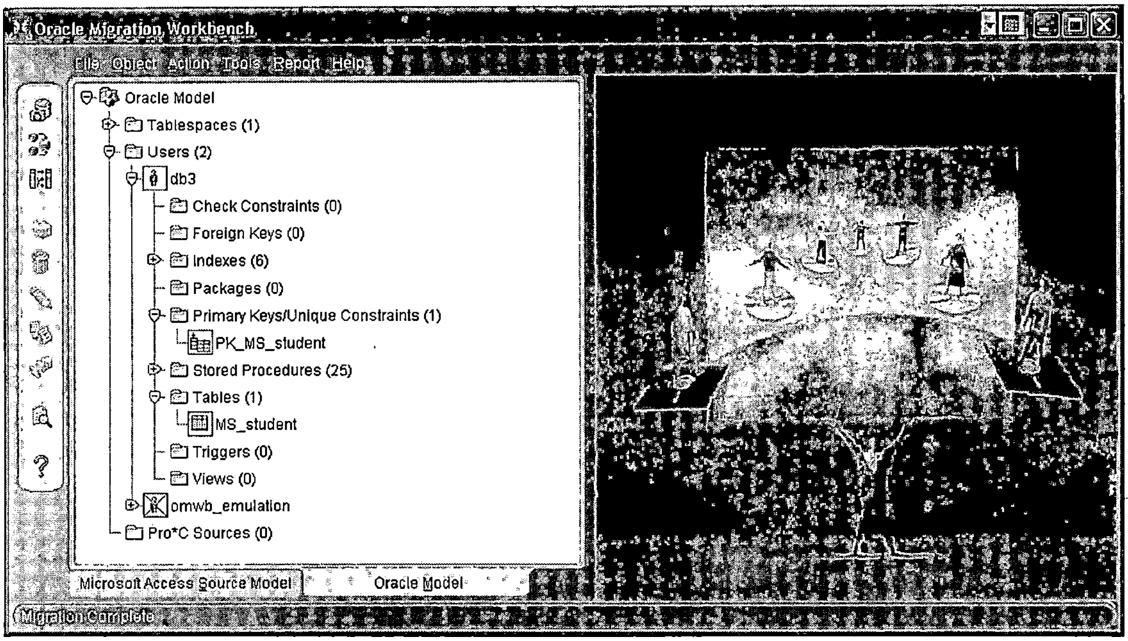
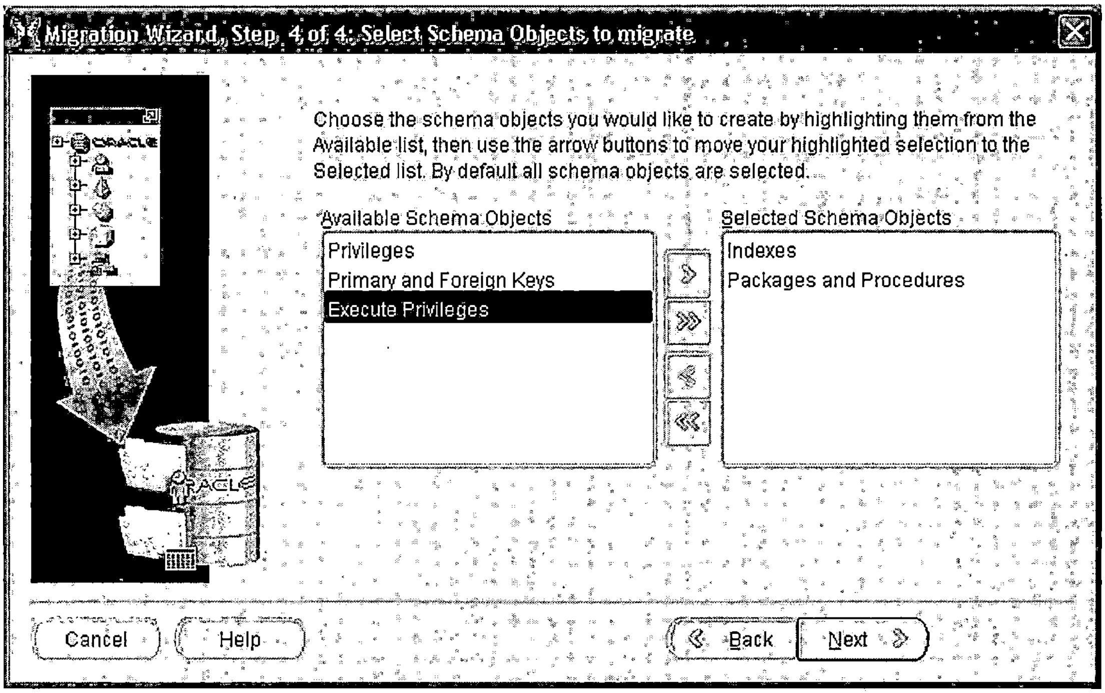


Figure 5. Workbench Repository

The advantage of this tool is that it allows a developer to modify the Source Model and generate an "equivalent" Oracle Model. Nevertheless, the target database structures are usually very different from the source database. The drawback is that these require the developer to code the migration script in PL/SQL to complete the rest of the process. However, it is still a useful tool which can cut developer time in half and greatly reduce the risks involved in the initiating stage.



Migration Wizard, Step 4 of 4: Select Schema Objects to migrate

Choose the schema objects you would like to create by highlighting them from the Available list, then use the arrow buttons to move your highlighted selection to the Selected list. By default all schema objects are selected.

Available Schema Objects
- Privileges
- Primary and Foreign Keys
- Execute Privileges

Selected Schema Objects
- Indexes
- Packages and Procedures

Figure 6. Migration Wizard

3.3.2 Data Elements Selection and the Mapping of Legacy Data Elements

Table 4. Legacy Data Elements Selection and the Mapping

Source Table & Column	Need Convert?	Target Table & Column
MS_STUDENT.SID		GRADSTUDENTS.STD_ID
MS_STUDENT.LAST		GRADSTUDENTS.LAST
MS_STUDENT.FIRST		GRADSTUDENTS.FIRST
MS_STUDENT.MIDDLE		GRADSTUDENTS.MIDDLE
MS_STUDENT.DOB		GRADSTUDENTS.DOB
MS_STUDENT.GENDER		GRADSTUDENTS.GENDER
MS_STUDENT.EMAIL		GRADSTUDENTS.EMAIL
MS_STUDENT.PHONE		GRADSTUDENTS.PHONE
MS_STUDENT.ADDRESS		GRADSTUDENTS.RES_ADDRS
MS_STUDENT.CITY		GRADSTUDENTS.CITY
MS_STUDENT.STATE		GRADSTUDENTS.STATE

Source Table & Column	Need Convert?	Target Table & Column
MS_STUDENT.ZIP		GRADSTUDENTS.ZIP
MS_STUDENT.PERMANENT_ADDR		GRADSTUDENTS.PERM_ADDRS
MS_STUDENT.COUNTRY		GRADSTUDENTS.COUNTRY
MS_STUDENT.GRE_DATE		GRADSTUDENTS.GRE_DATE
MS_STUDENT.GRE_VERBAL		GRADSTUDENTS.GRE_VERB
MS_STUDENT.GRE_QUANTIY		GRADSTUDENTS.GRE_QUANT
MS_STUDENT.GRE_ANALYTICS		GRADSTUDENTS.GRE_ANAL
MS_STUDENT.TOEFL_WAIVED		GRADSTUDENTS.TOEFL_WAIVED
MS_STUDENT.QTR_YEARADMITTED	Yes	GRADSTUDENTS.ADMIT_TERM
MS_STUDENT.ADMIT_STATUS		GRADSTUDENTS.ADMIT_STATUS
MS_STUDENT.CURRENT_STATUS		GRADSTUDENTS.CUR_STATUS
MS_STUDENT.ACADEMIC_STANDING		GRADSTUDENTS.ACAD_STAND
MS_STUDENT.QTR_YEARCLASSIFIED	Yes	GRADSTUDENTS.CLASSIFIED_TERM
MS_STUDENT.QTR_YEAR_CANDIDACY	Yes	GRADSTUDENTS.CANDIDACY_TERM
MS_STUDENT.LASTQTRATTENDE	Yes	GRADSTUDENTS.TERM_LASTATD
MS_STUDENT.PROJECT_OR_THESIS	Yes	GRADSTUDENTS.MASTER_OPTION
MS_STUDENT.PREREQCOMMENTS || ‘ ’ || MS_STUDENT.TESTCOMMENTS		GRADSTUDENTS.NOTES
MS_STUDENT.SID		PROBATION.STD_ID
MS_STUDENT.QTR_YEARPROBATIONENTER	Yes	PROBATION.PROB_START_TERM
S_STUDENT.QTR_YEARPROBATIONREMOVED	Yes	PROBATION.PROB_END_TERM
MS_STUDENT.PROBATIONREASON		PROBATION.PROB_REASON
	EXAM_SEQ.NEXTVAL	EXAM.EXAM_IDSEQ
MS_STUDENT.SID		EXAM.STD_ID
	‘TOEFL’	EXAM.EXAM_NAME
MS_STUDENT.TOEFL_SCORES		EXAM.SCORE
MS_STUDENT.TOEFL_DATE		EXAM.EXAM_DATE

Source Table & Column	Need Convert?	Target Table & Column
MS_STUDENT.SID		PROJECT.STD_ID or THESIS.STD_ID
MS_STUDENT.TITLE		PROJECT.TITLE or THESIS.TITLE
MS_STUDENT.GRADPRESENTATIONDATE		PROJECT.ORAL1_DATE
MS_STUDENT.ORALEXAM	Yes	PROJECT.IS_ORAL1_PASS
MS_STUDENT.COMMENTS		PROJECT.CMTE_RECOM or THESIS.DOC_REV_COMNT
MS_STUDENT.GRADPRESENTATIONDATE		PROJECT.PRESENT_DATE or THESIS.PRESENT_DATE
MS_STUDENT.PROJECTPASS		PROJECT.PJ_RATING
MS_STUDENT.SID		COMMITTEE.STD_ID
MS_STUDENT. ADVISOR	Yes	COMMITTEE.STAFF_ID
MS_STUDENT. ADVISOROTHER	Yes	
MS_STUDENT. OTHERMEMBER1	Yes	
MS_STUDENT. OTHERMEMBER2	Yes	
MS_STUDENT.COMMITEE_MEMBER_01	Yes	
MS_STUDENT.COMMITEE_MEMBER_02	Yes	
MS_STUDENT.COMMITEE_MEMBER_03	Yes	
	Y/N	COMMITTEE.ISADVISOR

3.4 Database Stored Program Units

At the heart of Oracle software is PL/SQL-a programming language that provides procedural extensions to Oracle's version of SQL. PL/SQL also serves as the programming language within the Oracle Developer toolset. Its primary strength is in providing a server-side, stored procedural language that is compatible with SQL, and is robust, portable, and secure. Thus, it offers a platform for high-performing enterprise applications, not only for Fortune 500 companies, but also for Oracle Applications.

This project leverages the strength of Oracle PL/SQL-based stored procedures to migrate SIS+ extraction data and the departmental MS Access database to the Oracle platform. The data migration process includes 13 PL/SQL procedures and four PL/SQL functions to map, translate, and transform the source data to the target database. The Advising Module application employs another seven PL/SQL functions that simplifies the data derivation and presentation.

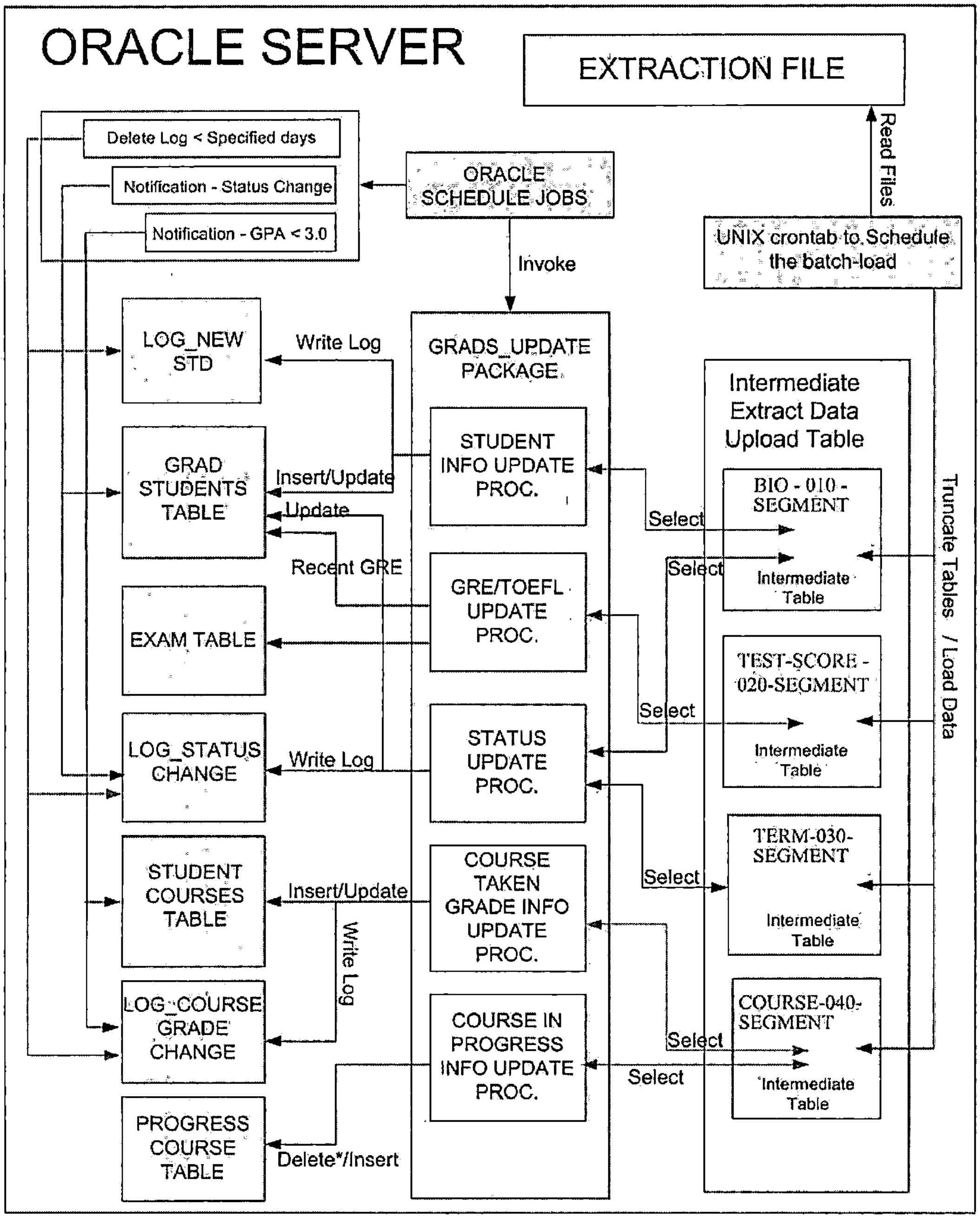
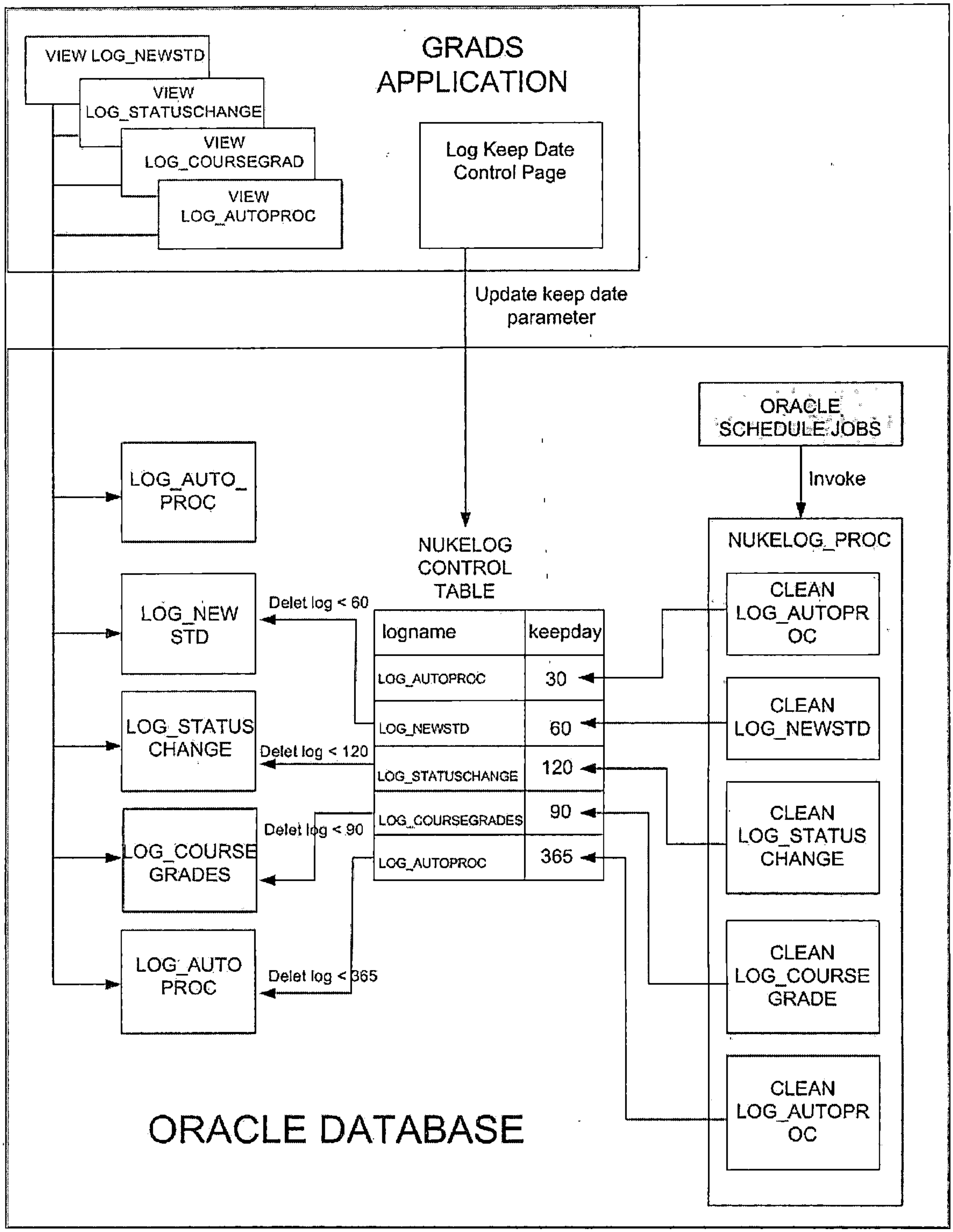


Figure 7. Migrating Procedure Invoking and Execution



logname	keepday
LOG_AUTOPROC	30
LOG_NEWSTD	60
LOG_STATUSCHANGE	120
LOG_COURSEGRADES	90
LOG_AUTOPROC	365

Figure 8. Migration Process Log Cleaning Procedure

3.4.1 Migrating Processes

All migrating processes are implemented by scheduling the UNIX crontab and Oracle jobs to automate the data migration process within the Oracle database server. The UNIX crontab is created to download extraction files to the Oracle server, and then invoke Oracle SQL*Loader to load data to the Oracle database. The Oracle jobs are created to schedule the migrating procedures to map, translate, and transform data to target tables, calculate GPA, and clean the automation process log tables periodically.

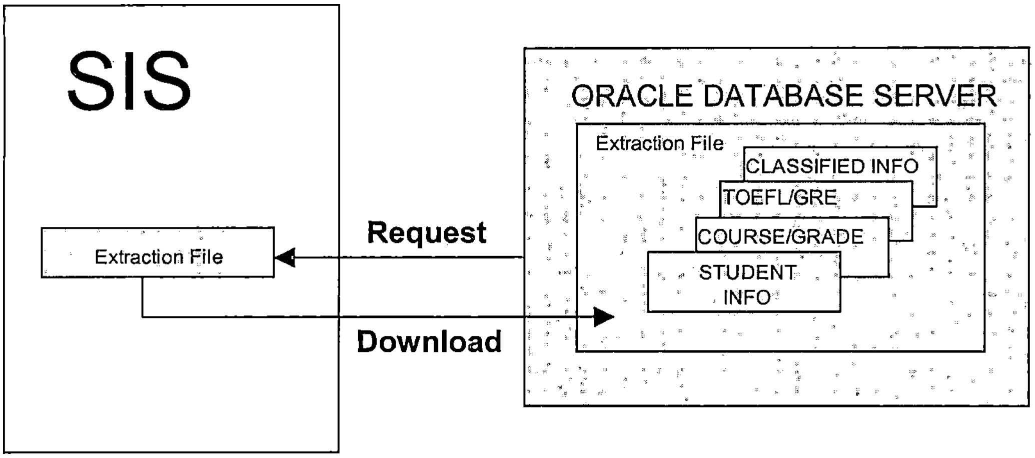
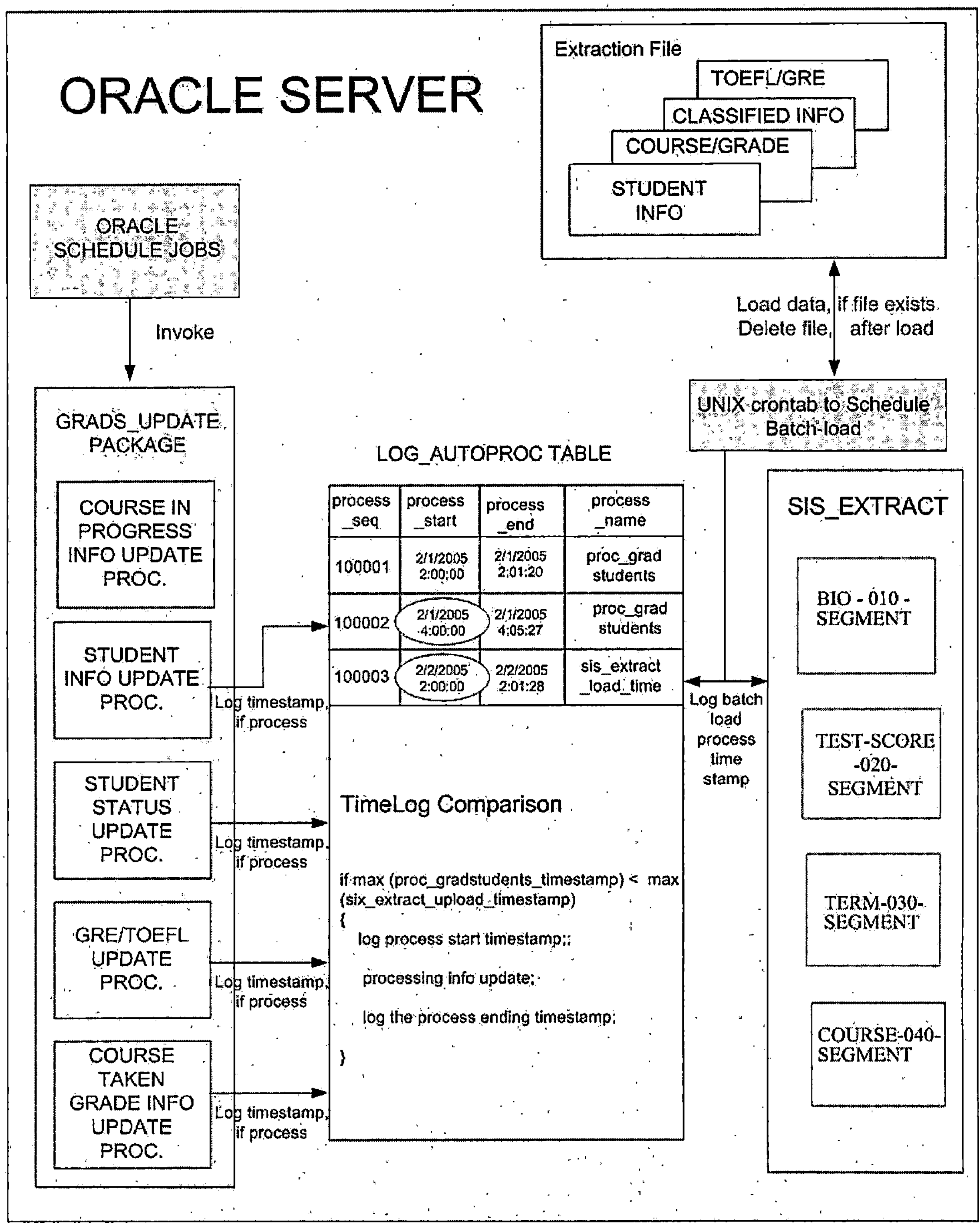


Figure 9. Request for Download the Extraction File


ORACLE SERVER
ORACLE SCHEDULE JOBS
Invoke
GRADS_UPDATE PACKAGE
COURSE IN PROGRESS INFO UPDATE PROC.
STUDENT INFO UPDATE PROC.
STUDENT STATUS UPDATE PROC.
GRE/TOEFL UPDATE PROC.
COURSE TAKEN GRADE INFO UPDATE PROC.
Log timestamp, if process
Log timestamp, if process
Log timestamp, if process
Log timestamp, if process
LOG_AUTOPROC TABLE
Extraction File
TOEFL/GRE
CLASSIFIED INFO
COURSE/GRADE
STUDENT INFO
Load data, if file exists.
Delete file, after load
UNIX crontab to Schedule Batch-load
Log batch load process time stamp
SIS_EXTRACT
BIO - 010 - SEGMENT
TEST-SCORE -020- SEGMENT
TERM-030- SEGMENT
COURSE-040- SEGMENT

process _seq	process _start	process _end	process _name
100001	2/1/2005 2:00:00	2/1/2005 2:01:20	proc_grad students
100002	2/1/2005 4:00:00	2/1/2005 4:05:27	proc_grad students
100003	2/2/2005 2:00:00	2/2/2005 2:01:28	sis_extract _load_time

TimeLog Comparison

```
if max (proc_gradstudents_timestamp) < max
(six_extract_upload_timestamp)
{
  log process start timestamp;;
    processing info update;
    log the process ending timestamp;
}
```

Figure 10. Migration Process Log

3.4.2 Procedural Language/Structured Query Language Program Units

Table 5. Migrating Procedures

Program Name	Purpose	Exception Handling
PROC_MSDB_MIGRATE	▪ Migrates data from MS Access database to GRADSTUDENTS, COMMITTEE, PROJECT, THESIS, COMEXAM, EXAM, and PROBATION tables.	▪ If migration process fails, the failed transaction data will be written into exception handling tables: PROB_EXCP, STDINFO_EXCP, MASTEROPTION_EXCP, TOEFL_EXCP, or COMMITTEE_EXCP.
SIS_MIGRATE	▪ Invokes SIS+ data migration procedures to update students' records in the database.	▪ All execution activities of the called procedures will be recorded in the log table LOG_AUTOPROC
PROC_GRADSTUDENTS	▪ Migrates student's bio/demographical data and admit term from SIS_EXTRACT table to GRADSTUDENTS table ▪ Logs new student's ID to LOG_NEWSTD table	▪ If migration process fails, the failed transaction data will be written into an exception handling table STDINFO_EXCP ▪ Sends an alert mail to database administrator
PROC_STUDENTCOURSES	▪ Migrates student's courses taken and grades history from SIS_EXTRACT table to STUDENTCOURSES table ▪ Records all update transactions in the LOG_COURSEGRADE with old and new data values ▪ Records all non-MSCS concerned courses and grades in	▪ If migration process fails, the failed transaction data will be written into an exception handling table STDCOURSES_EXCP ▪ Sends an alert mail to database administrator

Program Name	Purpose	Exception Handling
	STDCOURSES_NOSTORE table.	
PROC_PROG_COURSES	▪ Transforms student's course enrollment info from SIS_EXTRACT table to COURSE_IN_PROGRESS table ▪ Deletes all and reload fresh data daily	▪ If migration process fails, the failed transaction data will be written into an exception handling table STDCOURSES_EXCP ▪ Sends an alert mail to database administrator
PROC_GRE	▪ Migrates student's latest GRE exam info from SIS_EXTRACT TO GRADSTUDENTS table	▪ If migration process fails, the failed transaction data will be written into an exception handling table GRE_EXCP ▪ Sends an alert mail to database administrator
PROC_TOEFL	▪ Migrates student's TOEFL exam info from SIS_EXTRACT table to EXAM table	▪ If migration process fails, the failed transaction data will be written into an exception handling table TOEFL_EXCP ▪ Sends an alert mail to database administrator
PROC_CLASS_ATTEND	▪ Migrates student's classify/attend status info from SIS_EXTRACT table to SIS_CLASS_ATTEND table	▪ If migration process fails, the failed transaction data will be written into an exception handling table TOEFL_EXCP ▪ Sends an alert mail to database administrator
PROC_CLASSFIED_TERM	▪ Derives the data values for	▪ Sends an alert mail to database

Program Name	Purpose	Exception Handling
	CLASSIFIED & CANDIDACY_TERM of GRADUATESTUDENTS table from SIS_CLASS_ATTEND table	administrator
PROC_NEWSTD_PREQ	▪ Inserts new student's pre-requisite course requirements into STD_PREQ, base on the admit catalog year.	▪ If migration process fails, the failed transaction data will be written into exception handling table PREQ_EXCP ▪ Sends an alert mail to database administrator
PROC_GRADUATE	▪ Checks EXIT-ACTION field from SIS_EXTRACT, and then updates GRADSTUDENTS.ACAD_STAND and GRADSTUDENTS.CUR_STATUS to "G", if student graduates (GD).	▪ Sends an alert mail to database administrator
PROC_GPA	▪ Monitors the daily update list of student's courses take, calls FUNC_GPA to calculate student's GPA, then sends an alert mail to advisor if a student's GPA falls below 3.0	▪ Records student's ID in log table LOG_GPA_BELOW3POINT0 for further tracking, if student's GPA falls below 3.0
FUNC_GPA	▪ Calculates student's GPA	
FUNC_CATALGYR	▪ Converts a SIS+ term code (YYT) into a 4 digit year. Example: 2004-2005 school catalog includes 044, 051, 052, 053, and will be represented in	

Program Name	Purpose	Exception Handling
	'2004'	
GET_CLASS_ADV_TERM	▪ Derives student's first classify term and advancement to candidacy term from SIS_CLASS_ATTEND table	
PROC_NEWSTD_BS_CSUSB	▪ Sends an alert mail to advisor that the new student has a bachelor degree from CSUSB.	
PROC_NUKELOG	▪ Cleans the log tables, based on the info in NUKELOG_CTL table.	
MS_TERM	▪ Receives a combination of two characters and two digits format code (QQYY) and returns a SIS+ term code (YYT)	
GET_QTRYR	▪ Receives a SIS+ term code (YYT), and then returns the full description of term code in "Quarter YYYY" format.	
FULLTERM	▪ Receives a SIS+ term code (YYT), and then returns a 5 digits format year and quarter (YYYYT)	
SISTERM	▪ Receives a 5 digit format year and quarter (YYYYT), and then returns a 3 digit format SIS+ term code (YYT).	
GET_STD_E_COUNT_UNITS	▪ Counts student's total elective courses taken units (with satisfied letter grade)	

Program Name	Purpose	Exception Handling
GET_TOTAL_COUNT_UNITS	▪ Counts student's total course taken units (with satisfy letter grade).	
FORMAT_PHONE	▪ Receives a 10 digit phone number, and then returns a formatted phone number, e.g. telephone number 9098805000 will be formatted as (909)880-5000.	

3.5 Database Table Structures

Several tables are revised to accommodate the changes in the raw data sources and data definitions. There are more tables which are designed to store new data definitions or collect activity information from automating processes relating to data feeding from SIS+. These revised or newly created tables are summarized in Section 3.4.1 and 3.4.2, and design details are listed in Appendix A.

3.5.1 Revision of Previous Projects Designed Table

tabase Tables

<table>
<tr><th></th><th>ion</th><th>Modification</th></tr>
<tr><td></td><td>able
re
te
lt
ation</td><td>

- Dropped columns: SSN, QTR_ADMIT, YR_ADMIT, QTR_CLASSFIED, YR_CLASSFIED, CUR_QTR_START, CUR_YR_START, CUR_YR_END, QTR_LASTATD, YR_LASTATD, TOEFL_SCORES, TOEFL_DATE, GRE_SUBJ, GRE_SUBJ_DATE, QTR_CANDIDACY, YR_CANDIDACY, PRO_THESIS, TITLE, ORAL_EXAM, PASS, PRESENTATION
- Added columns: ADMIT_TERM, IS_CS, CLASSIFIED_TERM, CANDIDACY_TERM, CUR_START, CUR_END, TERM_LASTATD, SIS_EMAIL, SIS_PHONE, NATIONALITY, EXIT_ACTION_ID

Modified column name
<table>
<tr><td>Old</td><td>New</td></tr>
<tr><td>CSUSB_BS</td><td>IS_BS_CSUSB</td></tr>
<tr><td>CA_RES</td><td>IS_CA_RES</td></tr>
<tr><td>ETHNICITY</td><td>ETHNICITY_ID</td></tr>
<tr><td>PRO_THESIS</td><td>MASTER_OPTION</td></tr>
</table>
</td></tr>
<tr><td></td><td>es
ses
n by a
lent

dent has
many
ords as
rses
en</td><td>

- Replace QUARTER_ID and YEAR columns with one new column COURSE_TERM VARCHAR2 (5) to represent the course taken year and quarter
- Add an additional new column, DISCOUNT_GRADE CHAR (1), to flag this course units and grade whether need to be counted into GPA and course taken
</td></tr>
<tr><td>GRADES</td><td>ines letter grade and correspondi ng numerical</td><td>

- Replaces old system defined data values of GRADE_ID (letter grade) and SCORES (numerical grade) with SIS+ defined letter grade and numerical grade
</td></tr>
</table>

	grade and associated description, e.g. letter grade "A" corresponds to numerical grade "4.0".	▪ Adds an additional new column, GRADEID_DEFINITION VARCHAR2 (30), to store SIS+ Grade ID definition/description.
ETHNICSORGN	▪ Defines ethnic origin of student	▪ Replaces old system defined data values of ETHNICITY_ID and ETHNICITY with SIS+ ETHNICITY code ▪ Adds an additional new column, ETHNIC_DEFINITION VARCHAR2 (250), to store SIS+ Ethnic code associated definition and description
COUNTRIES	▪ Defines country where international student comes from.	▪ Adds an additional new column, DEF_STANDING VARCHAR2 (500), to store long description or definition of country code/id.
COURSES	▪ Defines courses taught or counted in the graduate program including recommended, prerequisite, elective and core courses.	▪ Adds the new column, DESCRIPTION VARCHAR2 (1024), to store course description ▪ Adds the new column, GROUPID CHAR (1), to store course group ID ▪ 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) ▪ Adds the new column, IS_GRADE_STORE CHAR (1), to indicate whether it is a course which grade will be stored in departmental database. ▪ Adds the new column, IS_MS CHAR (1), to indicate whether it is a course which will be

<table>
<tr><td></td><td></td><td>scheduled in study plan.</td></tr>
<tr><td>QUARTER</td><td>▪ Defines quarters for courses taken.
▪ In semester system there are only fall and spring values.</td><td>▪ Changes data definition for QUARTER_ID
<table><tr><td></td><td>Old</td><td>New</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></table></td></tr>
<tr><td>ACADSTANDING</td><td>▪ Define academic standing of students (probation, conditionally classified, classified, or advanced to candidacy)</td><td>▪ Adds an additional new column, DEF_STANDING VARCHAR2 (500), to store long description and definition of academic standing code.</td></tr>
<tr><td>ADMITSTATUS</td><td>▪ Define admission status of students (classified, conditionally classified, probation classified, or probation conditionally classified)</td><td>▪ Adds an additional 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,</td><td>▪ Adds an additional new column, DEF_CUR_STATUS VARCHAR2 (500), to store long description or definition of current status code.</td></tr>
</table>

	On Leave, Dismissed, Never Attended, Academic Probation, Withdraw)	

3.5.2 Newly Created Tables

All new tables are separated into three different tablespaces based on the prediction of growth rates and the frequency of DML activities such as INSERT, UPDATE, and DELETE commands. A high DML activity table is most likely to experience row chaining, row migration, or high fragmentation problems, which degrade performance. Querying on chained or migrated rows requires greater consumption of memory, CPU, and I/O. Separating different energy level table objects into different tablespaces will help the DBA to easily fine tune database performance.

3.5.2.1 Data Tables.

Table 7. Data Tables

Table Name	Description
SIS_EXTRACT	▪ Stores raw data which extracted from SIS+
SIS_CLASS_ATTEND	▪ Stores graduate student's classified status (Conditionally Classified, Classified, Advanced to Candidacy) and attending status (Attending, Withdrawn, Dropped) in each term
EXAM	▪ Stores graduate student's exam associated information, e.g. TOEFL, WREE.
COMEXAM	▪ Stores graduate student's master comprehensive exam option information
PROJECT	▪ Stores graduate student's master project option information
THESIS	▪ Stores graduate student's master thesis option information

3.5.2.2 Lookup Tables.

Table 8. Lookup Tables

Table Name	Description
TERMS	▪ Stores list of 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, e.g. Winter is 1, Spring is 2, Summer is 3, and Fall is 4.
EXAMINFO	▪ Stores information and rules regarding each specific type of exam, such as WREE, TOEFL.
SIS_AA651_EXIT_ACTION	▪ Stores list of SIS+ raw data and associated definition regarding to the action which brought about the termination of this matriculation (e.g. "GD" for "Graduation", "LA" for "Leave of Absence", "DD" for "Disenroll/Drop").
SIS_EXTRACT_LAYOUT	▪ Stores list of record layouts of SIS+ extract segments and data elements.
SIS_RT047_ATTEND_STAT	▪ Stores list of SIS+ raw data and associated definition regarding to student's attendance (e.g. "W" for "Withdrawn Regular", "N" for "Not Attending", and "1" for "Dropped Regular Retained").
SIS_RT080_ACAD_ACTION	▪ Stores list of SIS+ raw data and associated definition regarding to the academic actions (e.g. "D" for "Disqualified", "U" for "Probation", "C" for "Continued on Probation", and "M" for "Dismissal").
SIS_RT14C_CLASSIFICATION	▪ Stores list of SIS+ raw data and associated definition regarding to student's classification (e.g. "51" for "Conditionally Classified", "52" for "Classified", and "53" for "Advanced to Candidacy").
SIS_RT310_GRADE_TYPE	▪ Store list of SIS+ raw data and

	associated definition regarding to the type of grading method selected by the student for this course or assigned by the registrar’s staff to handle special grading considerations. (e.g. “CN” for “Credit/No Credit”, “CE” for “Credit by Exam”, “AU” for “Audit”, and “DC” for “Discount credit on repeats”).
SIS_RT312_GRADE_RULE	▪ Stores list of SIS+ raw data and associated definition regarding to the Grade Rule applicable to this SPE (e.g. “E” for “Extended Education”, “U” for “Undergraduate”, and “G” for “Graduate”). ▪ 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
SIS_RT345_CLASS_ENROLL	▪ Stores list of SIS+ raw data and associated definition reflect the student’s status in the class section. ▪ Active requests, attempts, permits, authorizations and drops are included. ▪ 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.

3.5.2.3 Exception Handling and Log Tables.

Table 9. Exception Handling and Log Tables

Table Name	Description
GRE_EXCP	▪ An exception handling log table to track the failed transaction data on GRE exam
LOG_AUTOPROC	▪ A log table that tracks execution of automation process ▪ Records starting and ending time when executed each stored procedure
LOG_CLASSCHANGE	▪ A log table that tracks change made on student's classification status
LOG_COURSEGRADE	▪ A log table that tracks change made on student's courses taken information of STUDENTCOURSES table
LOG_NEWSTD	▪ A log table that tracks new student record inserted into GRADSTUDENTS table by the automation procedure
LOG_STATUSCHANGE	▪ A log table that tracks any student's status change in GRADSTUDENTS table by the automation procedure
LOG_GPA_BELOW3POINT0	▪ A log table that stores those whose GPA blows 3.0
NUKELOG_CTL	▪ A lookup table which provides information for automation procedure to delete the old logs in an assigned timely intervals
STDCOURSES_EXCP	▪ An exception handling log table to track failed transactions on migrating student courses taken and grade
STDCOURSES_NOSTORE	▪ A log table to track students' non-stored (filtered out by procedure) courses and grade, such as PE class or non CSCI courses.
STDINFO_EXCP	▪ An exception handling log table to track failed transactions on migrating graduate student's bio/demo data.
STDPREQ_EXCP	▪ An exception handling log table to track failed transactions on inserting prerequisite requirements

TOEFL_EXCP	▪ An exception handling log table to track failed transactions on migrating student's TOEFL exam
PROB_EXCP	▪ An exception handling log table to track failed transactions on migrating student's probation information
COMMITTEE_EXCP	▪ An exception handling log table to track failed transactions on migrating student's committee information
MASTEROPTION_EXCP	▪ An exception handling log table to track failed transactions on migrating student' project or thesis information
USER_LOG	▪ A log table that tracks application login activities, such as login time, IP, and the number of failed login attempts.
VISIT_LOG	▪ A log table that tracks application usage by capturing visitor's activities, such as visited pages and visited time.

CHAPTER FOUR

APPLICATION IMPLEMENTATION

This chapter presents major function sets within the current application along with screenshots on page layouts, expected end-user inputs and outputs in all aspects of logical characteristics of application interface with the person who must use the system.

4.1 Main User Login

This page is the starting point for the system. There is no login requirement for prospective students to obtain recommendations and study-plan. However, a continuing student, faculty, coordinator, or DBA/developer will be required to enter a Login ID and Password created by the Web-based Database Management System.

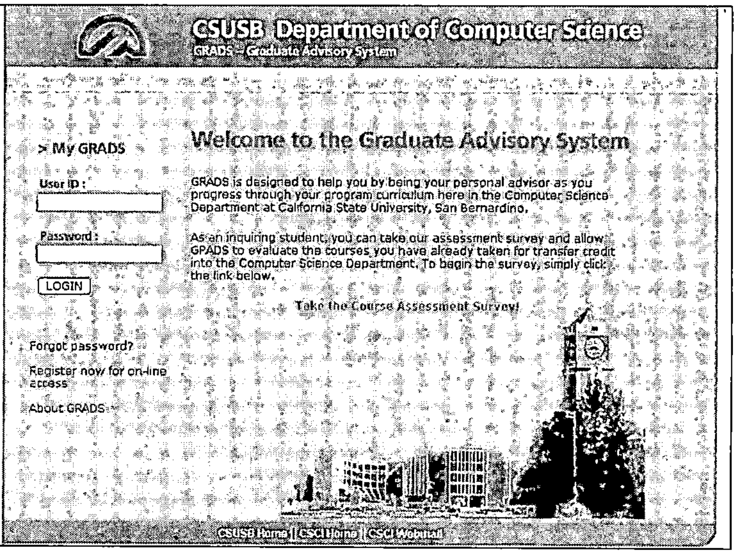
User Input:

- User ID
- Password

Output:

Verify then redirect to main page of appropriate user role:

- Continuing Students
- Coordinator
- Faculty
- DBA/Developer
- Login error message



CSUSB Department of Computer Science
GRADS – Graduate Advisory System
> My GRADS
User ID :
Password :
LOGIN
Forgot password?
Register now for on-line access
About GRADS
CSUSB Home | CSCI Home | CSCI Webmail

Welcome to the Graduate Advisory System

GRADS is designed to help you by being your personal advisor as you progress through your program curriculum here in the Computer Science Department at California State University, San Bernardino.

As an inquiring student, you can take our assessment survey and allow GPADS to evaluate the courses you have already taken for transfer credit into the Computer Science Department. To begin the survey, simply click the link below.

Take the Course Assessment Survey!

Figure 11. Main User Login Page

4.2 Continuing Student: GRADS Access Request

This function allows the user to fill out a form to request a GRADS account. The request is then sent via email to the designated GRADS administrator.

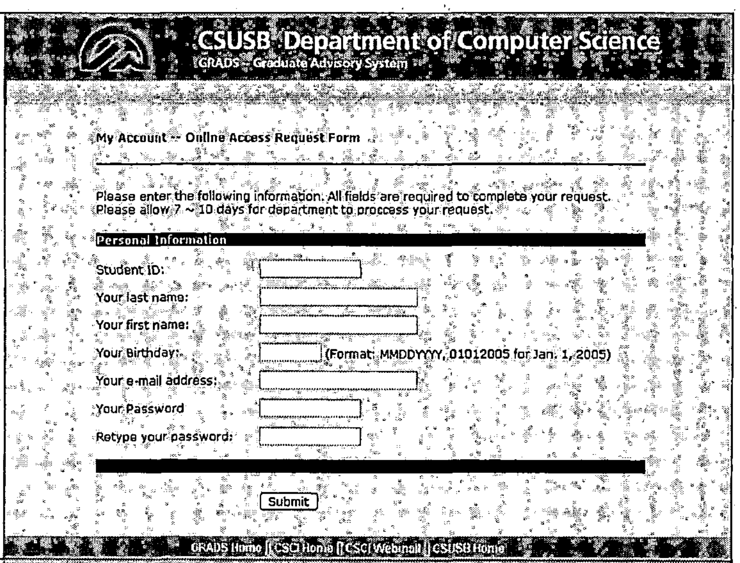
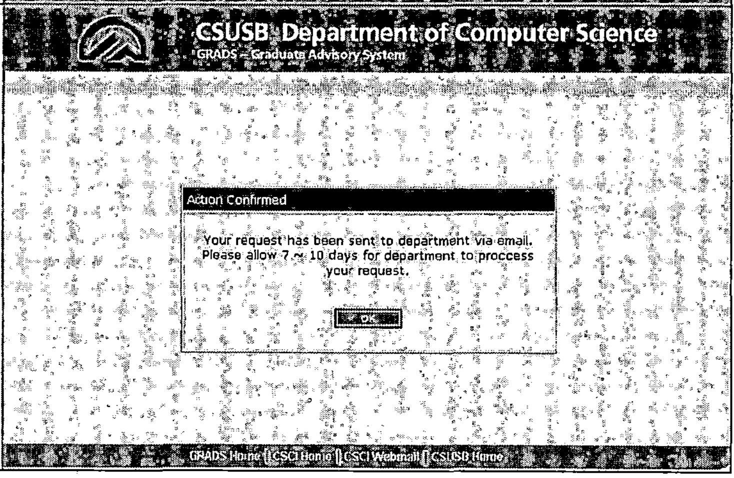
User Input: ▪ Student ID ▪ Student Name ▪ DOB ▪ Email ▪ Desired password	 CSUSB Department of Computer Science GRADS – Graduate Advisory System My Account -- Online Access Request Form Please enter the following information. All fields are required to complete your request. Please allow 7 ~ 10 days for department to proccess your request. Personal Information Student ID: Your last name: Your first name: Your Birthday: (Format: MMDDYYYY, 01012005 for Jan. 1, 2005) Your e-mail address: Your Password Retype your password: Submit GRADS Home || CSCI Home || CSCI Webmail || CSUSB Home
Output: ▪ System message describes the result of processing ▪ A confirmation mail to student ▪ A request mail to administrator	 CSUSB Department of Computer Science GRADS – Graduate Advisory System Action Confirmed Your request has been sent to department via email. Please allow 7 ~ 10 days for department to proccess your request. OK GRADS Home || CSCI Home || CSCI Webmail || CSUSB Home

Figure 12. Grads Access Request Page

4.3 Continuing Student: Password Recovery

This function verifies the user's request and to recover password and automatically retrieves the password and sends using the student's registered email account.

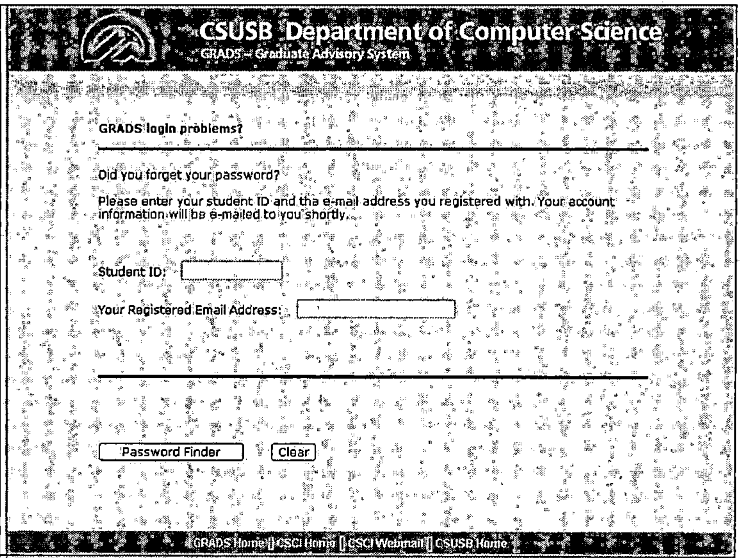
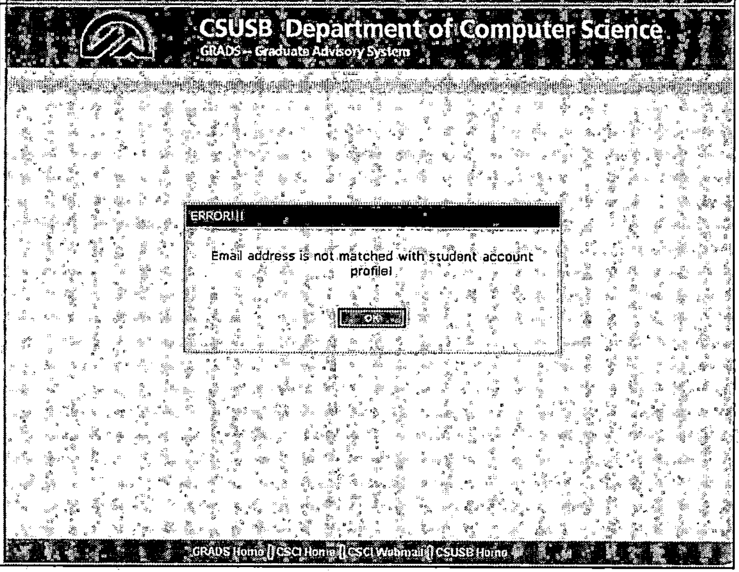
User Input: ▪ Student ID ▪ Registered email address	 CSUSB Department of Computer Science GRADS – Graduate Advisory System GRADS login problems? Did you forget your password? Please enter your student ID and the e-mail address you registered with. Your account information will be e-mailed to you shortly. Student ID: Your Registered Email Address: Password Finder Clear GRADS Home [] CSCI Home [] CSCI Webmail [] CSUSB Home
Output: ▪ System message describes the result of processing ▪ Email password to student registered email account, if information matched	 CSUSB Department of Computer Science GRADS – Graduate Advisory System ERROR!!! Email address is not matched with student account profile!

Figure 13. Password Recovery Page and the Processing Result Message Page

4.4 Continuing Student: Contact Information

This function allows a student to view his/her personal contact information. It also provides a link to an information update page to update his/her contact information.

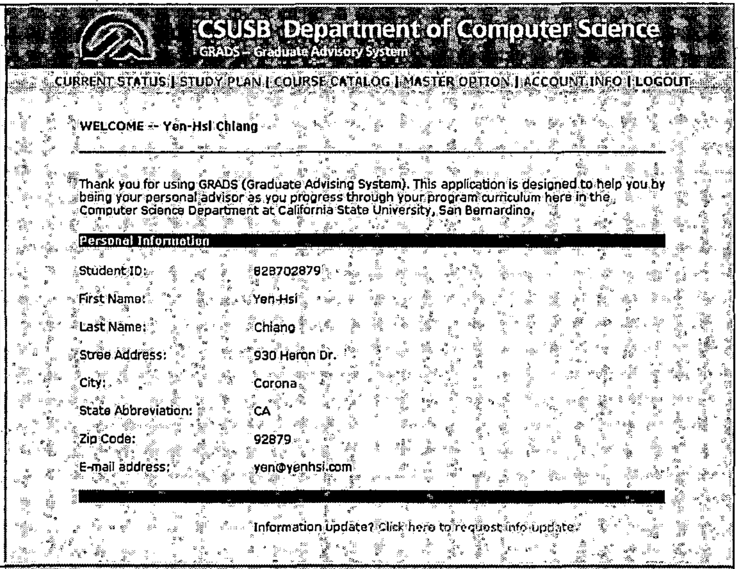
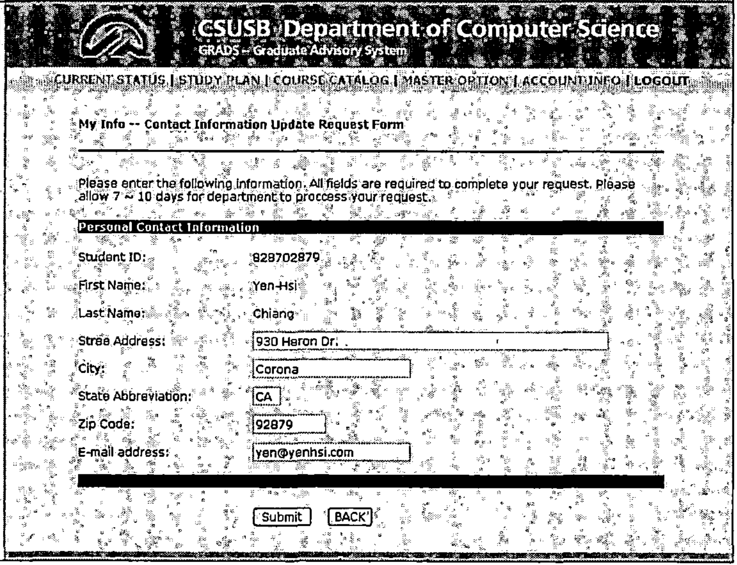
User Input: - None Output: - Student ID - Name - Street address - Email	 CSUSB Department of Computer Science GRADS -- Graduate Advisory System CURRENT STATUS | STUDY PLAN | COURSE CATALOG | MASTER OPTION | ACCOUNT INFO | LOGOUT WELCOME -- Yen-Hsi Chiang Thank you for using GRADS (Graduate Advising System). This application is designed to help you by being your personal advisor as you progress through your program curriculum here in the Computer Science Department at California State University, San Bernardino. **Personal Information** Student ID: 828702879 First Name: Yen-Hsi Last Name: Chiang Stree Address: 930 Heron Dr. City: Corona State Abbreviation: CA Zip Code: 92879 E-mail address: yen@yenhsi.com Information update? Click here to request info update.
User Input: - Street address - City - State - Zip - Email address Output: - System message describes the result of processing - Confirmation mail to student	 CSUSB Department of Computer Science GRADS -- Graduate Advisory System CURRENT STATUS | STUDY PLAN | COURSE CATALOG | MASTER OPTION | ACCOUNT INFO | LOGOUT Submit BACK My Info -- Contact Information Update Request Form Please enter the following information. All fields are required to complete your request. Please allow 7 ~ 10 days for department to process your request. **Personal Contact Information** Student ID: 828702879 First Name: Yen-Hsi Last Name: Chiang Stree Address: 930 Heron Dr. City: Corona State Abbreviation: CA Zip Code: 92879 E-mail address: yen@yenhsi.com

Figure 14. View Student Contact Information Pages

4.5 Continuing Student: Performance Report

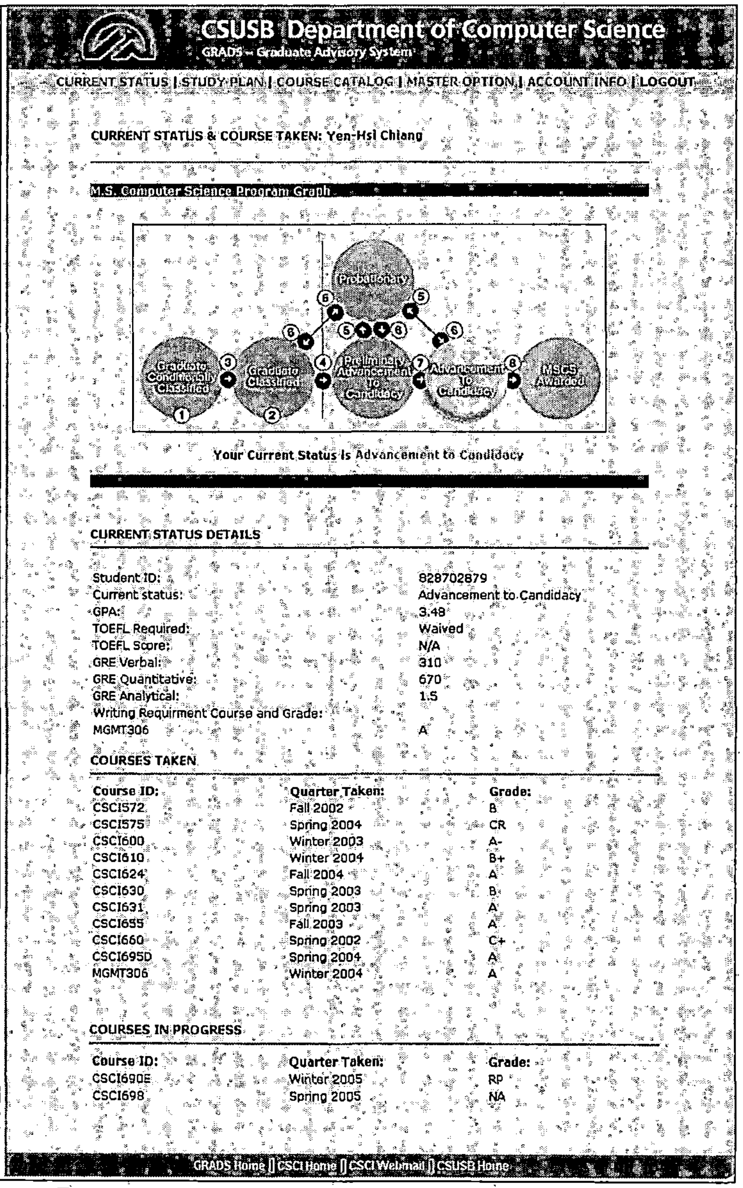
The performance report function provides the student his/her courses and grades history, current enrollment and academic status in the graduate program.

User Input:

- None

Output:

- Current status (Probationary, Graduate Conditionally Classified, Graduate Classified, Preliminary Advancement to Candidacy, Advancement to Candidacy, and MSCS Awarded) in the program
- GPA
- TOEFL score
- GRE Score
- Course and grades history
- Current course(s) registered for


CSUSB Department of Computer Science
GRADS – Graduate Advisory System
CURRENT STATUS | STUDY PLAN | COURSE CATALOG | MASTER OPTION | ACCOUNT INFO | LOGOUT
M.S. Computer Science Program Graph
Probationary
Graduate Conditionally Classified
Graduate Classified
Preliminary Advancement To Candidacy
Advancement To Candidacy
MSCS Awarded
1
2
3
4
5
6
7
8
GRADS Home [] CSCI Home [] CSCI Webmail [] CSUSB Home

CURRENT STATUS & COURSE TAKEN: Yen-Hsi Chiang

Your Current Status Is Advancement to Candidacy

CURRENT STATUS DETAILS

Student ID:	828702879
Current status:	Advancement to Candidacy
GPA:	3.48
TOEFL Required:	Waived
TOEFL Score:	N/A
GRE Verbal:	310
GRE Quantitative:	670
GRE Analytical:	1.5
Writing Requirment Course and Grade:	
MGMT306	A

COURSES TAKEN

Course ID:	Quarter Taken:	Grade:
CSCI572	Fall 2002	B
CSCI575	Spring 2004	CR
CSCI600	Winter 2003	A-
CSCI610	Winter 2004	B+
CSCI624	Fall 2004	A
CSCI630	Spring 2003	B
CSCI631	Spring 2003	A
CSCI655	Fall 2003	A
CSCI660	Spring 2002	C+
CSCI695D	Spring 2004	A
MGMT306	Winter 2004	A

COURSES IN PROGRESS

Course ID:	Quarter Taken:	Grade:
CSCI690E	Winter 2005	RP
CSCI698	Spring 2005	NA

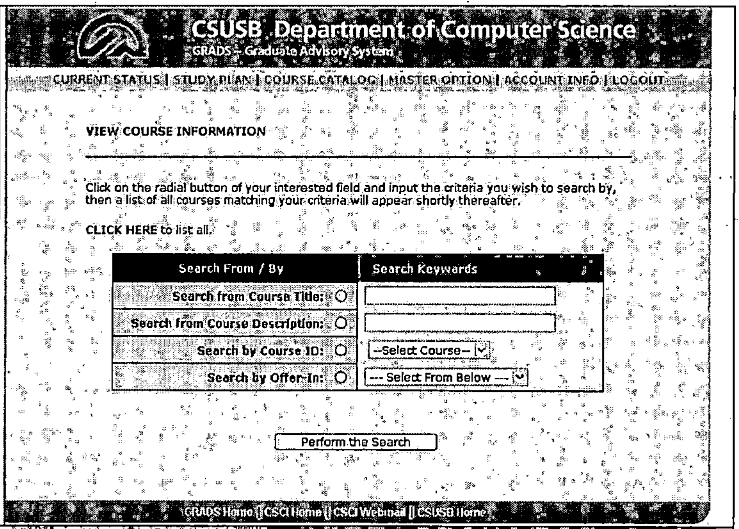
Figure 15. Performance Report Page

4.6 Continuing Student: Course Catalog

This function allows a student to perform a keyword search on course title or course description. After student enters the search criteria, the screen presents a list of matching results with embedded links to pop-up details of course information that are available within the Computer Science Department at CSUSB.

User Input:
- Searching keyword, or
- Course ID, or
- Course offered quarter

Output:
- A list of course IDs, course titles, courses offered quarter, and units which matches the search criteria.
- Embedded links on course ID



CSUSB Department of Computer Science
GRADS – Graduate Advisory System
CURRENT STATUS | STUDY PLAN | COURSE CATALOG | MASTER OPTION | ACCOUNT INFO | LOGOUT

VIEW COURSE INFORMATION

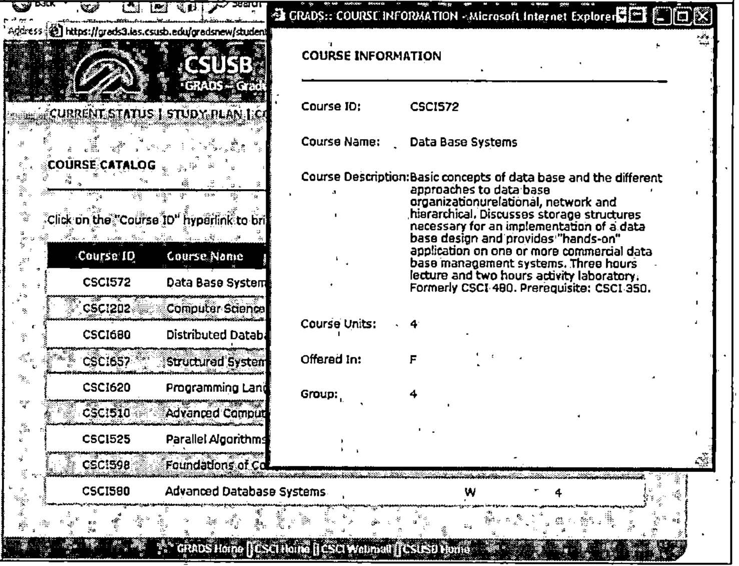
Click on the radial button of your interested field and input the criteria you wish to search by, then a list of all courses matching your criteria will appear shortly thereafter.

CLICK HERE to list all.

Search From / By	Search Keywards
Search from Course Title: ○	
Search from Course Description: ○	
Search by Course ID: ○	--Select Course--
Search by Offer-In: ○	--- Select From Below ---

User Input:
- Click on course ID

Output:
- A pop-up page with detailed course description and information



Address https://grads3.ias.csusb.edu/gradsnew/student
CSUSB
GRADS – Grad
CURRENT STATUS | STUDY PLAN | C

COURSE CATALOG

Click on the "Course ID" hyperlink to bri

Course ID	Course Name		
CSCI572	Data Base System		
CSCI202	Computer Science		
CSCI680	Distributed Datab		
CSCI657	Structured System		
CSCI620	Programming Lang		
CSCI510	Advanced Comput		
CSCI525	Parallel Algorithms		
CSCI598	Foundations of Co		
CSCI580	Advanced Database Systems	W	4

GRADS:: COURSE INFORMATION - Microsoft Internet Explorer

COURSE INFORMATION

Course ID: CSCI572

Course Name: Data Base Systems

Course Description: Basic concepts of data base and the different approaches to data base organizationurelational, network and hierarchical. Discusses storage structures necessary for an implementation of a data base design and provides "hands-on" application on one or more commercial data base management systems. Three hours lecture and two hours activity laboratory. Formerly CSCI 480. Prerequisite: CSCI 350.

Course Units: 4

Offered In: F

Group: 4

Figure 16. Course Catalog Searching and Result Pages

4.7 Continuing Student: Master Option

This set of functions allows a student to view his/her chosen master's option (track, title, advisors, and committee) and request an information update. The completed information updates will be emailed to the graduate program assistant for further processing.

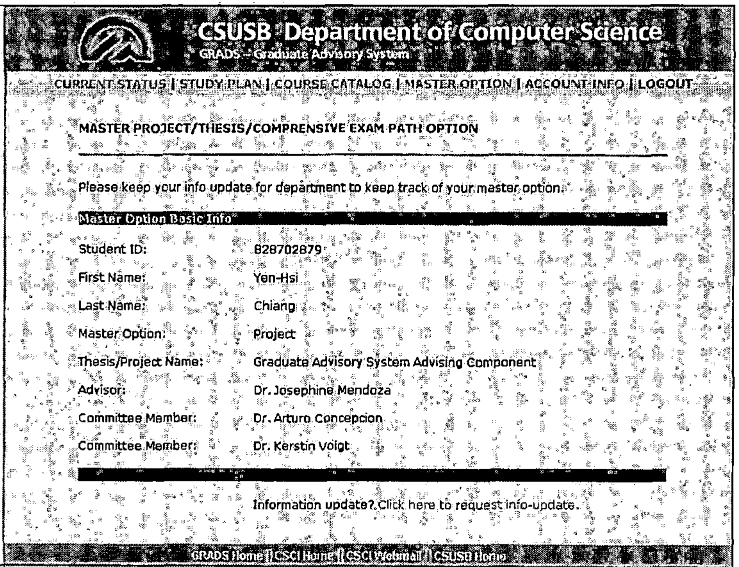
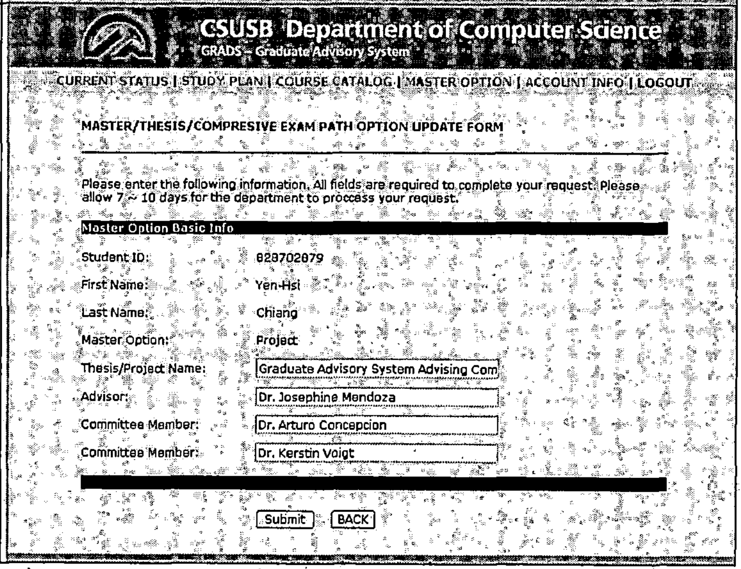
User Input: ▪ None Output: ▪ Student ID ▪ Name ▪ Master Option ▪ Thesis/Project name, if any ▪ Advisor name ▪ Committee members' names	 CSUSB Department of Computer Science GRADS – Graduate Advisory System CURRENT STATUS | STUDY PLAN | COURSE CATALOG | MASTER OPTION | ACCOUNT INFO | LOGOUT MASTER PROJECT/THESIS/COMPRENSIVE EXAM PATH OPTION Please keep your info update for department to keep track of your master option. Master Option Basic Info Student ID: 828702879 First Name: Yen-Hsi Last Name: Chiang Master Option: Project Thesis/Project Name: Graduate Advisory System Advising Component Advisor: Dr. Josephine Mendoza Committee Member: Dr. Arturo Concepcion Committee Member: Dr. Kerstin Voigt Information update?.Click here to request info-update.
User Input: ▪ Project/Thesis name ▪ Advisor name ▪ Committee members' name Output: ▪ System message describes the result of processing ▪ A confirmation mail to student ▪ A request mail to PA	 CSUSB Department of Computer Science GRADS – Graduate Advisory System CURRENT STATUS | STUDY PLAN | COURSE CATALOG | MASTER OPTION | ACCOUNT INFO | LOGOUT MASTER/THESIS/COMPRESIVE EXAM PATH OPTION UPDATE FORM Please enter the following information. All fields are required to complete your request. Please allow 7 ~ 10 days for the department to proccess your request. Master Option Basic Info Student ID: 828702879 First Name: Yen-Hsi Last Name: Chiang Master Option: Project Thesis/Project Name: Graduate Advisory System Advising Com Advisor: Dr. Josephine Mendoza Committee Member: Dr. Arturo Concepcion Committee Member: Dr. Kerstin Voigt

Figure 17. Master Option Information Pages

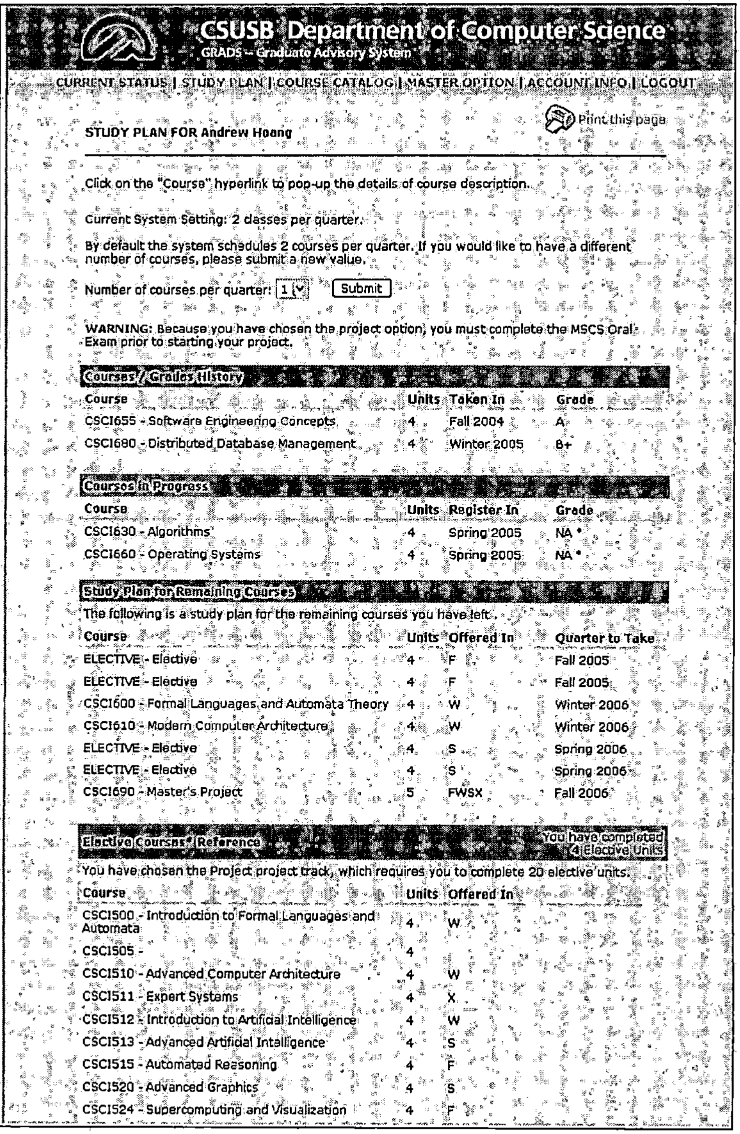
4.8 Continuing Student: Study Plan

This function provides a student with an effective study plan (courses taken, prerequisites needed, core courses needed, and electives) to meet the requirements of the MSCS program.

Default Setting:
- 2 courses per quarter

User Input:
- Number of courses planned to be completed or taken per quarter
- Printer icon or text link "Print Version View"

Output:
- Course/Grades history
- Course(s) in progress
- Study plan for remaining courses
- Embedded link on course title to pop-up course description
- A printer-friendly study plan



CSUSB Department of Computer Science

GRADS – Graduate Advisory System

CURRENT STATUS | STUDY PLAN | COURSE CATALOG | MASTER OPTION | ACCOUNT INFO | LOGOUT

Print this page

STUDY PLAN FOR Andrew Hoang

Click on the "Course" hyperlink to pop-up the details of course description.

Current System Setting: 2 classes per quarter.

By default the system schedules 2 courses per quarter. If you would like to have a different number of courses, please submit a new value.

Number of courses per quarter: 1 Submit

WARNING: Because you have chosen the project option, you must complete the MSCS Oral Exam prior to starting your project.

Courses / Grades History

Course	Units	Taken In	Grade
CSCI655 - Software Engineering Concepts	4	Fall 2004	A
CSCI680 - Distributed Database Management	4	Winter 2005	B+

Courses in Progress

Course	Units	Register In	Grade
CSCI630 - Algorithms	4	Spring 2005	NA *
CSCI660 - Operating Systems	4	Spring 2005	NA *

Study Plan for Remaining Courses

The following is a study plan for the remaining courses you have left.

Course	Units	Offered In	Quarter to Take
ELECTIVE - Elective	4	F	Fall 2005
ELECTIVE - Elective	4	F	Fall 2005
CSCI600 - Formal Languages and Automata Theory	4	W	Winter 2006
CSCI610 - Modern Computer Architecture	4	W	Winter 2006
ELECTIVE - Elective	4	S	Spring 2006
ELECTIVE - Elective	4	S	Spring 2006
CSCI690 - Master's Project	5	FWSX	Fall 2006

Elective Courses* Reference

You have completed 4 Elective Units

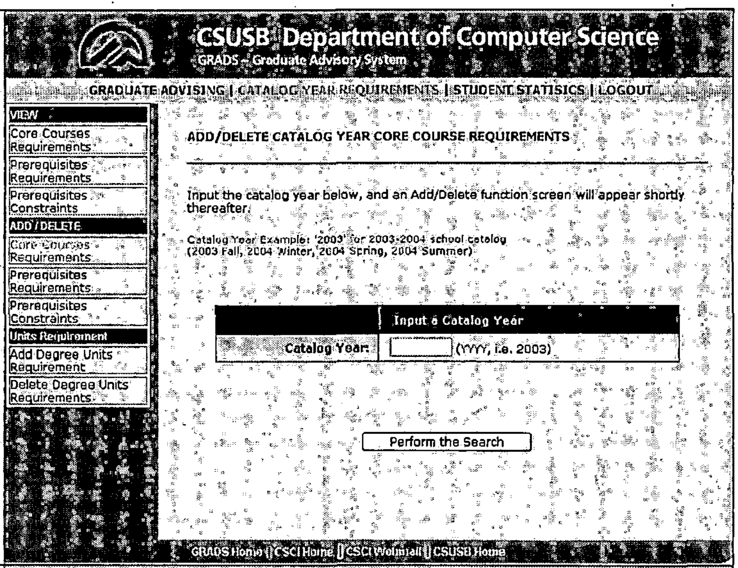
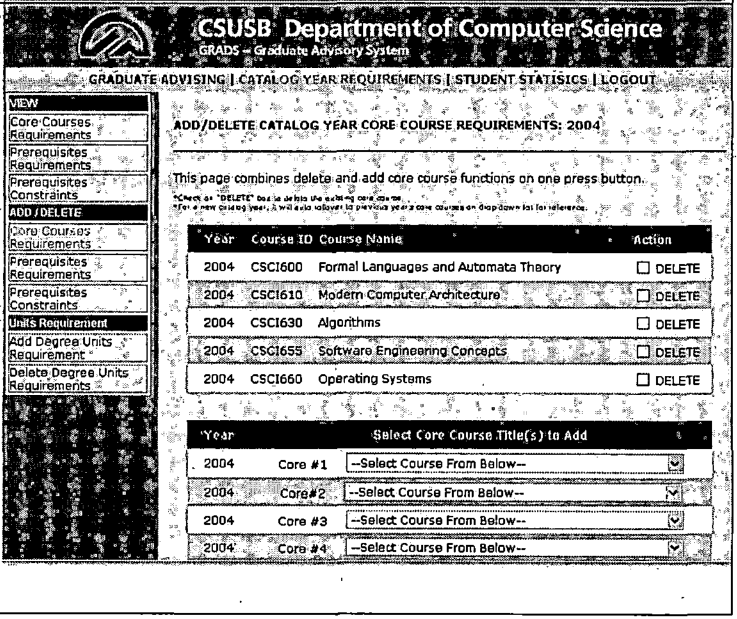
You have chosen the Project project track, which requires you to complete 20 elective units.

Course	Units	Offered In
CSCI500 - Introduction to Formal Languages and Automata	4	W
CSCI505 -	4	
CSCI510 - Advanced Computer Architecture	4	W
CSCI511 - Expert Systems	4	X
CSCI512 - Introduction to Artificial Intelligence	4	W
CSCI513 - Advanced Artificial Intelligence	4	S
CSCI515 - Automated Reasoning	4	F
CSCI520 - Advanced Graphics	4	S
CSCI524 - Supercomputing and Visualization	4	F

Figure 18. Study Plan Page

4.9 Coordinator: Add/Delete Catalog Year Prerequisite

This set of functions allows the Coordinator to set up the administrative rules for course requirements in the MSCS program for a given catalog year.

User Input: ▪ Catalog year Output: ▪ Prerequisite requirements	
Default Setting: ▪ Automatically roll-over to previous year's prerequisite requirements for a new catalog year User Input: ▪ Click on DELETE check box(s) to delete course(s), or ▪ Select the course(s) on drop-down list to add Output: ▪ System message describes the result of processing	

CSUSB Department of Computer Science
GRADS – Graduate Advisory System

GRADUATE ADVISING | CATALOG YEAR REQUIREMENTS | STUDENT STATISICS | LOGOUT

VIEW
Core Courses Requirements
Prerequisites Requirements
Prerequisites Constraints
ADD / DELETE
Core Courses Requirements
Prerequisites Requirements
Prerequisites Constraints
Units Requirement
Add Degree Units Requirement
Delete Degree Units Requirements

ADD/DELETE CATALOG YEAR CORE COURSE REQUIREMENTS

Input the catalog year below, and an Add/Delete function screen will appear shortly thereafter.

Catalog Year Example: '2003' for 2003-2004 school catalog (2003 Fall, 2004 Winter, 2004 Spring, 2004 Summer)

Input a Catalog Year

Catalog Year: [ ] (YYYY, i.e. 2003)

CSUSB Department of Computer Science
GRADS – Graduate Advisory System

GRADUATE ADVISING | CATALOG YEAR REQUIREMENTS | STUDENT STATISICS | LOGOUT

VIEW
Core Courses Requirements
Prerequisites Requirements
Prerequisites Constraints
ADD / DELETE
Core Courses Requirements
Prerequisites Requirements
Prerequisites Constraints
Units Requirement
Add Degree Units Requirement
Delete Degree Units Requirements

ADD/DELETE CATALOG YEAR CORE COURSE REQUIREMENTS: 2004

This page combines delete and add core course functions on one press button.

*Check on "DELETE" box to delete the existing core course.
*For a new catalog year, it will auto rollover to previous year's core courses on drop down list for reference.

Year	Course ID	Course Name	Action
2004	CSCI600	Formal Languages and Automata Theory	□ DELETE
2004	CSCI610	Modern Computer Architecture	□ DELETE
2004	CSCI630	Algorithms	□ DELETE
2004	CSCI655	Software Engineering Concepts	□ DELETE
2004	CSCI660	Operating Systems	□ DELETE

Year		Select Core Course Title(s) to Add
2004	Core #1	--Select Course From Below--
2004	Core#2	--Select Course From Below--
2004	Core #3	--Select Course From Below--
2004	Core #4	--Select Course From Below--

Figure 19. Add/Delete Catalog Year Prerequisites Pages

4.10 Coordinator: Add/Delete Catalog Year Core Courses

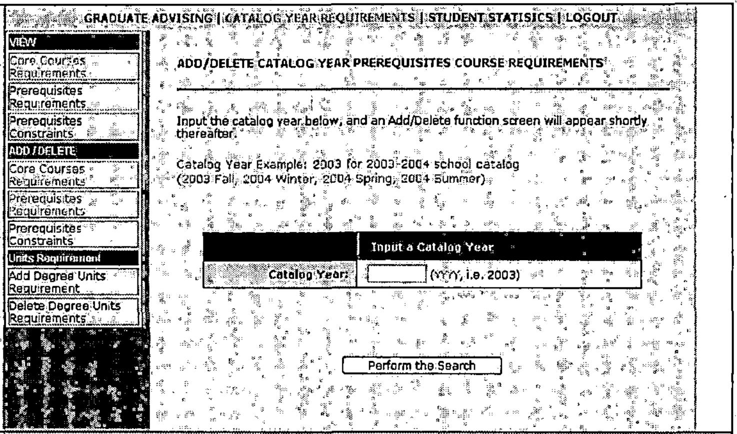
This set of functions allows coordinator to set up administrative rules for core course requirements of the MSCS program for a given catalog year.

User Input:

- Catalog year

Output:

- Core course requirements



GRADUATE ADVISING | CATALOG YEAR REQUIREMENTS | STUDENT STATISICS | LOGOUT

VIEW
- Core Courses Requirements
- Prerequisites Requirements
- Prerequisites Constraints

ADD / DELETE
- Core Courses Requirements
- Prerequisites Requirements
- Prerequisites Constraints

Units Requirement
- Add Degree Units Requirement
- Delete Degree Units Requirements

ADD/DELETE CATALOG YEAR PREREQUISITES COURSE REQUIREMENTS

Input the catalog year below, and an Add/Delete function screen will appear shortly thereafter.

Catalog Year Example: 2003 for 2003-2004 school catalog (2003 Fall, 2004 Winter, 2004 Spring, 2004 Summer)

	Input a Catalog Year
Catalog Year:	(YYYY, i.e. 2003)

Perform the Search

Default Setting:

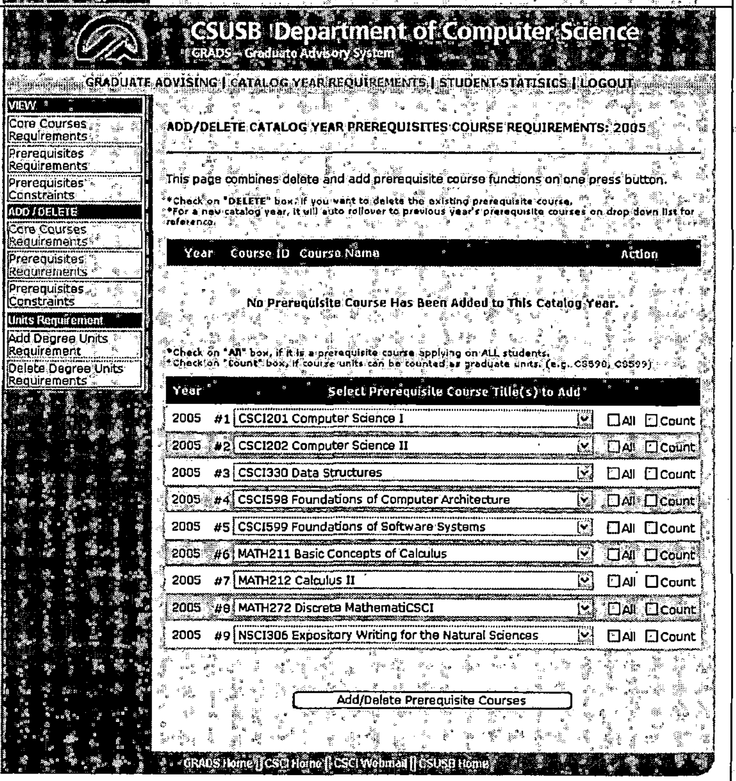
- Automatically roll-over to previous year's core course requirements for a new catalog year

User Input:

- Click on DELETE check box(s) for specified course(s), or
- Select the course(s) on drop-down list to add
- Click on ALL check box(s), if the course is required by all students
- Click on COUNT check box(s), if its unite is counted into graduate units

Output:

- System message describes the result of processing



CSUSB Department of Computer Science
GRADS – Graduate Advisory System

GRADUATE ADVISING | CATALOG YEAR REQUIREMENTS | STUDENT STATISICS | LOGOUT

VIEW
- Core Courses Requirements
- Prerequisites Requirements
- Prerequisites Constraints

ADD / DELETE
- Core Courses Requirements
- Prerequisites Requirements
- Prerequisites Constraints

Units Requirement
- Add Degree Units Requirement
- Delete Degree Units Requirements

ADD/DELETE CATALOG YEAR PREREQUISITES COURSE REQUIREMENTS: 2005

This page combines delete and add prerequisite course functions on one press button.

*Check on "DELETE" box, if you want to delete the existing prerequisite course.
*For a new catalog year, it will auto rollover to previous year's prerequisite courses on drop down list for reference.

Year	Course ID	Course Name	Action

No Prerequisite Course Has Been Added to This Catalog Year.

*Check on "All" box, if it is a prerequisite course applying on ALL students.
*Check on "Count" box, if course units can be counted as graduate units. (e.g. CS590, CS599)

Year		Select Prerequisite Course Title(s) to Add		
2005	#1	CSCI201 Computer Science I	□All	□Count
2005	#2	CSCI202 Computer Science II	□All	□Count
2005	#3	CSCI330 Data Structures	□All	□Count
2005	#4	CSCI598 Foundations of Computer Architecture	□All	□Count
2005	#5	CSCI599 Foundations of Software Systems	□All	□Count
2005	#6	MATH211 Basic Concepts of Calculus	□All	□Count
2005	#7	MATH212 Calculus II	□All	□Count
2005	#8	MATH272 Discrete MathematiCSCI	□All	□Count
2005	#9	NSCI306 Expository Writing for the Natural Sciences	□All	□Count

Add/Delete Prerequisite Courses

GRADS Home || CSCI Home || CSCI Webmail || CSUSB Home

Figure 20. Add/Delete Catalog Year Core Course Pages

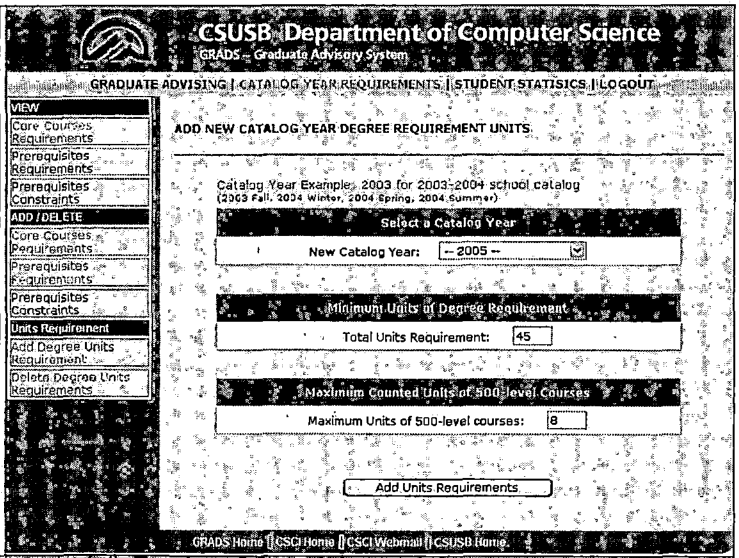
4.11 Coordinator: Add/Delete Catalog Year Graduate Units Requirement Rule

This set of functions allows the coordinator to set up administrative rules for the number of units required for an MSCS degree and the number of 500 level units allowed for an MSCS degree.

Default Setting:
- Automatically roll-over to previous year's requirement

User Input:
- Select new catalog year from drop-down list
- Total units require
- Maximum counted units of 500-level courses

Output:
- System message describes the result of processing


CSUSB Department of Computer Science
GRADS – Graduate Advisory System
GRADUATE ADVISING | CATALOG YEAR REQUIREMENTS | STUDENT STATISICS | LOGOUT
VIEW
Core Courses Requirements
Prerequisites Requirements
Prerequisites Constraints
ADD / DELETE
Core Courses Requirements
Prerequisites Requirements
Prerequisites Constraints
Units Requirement
Add Degree Units Requirement
Delete Degree Units Requirements
Add Units Requirements
GRADS Home || CSCI Home || CSCI Webmail || CSUSB Home

ADD NEW CATALOG YEAR DEGREE REQUIREMENT UNITS

Catalog Year Example: 2003 for 2003-2004 school catalog
(2003 Fall, 2004 Winter, 2004 Spring, 2004 Summer)

Select a Catalog Year

New Catalog Year: -- 2005 --

Minimum Units of Degree Requirement

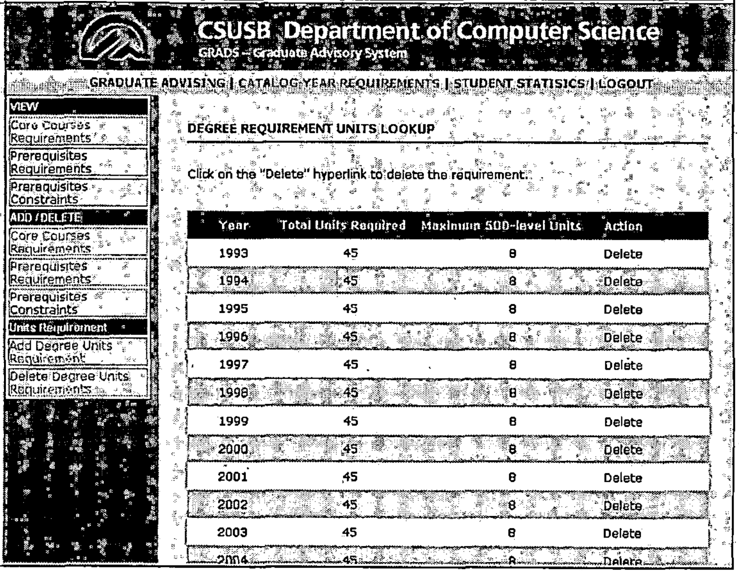
Total Units Requirement: 45

Maximum Counted Units of 500-level Courses

Maximum Units of 500-level courses: 8

User Input:
- Click on DELETE link to delete the units requirement for a specified year

Output:
- System message describes the result of processing


CSUSB Department of Computer Science
GRADS – Graduate Advisory System
GRADUATE ADVISING | CATALOG YEAR REQUIREMENTS | STUDENT STATISICS | LOGOUT
VIEW
Core Courses Requirements
Prerequisites Requirements
Prerequisites Constraints
ADD / DELETE
Core Courses Requirements
Prerequisites Requirements
Prerequisites Constraints
Units Requirement
Add Degree Units Requirement
Delete Degree Units Requirements

DEGREE REQUIREMENT UNITS LOOKUP

Click on the "Delete" hyperlink to delete the requirement.

Year	Total Units Required	Maximum 500-level Units	Action
1993	45	8	Delete
1994	45	8	Delete
1995	45	8	Delete
1996	45	8	Delete
1997	45	8	Delete
1998	45	8	Delete
1999	45	8	Delete
2000	45	8	Delete
2001	45	8	Delete
2002	45	8	Delete
2003	45	8	Delete
2004	45	8	Delete

Figure 21. Add/Delete Catalog Year Graduate Units Requirement Pages

4.12 Coordinator: Add/Waive Student's Prerequisite and Recommend Courses

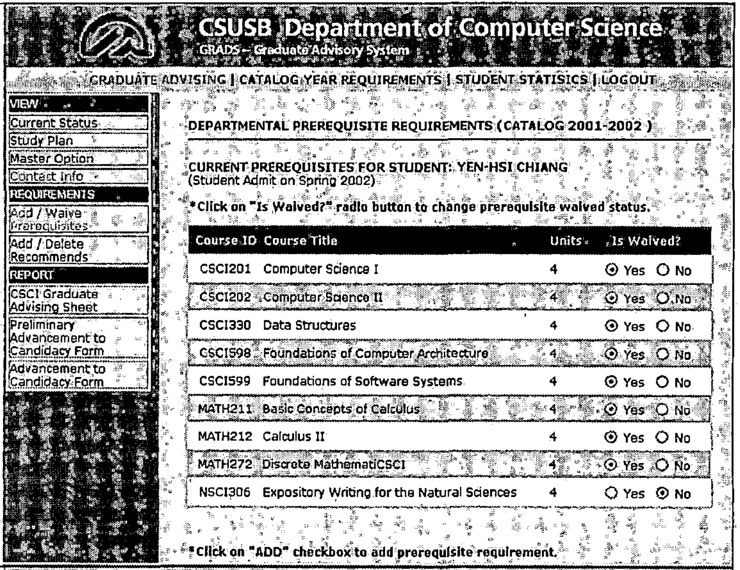
This set of functions allows the coordinator to add or/and waive/delete prerequisite courses and recommend courses for a student.

User Input:

- Click on Yes/No radio buttons to Waive/Not Waive prerequisites for this student, or
- Select a prerequisite course from drop-down list to add prerequisite for this student

Output:

- System message describes the result of processing


CSUSB Department of Computer Science
GRADS – Graduate Advisory System
GRADUATE ADVISING | CATALOG YEAR REQUIREMENTS | STUDENT STATISICS | LOGOUT
VIEW
Current Status
Study Plan
Master Option
Contact Info
REQUIREMENTS
Add / Waive Prerequisites
Add / Delete Recommends
REPORT
CSCI Graduate Advising Sheet
Preliminary Advancement to Candidacy Form
Advancement to Candidacy Form

DEPARTMENTAL PREREQUISITE REQUIREMENTS (CATALOG 2001-2002)

CURRENT PREREQUISITES FOR STUDENT: YEN-HSI CHIANG
(Student Admit on Spring 2002)

*Click on "Is Waived?" radio button to change prerequisite waived status.

Course ID	Course Title	Units	Is Waived?
CSCI201	Computer Science I	4	● Yes ○ No
CSCI202	Computer Science II	4	● Yes ○ No
CSCI330	Data Structures	4	● Yes ○ No
CSCI598	Foundations of Computer Architecture	4	● Yes ○ No
CSCI599	Foundations of Software Systems	4	● Yes ○ No
MATH211	Basic Concepts of Calculus	4	● Yes ○ No
MATH212	Calculus II	4	● Yes ○ No
MATH272	Discrete MathematiCSCI	4	● Yes ○ No
NSCI306	Expository Writing for the Natural Sciences	4	○ Yes ● No

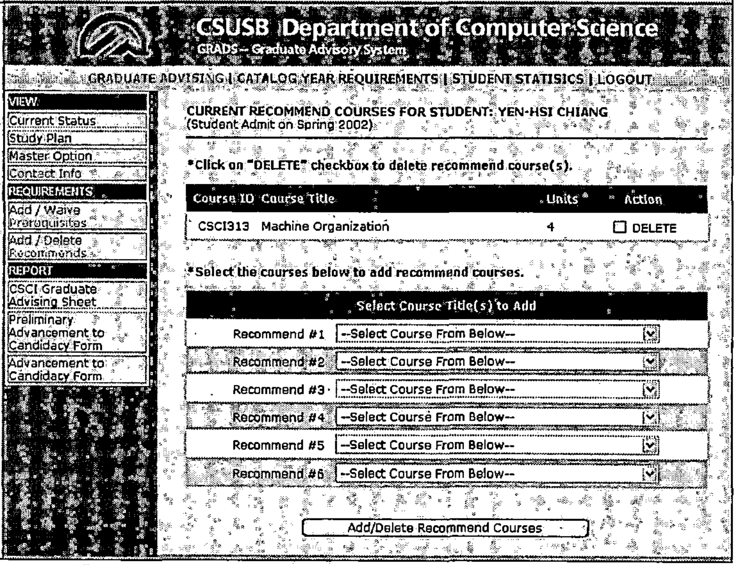
*Click on "ADD" checkbox to add prerequisite requirement.

User Input:

- Click on DELETE checkbox to delete recommended course for this student, or
- Select a recommended course from drop-down list to add as recommended course(s) for this student

Output:

- System message describes the result of processing


CSUSB Department of Computer Science
GRADS – Graduate Advisory System
GRADUATE ADVISING | CATALOG YEAR REQUIREMENTS | STUDENT STATISICS | LOGOUT
VIEW
Current Status
Study Plan
Master Option
Contact Info
REQUIREMENTS
Add / Waive Prerequisites
Add / Delete Recommends
REPORT
CSCI Graduate Advising Sheet
Preliminary Advancement to Candidacy Form
Advancement to Candidacy Form

CURRENT RECOMMEND COURSES FOR STUDENT: YEN-HSI CHIANG
(Student Admit on Spring 2002)

*Click on "DELETE" checkbox to delete recommend course(s).

Course ID	Course Title	Units	Action
CSCI313	Machine Organization	4	□ DELETE

*Select the courses below to add recommend courses.

Select Course Title(s) to Add	
Recommend #1	--Select Course From Below--
Recommend #2	--Select Course From Below--
Recommend #3	--Select Course From Below--
Recommend #4	--Select Course From Below--
Recommend #5	--Select Course From Below--
Recommend #6	--Select Course From Below--

Figure 22. Add/Waive Student's Prerequisite and Recommend Courses Pages

4.13 Coordinator: Computer Science Graduate Advising Sheet

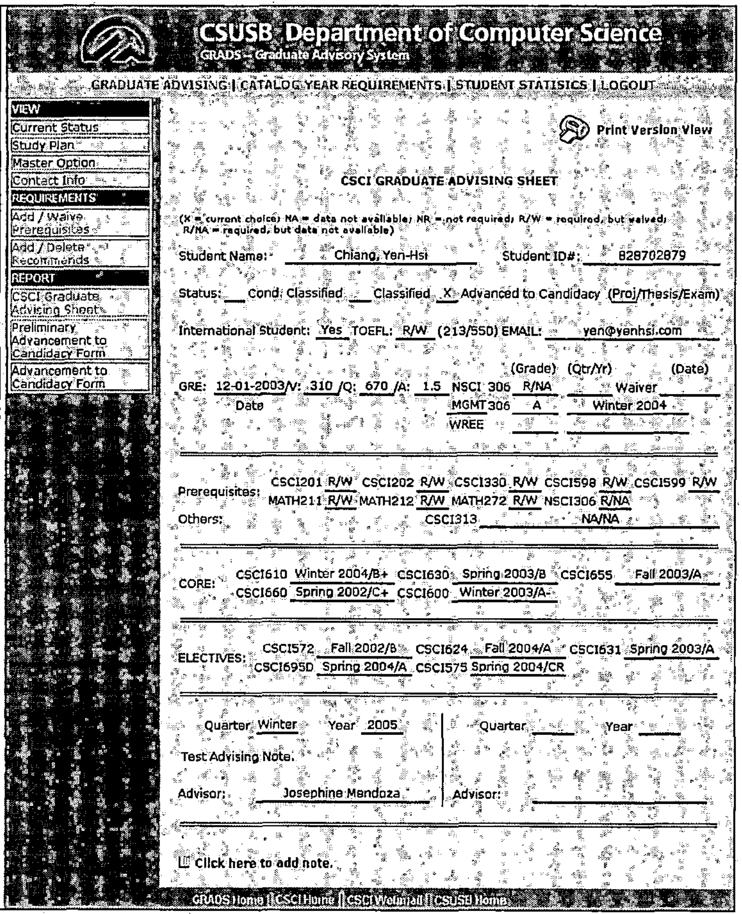
The electronic version of CSCI Graduate Advising sheet will be used by the graduate advisor to obtain a summary of information (GRE scores, WREE scores, TOEFL scores, prerequisite course grades, core course grades, and elective course grades), and input advising notes.

User Input:

- Text link "Click here to add note"
- Printer icon or text link "Print Version View"

Output:

- A printer-friendly Advising Sheet
- An embedded link at the bottom of this form which links to an advising note input page for the Coordinator to add advising notes


CSUSB Department of Computer Science
GRADS -- Graduate Advisory System
GRADUATE ADVISING | CATALOG YEAR REQUIREMENTS | STUDENT STATISICS | LOGOUT
VIEW
Current Status
Study Plan
Master Option
Contact Info
REQUIREMENTS
Add / Waive Prerequisites
Add / Delete Recommends
REPORT
CSCI Graduate Advising Sheet
Preliminary Advancement to Candidacy Form
Advancement to Candidacy Form
Print Version View
GRADS Home | CSCI Home | CSCI Webmail | CSUSB Home

CSCI GRADUATE ADVISING SHEET

(X = current choice; NA = data not available; NR = not required; R/W = required, but waived; R/NA = required, but data not available)

Student Name: Chiang, Yen-Hsi Student ID#: 828702879

Status: __ Cond. Classified __ Classified X Advanced to Candidacy (Proj/Thesis/Exam)

International Student: Yes TOEFL: R/W (213/550) EMAIL: yen@yenhsi.com

		(Grade)	(Qtr/Yr)	(Date)
GRE: 12-01-2003/V: 310 /Q: 670 /A: 1.5 Date	NSCI 306	R/NA		Waiver
	MGMT 306	A	Winter 2004	
	WREE			

Prerequisites: CSCI201 R/W CSCI202 R/W CSCI330 R/W CSCI598 R/W CSCI599 R/W MATH211 R/W MATH212 R/W MATH272 R/W NSCI306 R/NA

Others: CSCI313 NA/NA

CORE: CSCI610 Winter 2004/B+ CSCI630 Spring 2003/B CSCI655 Fall 2003/A CSCI660 Spring 2002/C+ CSCI600 Winter 2003/A-

ELECTIVES: CSCI572 Fall 2002/B CSCI624 Fall 2004/A CSCI631 Spring 2003/A CSCI695D Spring 2004/A CSCI575 Spring 2004/CR

Quarter Winter Year 2005
Test Advising Note.
Advisor: Josephine Mendoza

Quarter __ Year __
Advisor: __

Click here to add note.

Figure 23. Computer Science Graduate Advising Sheet Page

4.14 Coordinator: Preliminary Advancement to Candidacy Form

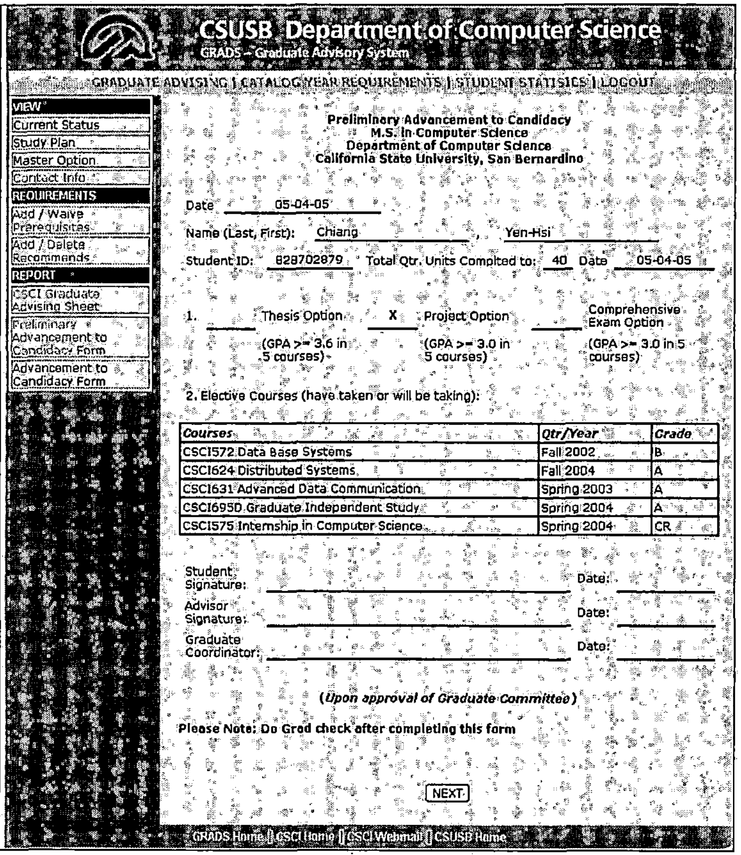
This function allows the coordinator to replicate the hard copy of the Preliminary Advancement to Candidacy Form in an electronic format.

User Input:

- Select the course title from drop-down list to add elective course(s), if the total units of elective courses taken do not meet the minimum required units

Output:

- Preliminary Advancement to Candidacy Sheet


CSUSB Department of Computer Science
GRADS – Graduate Advisory System
GRADUATE ADVISING | CATALOG YEAR REQUIREMENTS | STUDENT STATISICS | LOGOUT
VIEW
Current Status
Study Plan
Master Option
Contact Info
REQUIREMENTS
Add / Waive Prerequisites
Add / Delete Recommends
REPORT
CSCI Graduate Advising Sheet
Preliminary Advancement to Candidacy Form
Advancement to Candidacy Form
NEXT
GRADS Home || CSCI Home || CSCI Webmail || CSUSB Home

Preliminary Advancement to Candidacy
M.S. in Computer Science
Department of Computer Science
California State University, San Bernardino

Date 05-04-05

Name (Last, First): Chiang , Yen-Hsi

Student ID: 828702879 Total Qtr. Units Complted to: 40 Date 05-04-05

1. ____ Thesis Option (GPA >= 3.6 in 5 courses) X Project Option (GPA >= 3.0 in 5 courses) ____ Comprehensive Exam Option (GPA >= 3.0 in 5 courses)

2. Elective Courses (have taken or will be taking):

Courses	*Qtr/Year*	*Grade*
CSCI572 Data Base Systems	Fall 2002	B
CSCI624 Distributed Systems	Fall 2004	A
CSCI631 Advanced Data Communication	Spring 2003	A
CSCI695D Graduate Independent Study	Spring 2004	A
CSCI575 Internship in Computer Science	Spring 2004	CR

Student Signature: ____________ Date: ______

Advisor Signature: ____________ Date: ______

Graduate Coordinator: ____________ Date: ______

(Upon approval of Graduate Committee)

Please Note: Do Grad check after completing this form

Figure 24. Preliminary Advancement to Candidacy Form Page

4.15 Coordinator: Advancement to Candidacy Form

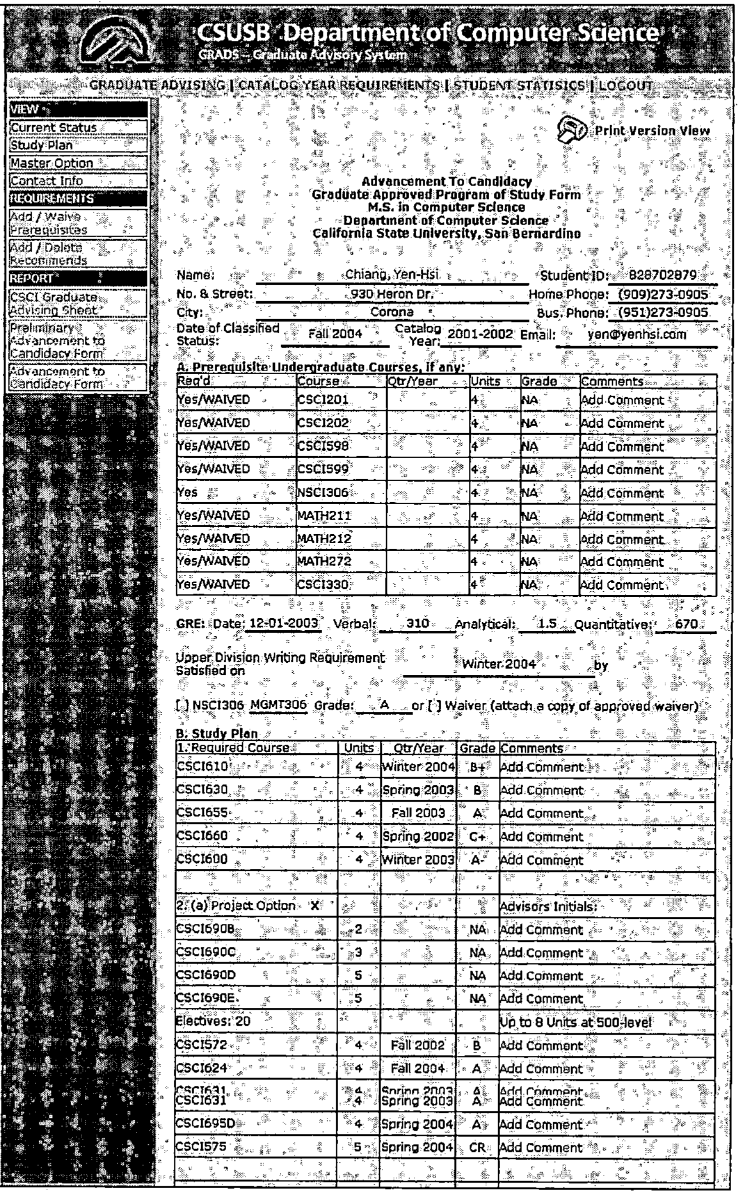
This function allows the coordinator to replicate the hard copy of the Advancement to Candidacy Form in an electronic format.

User Input:

- Printer icon or text link "Print Version View"
- Click on the embedded link of Add Comment

Output:

- A printer-friendly Advancement to Candidacy Form
- Comment input page for proceeding selected course


CSUSB Department of Computer Science
GRADS – Graduate Advisory System
GRADUATE ADVISING | CATALOG YEAR REQUIREMENTS | STUDENT STATISICS | LOGOUT
VIEW
Current Status
Study Plan
Master Option
Contact Info
REQUIREMENTS
Add / Waive Prerequisites
Add / Delete Recommends
REPORT
CSCI Graduate Advising Sheet
Preliminary Advancement to Candidacy Form
Advancement to Candidacy Form
Print Version View

Advancement To Candidacy
Graduate Approved Program of Study Form
M.S. in Computer Science
Department of Computer Science
California State University, San Bernardino

Name: Chiang, Yen-Hsi Student ID: 828702879
No. & Street: 930 Heron Dr. Home Phone: (909)273-0905
City: Corona Bus. Phone: (951)273-0905
Date of Classified Status: Fall 2004 Catalog Year: 2001-2002 Email: yen@yenhsi.com

A. Prerequisite Undergraduate Courses, if any:

Req'd	Course	Qtr/Year	Units	Grade	Comments
Yes/WAIVED	CSCI201		4	NA	Add Comment
Yes/WAIVED	CSCI202		4	NA	Add Comment
Yes/WAIVED	CSCI598		4	NA	Add Comment
Yes/WAIVED	CSCI599		4	NA	Add Comment
Yes	NSCI306		4	NA	Add Comment
Yes/WAIVED	MATH211		4	NA	Add Comment
Yes/WAIVED	MATH212		4	NA	Add Comment
Yes/WAIVED	MATH272		4	NA	Add Comment
Yes/WAIVED	CSCI330		4	NA	Add Comment

GRE: Date: 12-01-2003 Verbal: 310 Analytical: 1.5 Quantitative: 670

Upper Division Writing Requirement Satisfied on Winter 2004 by

[ ] NSCI306 MGMT306 Grade: A or [ ] Waiver (attach a copy of approved waiver)

B. Study Plan

1. Required Course	Units	Qtr/Year	Grade	Comments
CSCI610	4	Winter 2004	B+	Add Comment
CSCI630	4	Spring 2003	B	Add Comment
CSCI655	4	Fall 2003	A	Add Comment
CSCI660	4	Spring 2002	C+	Add Comment
CSCI600	4	Winter 2003	A-	Add Comment
2. (a) Project Option X				Advisors Initials:
CSCI690B	2		NA	Add Comment
CSCI690C	3		NA	Add Comment
CSCI690D	5		NA	Add Comment
CSCI690E	5		NA	Add Comment
Electives: 20				Up to 8 Units at 500-level
CSCI572	4	Fall 2002	B	Add Comment
CSCI624	4	Fall 2004	A	Add Comment
CSCI631 CSCI631	4 4	Spring 2003 Spring 2003	A A	Add Comment Add Comment
CSCI695D	4	Spring 2004	A	Add Comment
CSCI575	5	Spring 2004	CR	Add Comment

Figure 25. Advancement to Candidacy Form Page

4.16 Coordinator/Faculty: Statistical Information for Existing Student

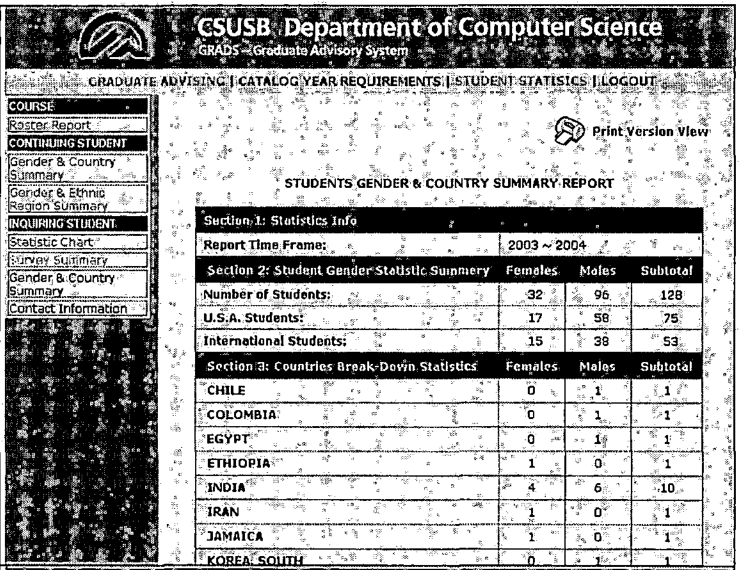
This set of functions allows the faculty/coordinator to see a statistical report of existing students based on gender versus country and gender versus ethnicity.

User Input:

- Time frame statistical criteria
- Printer icon or text link "Print Version View"

Output:

- A printer-friendly statistical report -- student's gender VS. country


CSUSB Department of Computer Science
GRADS – Graduate Advisory System
GRADUATE ADVISING | CATALOG YEAR REQUIREMENTS | STUDENT STATISICS | LOGOUT
COURSE
Roster Report
CONTINUING STUDENT
Gender & Country Summary
Gender & Ethnic Region Summary
INQUIRING STUDENT
Statistic Chart
Survey Summary
Gender & Country Summary
Contact Information
Print Version View

STUDENTS GENDER & COUNTRY SUMMARY REPORT

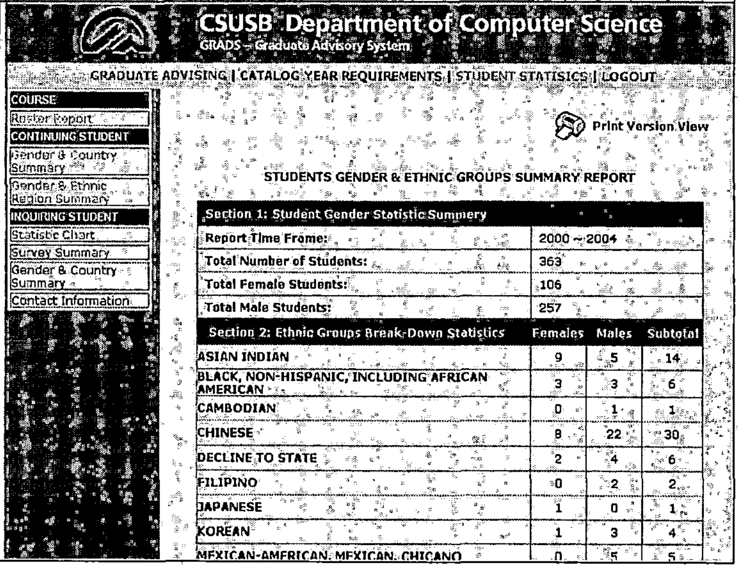
Section 1: Statistics Info			
Report Time Frame:	2003 ~ 2004		
Section 2: Student Gender Statistic Summery	**Females**	**Males**	**Subtotal**
Number of Students:	32	96	128
U.S.A. Students:	17	58	75
International Students:	15	38	53
Section 3: Countries Break-Down Statistics	**Females**	**Males**	**Subtotal**
CHILE	0	1	1
COLOMBIA	0	1	1
EGYPT	0	1	1
ETHIOPIA	1	0	1
INDIA	4	6	10
IRAN	1	0	1
JAMAICA	1	0	1
KOREA, SOUTH	0	1	1

User Input:

- Time frame statistical criteria
- Printer icon or text link "Print Version View"

Output:

- A printer-friendly statistical report -- student's gender VS. ethnicity


CSUSB Department of Computer Science
GRADS – Graduate Advisory System
GRADUATE ADVISING | CATALOG YEAR REQUIREMENTS | STUDENT STATISICS | LOGOUT
COURSE
Roster Report
CONTINUING STUDENT
Gender & Country Summary
Gender & Ethnic Region Summary
INQUIRING STUDENT
Statistic Chart
Survey Summary
Gender & Country Summary
Contact Information
Print Version View

STUDENTS GENDER & ETHNIC GROUPS SUMMARY REPORT

Section 1: Student Gender Statistic Summery			
Report Time Frame:	2000 ~ 2004		
Total Number of Students:	363		
Total Female Students:	106		
Total Male Students:	257		
Section 2: Ethnic Groups Break-Down Statistics	**Females**	**Males**	**Subtotal**
ASIAN INDIAN	9	5	14
BLACK, NON-HISPANIC, INCLUDING AFRICAN AMERICAN	3	3	6
CAMBODIAN	0	1	1
CHINESE	8	22	30
DECLINE TO STATE	2	4	6
FILIPINO	0	2	2
JAPANESE	1	0	1
KOREAN	1	3	4
MEXICAN-AMERICAN, MEXICAN, CHICANO	0	5	5

Figure 26. Statistical Information for Existing Student Page

4.17 Coordinator/Faculty: Statistical Information for Prospective Students

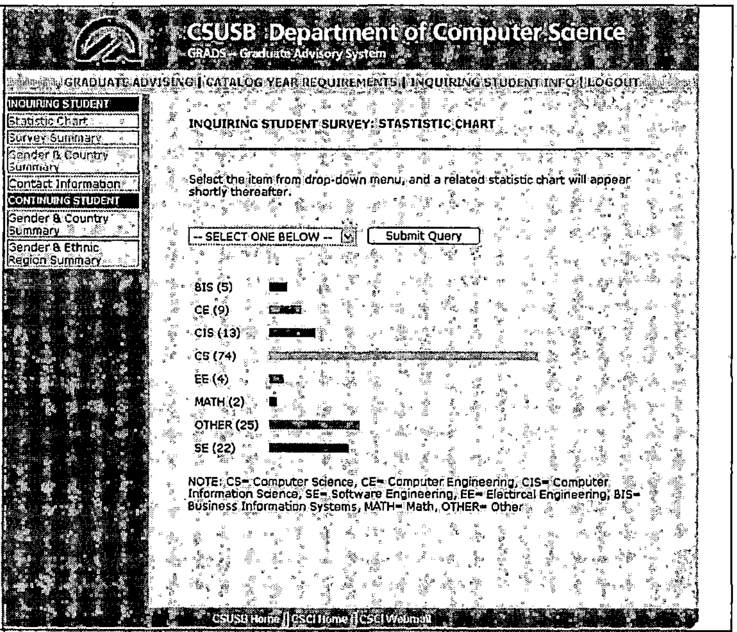
Department faculty/coordinator can utilize this set of functions to obtain statistical information on prospective students who have taken the surveys.

User Input:

- Time frame statistical criteria, and
- Select a survey question from drop-down list

Output:

- Statistical bar based on selected question


CSUSB Department of Computer Science
GRADS – Graduate Advisory System
GRADUATE ADVISING | CATALOG YEAR REQUIREMENTS | INQUIRING STUDENT INFO | LOGOUT
INQUIRING STUDENT
Statistic Chart
Survey Summary
Gender & Country Summary
Contact Information
CONTINUING STUDENT
Gender & Country Summary
Gender & Ethnic Region Summary
-- SELECT ONE BELOW --
Submit Query
CSUSB Home || CSCI Home || CSCI Webmail

INQUIRING STUDENT SURVEY: STASTISTIC CHART

Select the item from drop-down menu, and a related statistic chart will appear shortly thereafter.

BIS (5)
CE (9)
CIS (13)
CS (74)
EE (4)
MATH (2)
OTHER (25)
SE (22)

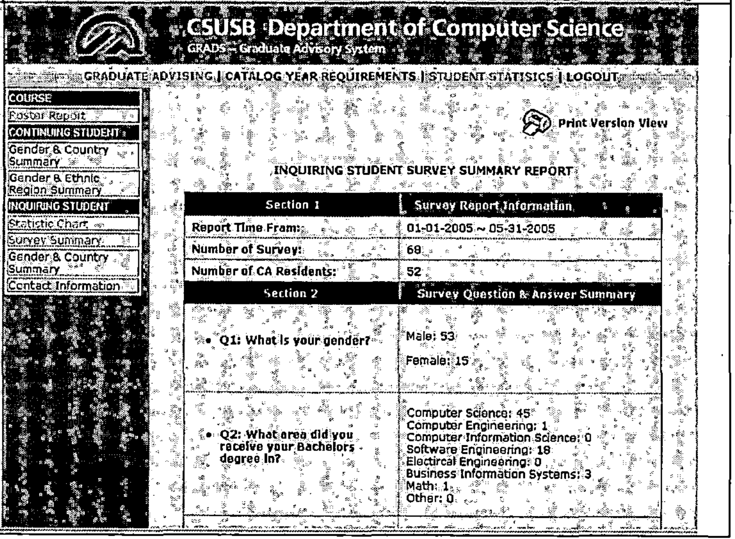
NOTE: CS= Computer Science, CE= Computer Engineering, CIS= Computer Information Science, SE= Software Engineering, EE= Electrical Engineering, BIS= Business Information Systems, MATH= Math, OTHER= Other

User Input:

- Printer icon or text link "Print Version View"

Output:

- A statistical report for survey statistical information


CSUSB Department of Computer Science
GRADS – Graduate Advisory System
GRADUATE ADVISING | CATALOG YEAR REQUIREMENTS | STUDENT STATISICS | LOGOUT
COURSE
Roster Report
CONTINUING STUDENT
Gender & Country Summary
Gender & Ethnic Region Summary
INQUIRING STUDENT
Statistic Chart
Survey Summary
Gender & Country Summary
Contact Information
Print Version View

INQUIRING STUDENT SURVEY SUMMARY REPORT

Section 1	Survey Report Information
Report Time Fram:	01-01-2005 ~ 05-31-2005
Number of Survey:	68
Number of CA Residents:	52
Section 2	**Survey Question & Answer Summary**
• Q1: What is your gender?	Male: 53 Female: 15
• Q2: What area did you receive your Bachelors degree in?	Computer Science: 45 Computer Engineering: 1 Computer Information Science: 0 Software Engineering: 18 Electircal Engineering: 0 Business Information Systems: 3 Math: 1 Other: 0

Figure 27. Prospective Students Statistical Information Pages

4.18 Prospective Student: Survey

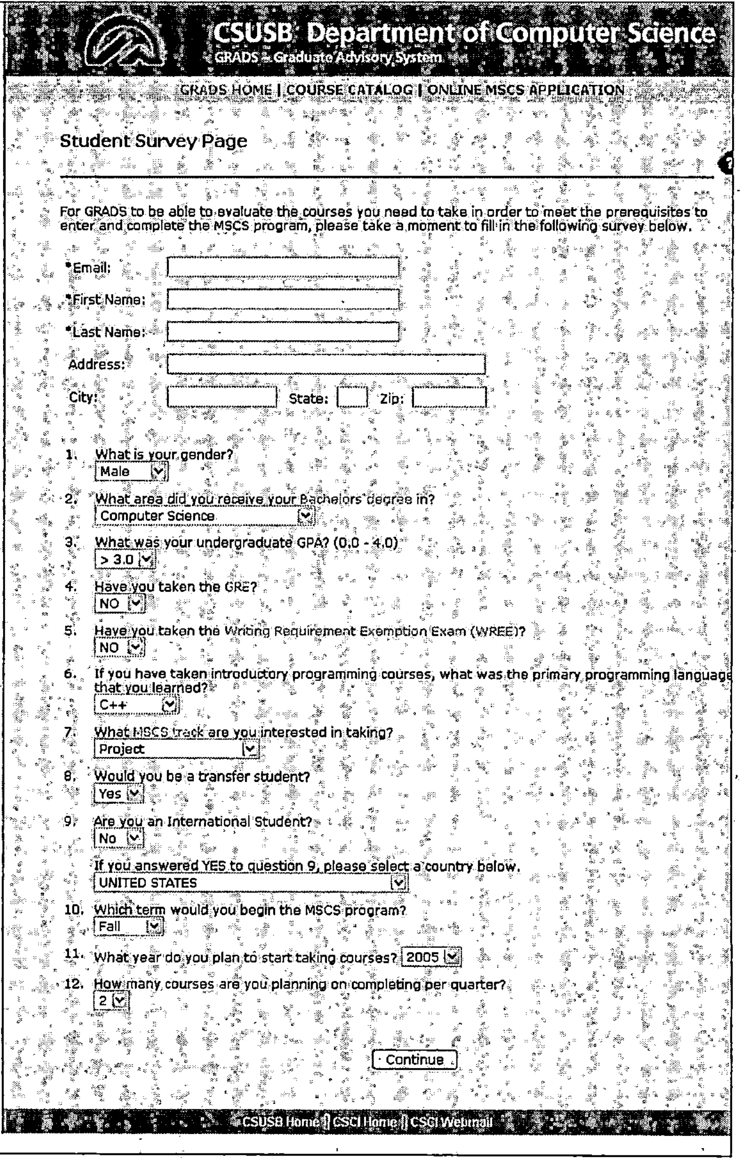
This page may be broken down into several pages for manageability, but the survey will ask the user information on his/her prior academic background.

User Input:

- Email address, name, street address, city, state, zip
- Bachelors Degree type earned (Computer Science, Computer Information Science, Software Engineering, Computer Engineering, Other)
- Primary programming language taken as introductory programming courses (i.e., C, FORTRAN, COBOL, C++, or Java)
- Degree program option the student is interested in taking at CSUSB (i.e., thesis, project, or cumulative test)
- The number of courses a quarter the student wishes to take (i.e., 1, 2, 3)
- Has GRE been taken
- Has WREE been taken

Output:

- Continue on transfer evaluation page



CSUSB Department of Computer Science
GRADS – Graduate Advisory System

GRADS HOME | COURSE CATALOG | ONLINE MSCS APPLICATION

Student Survey Page

For GRADS to be able to evaluate the courses you need to take in order to meet the prerequisites to enter and complete the MSCS program, please take a moment to fill in the following survey below.

*Email:
*First Name:
*Last Name:
Address:
City: State: Zip:

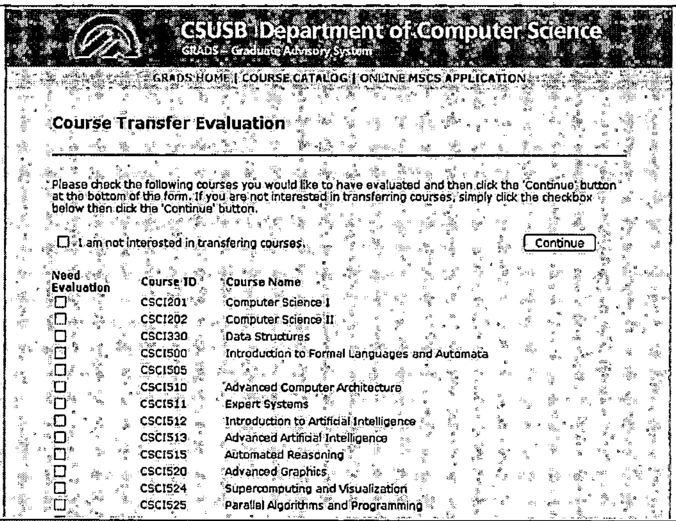
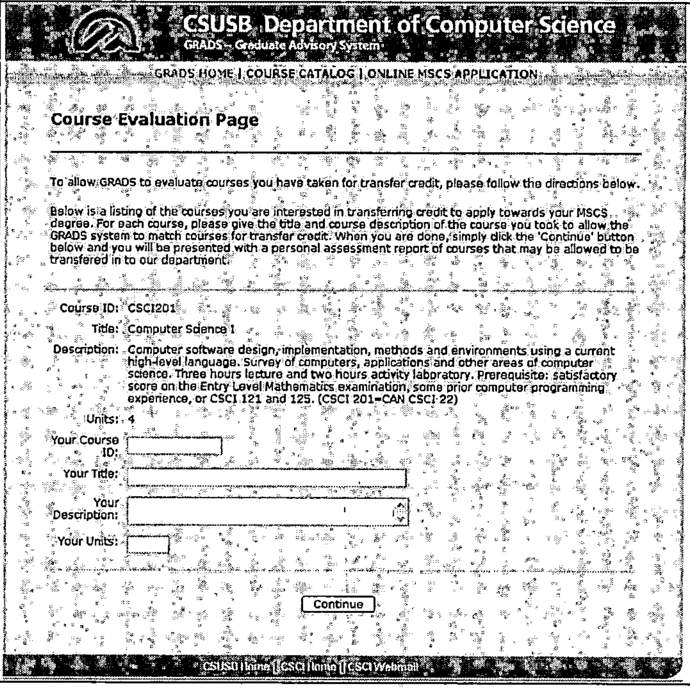
1. What is your gender?
   Male
2. What area did you receive your Bachelors degree in?
   Computer Science
3. What was your undergraduate GPA? (0.0 - 4.0)
   > 3.0
4. Have you taken the GRE?
   NO
5. Have you taken the Writing Requirement Exemption Exam (WREE)?
   NO
6. If you have taken introductory programming courses, what was the primary programming language that you learned?
   C++
7. What MSCS track are you interested in taking?
   Project
8. Would you be a transfer student?
   Yes
9. Are you an International Student?
   No

   If you answered YES to question 9, please select a country below.
   UNITED STATES
10. Which term would you begin the MSCS program?
    Fall
11. What year do you plan to start taking courses? 2005
12. How many courses are you planning on completing per quarter?
    2

Figure 28. Survey Page

4.19 Prospective Student: Evaluation

This set of functions allows the prospective student to select courses they are interested in transferring, and then the system will provide a personal assessment report of courses that may be allowed to be transferred.

User Input: - Click on course checkbox(s), then - Press on "Continue" button Output: - Continue on evaluation page	
User Input: - Transferring course ID - Transferring course title - Transferring course description - Transferring course units Output: - Study Plan	

CSUSB Department of Computer Science
GRADS – Graduate Advisory System
GRADS HOME | COURSE CATALOG | ONLINE MSCS APPLICATION

Course Transfer Evaluation

Please check the following courses you would like to have evaluated and then click the 'Continue' button at the bottom of the form. If you are not interested in transferring courses, simply click the checkbox below then click the 'Continue' button.

□ I am not interested in transfering courses. [Continue]

Need Evaluation	Course ID	Course Name
□	CSCI201	Computer Science I
□	CSCI202	Computer Science II
□	CSCI330	Data Structures
□	CSCI500	Introduction to Formal Languages and Automata
□	CSCI505	
□	CSCI510	Advanced Computer Architecture
□	CSCI511	Expert Systems
□	CSCI512	Introduction to Artificial Intelligence
□	CSCI513	Advanced Artificial Intelligence
□	CSCI515	Automated Reasoning
□	CSCI520	Advanced Graphics
□	CSCI524	Supercomputing and Visualization
□	CSCI525	Parallel Algorithms and Programming

CSUSB Department of Computer Science
GRADS – Graduate Advisory System
GRADS HOME | COURSE CATALOG | ONLINE MSCS APPLICATION

Course Evaluation Page

To allow GRADS to evaluate courses you have taken for transfer credit, please follow the directions below.

Below is a listing of the courses you are interested in transferring credit to apply towards your MSCS degree. For each course, please give the title and course description of the course you took to allow the GRADS system to match courses for transfer credit. When you are done, simply click the 'Continue' button below and you will be presented with a personal assessment report of courses that may be allowed to be transfered in to our department.

Course ID: CSCI201
Title: Computer Science I
Description: Computer software design, implementation, methods and environments using a current high-level language. Survey of computers, applications and other areas of computer science. Three hours lecture and two hours activity laboratory. Prerequisite: satisfactory score on the Entry Level Mathematics examination, some prior computer programming experience, or CSCI 121 and 125. (CSCI 201=CAN CSCI 22)
Units: 4
Your Course ID: ____
Your Title: ____
Your Description: ____
Your Units: ____

Figure 29. Evaluation Page

4.20 Prospective Student: Study Plan

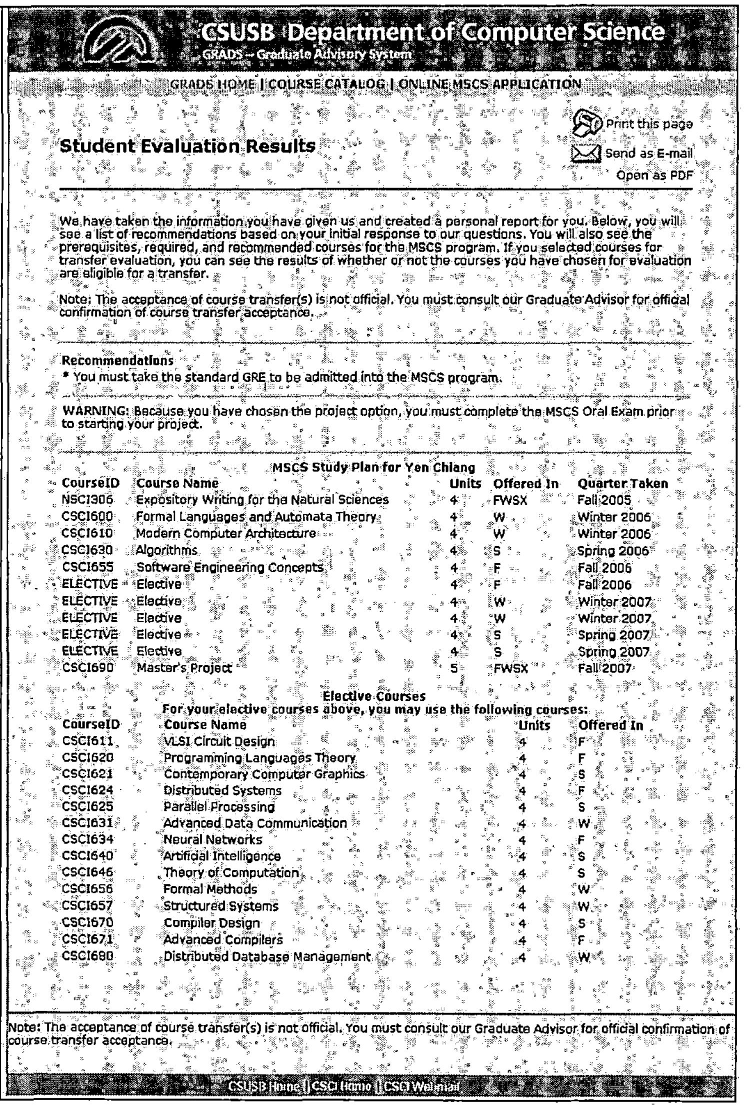
This function provides prospective students a list of recommendations and the result of transfer evaluation based on their initial response to the survey.

User Input:

- None, derived from preceding survey
- Printer icon or text link "Print Version View"
- Email icon or text link "Send as E-mail"
- Text link "Open as PDF"

Output:

- Recommendations and the transfer evaluation result
- A printer-friendly study plan
- E-mail the study plan to prospective student
- Study plan in PDF format


CSUSB Department of Computer Science
GRADS – Graduate Advisory System
GRADS HOME | COURSE CATALOG | ONLINE MSCS APPLICATION
Print this page
Send as E-mail
Open as PDF

Student Evaluation Results

We have taken the information you have given us and created a personal report for you. Below, you will see a list of recommendations based on your initial response to our questions. You will also see the prerequisites, required, and recommended courses for the MSCS program. If you selected courses for transfer evaluation, you can see the results of whether or not the courses you have chosen for evaluation are eligible for a transfer.

Note: The acceptance of course transfer(s) is not official. You must consult our Graduate Advisor for official confirmation of course transfer acceptance.

Recommendations

* You must take the standard GRE to be admitted into the MSCS program.

WARNING: Because you have chosen the project option, you must complete the MSCS Oral Exam prior to starting your project.

MSCS Study Plan for Yen Chiang

CourseID	Course Name	Units	Offered In	Quarter Taken
NSCI306	Expository Writing for the Natural Sciences	4	FWSX	Fall 2005
CSCI600	Formal Languages and Automata Theory	4	W	Winter 2006
CSCI610	Modern Computer Architecture	4	W	Winter 2006
CSCI630	Algorithms	4	S	Spring 2006
CSCI655	Software Engineering Concepts	4	F	Fall 2006
ELECTIVE	Elective	4	F	Fall 2006
ELECTIVE	Elective	4	W	Winter 2007
ELECTIVE	Elective	4	W	Winter 2007
ELECTIVE	Elective	4	S	Spring 2007
ELECTIVE	Elective	4	S	Spring 2007
CSCI690	Master's Project	5	FWSX	Fall 2007

Elective Courses

For your elective courses above, you may use the following courses:

CourseID	Course Name	Units	Offered In
CSCI611	VLSI Circuit Design	4	F
CSCI620	Programming Languages Theory	4	F
CSCI621	Contemporary Computer Graphics	4	S
CSCI624	Distributed Systems	4	F
CSCI625	Parallel Processing	4	S
CSCI631	Advanced Data Communication	4	W
CSCI634	Neural Networks	4	F
CSCI640	Artificial Intelligence	4	S
CSCI646	Theory of Computation	4	S
CSCI656	Formal Methods	4	W
CSCI657	Structured Systems	4	W
CSCI670	Compiler Design	4	S
CSCI671	Advanced Compilers	4	F
CSCI680	Distributed Database Management	4	W

Note: The acceptance of course transfer(s) is not official. You must consult our Graduate Advisor for official confirmation of course transfer acceptance.

Figure 30. Prospective Student Study Plan Page

4.21 Database Administrator/Developer: Control Logs

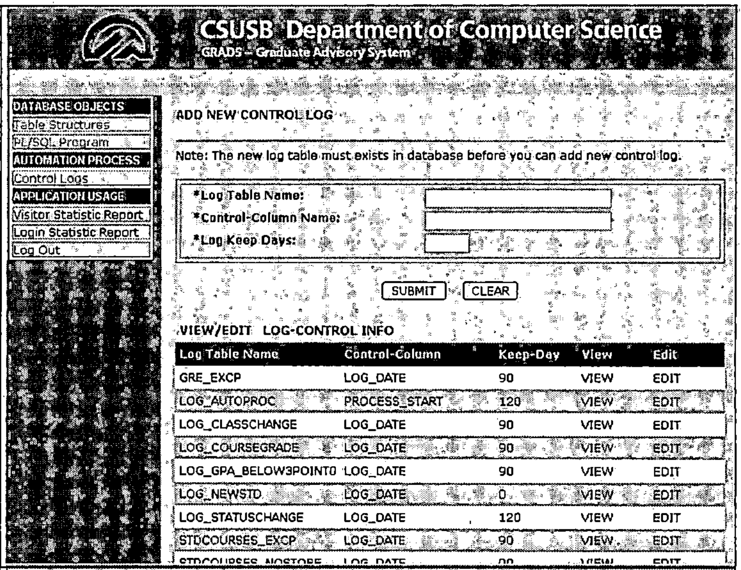
This set of functions allows the DBA/Developer to set up rules for scheduled procedure to delete the old logs. It also provides the DBA/Developer a handy tool to quickly access the log contents.

User Input:

- Log table name, column name, and log keep days for inserting a new log
- Click on "VIEW" to query the specified log table
- Click on "EDIT" to edit the keep days

Output:

- Processing result message on screen, or
- A pop-up query page



ADD NEW CONTROL LOG

Note: The new log table must exists in database before you can add new control log.

*Log Table Name:
*Control-Column Name:
*Log Keep Days:

SUBMIT CLEAR

VIEW/EDIT LOG-CONTROL INFO

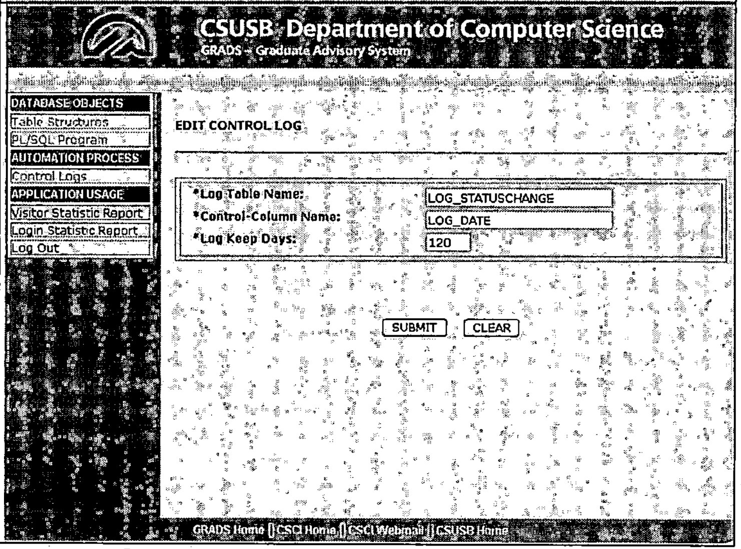
Log Table Name	Control-Column	Keep-Day	View	Edit
GRE_EXCP	LOG_DATE	90	VIEW	EDIT
LOG_AUTOPROC	PROCESS_START	120	VIEW	EDIT
LOG_CLASSCHANGE	LOG_DATE	90	VIEW	EDIT
LOG_COURSEGRADE	LOG_DATE	90	VIEW	EDIT
LOG_GPA_BELOW3POINT0	LOG_DATE	90	VIEW	EDIT
LOG_NEWSTD	LOG_DATE	0	VIEW	EDIT
LOG_STATUSCHANGE	LOG_DATE	120	VIEW	EDIT
STDCOURSES_EXCP	LOG_DATE	90	VIEW	EDIT
STDCOURSES_NOSTORE	LOG_DATE	90	VIEW	EDIT

User Input:

- Log table name, column name, and log keep days for update an existing log
- Click on "SUBMIT" to update s

Output:

- Processing result message on screen, or



EDIT CONTROL LOG

*Log Table Name: LOG_STATUSCHANGE
*Control-Column Name: LOG_DATE
*Log Keep Days: 120

SUBMIT CLEAR

Figure 31. Track System Update Logs Page

2.22 Database Administrator/Developer: View Table Structures

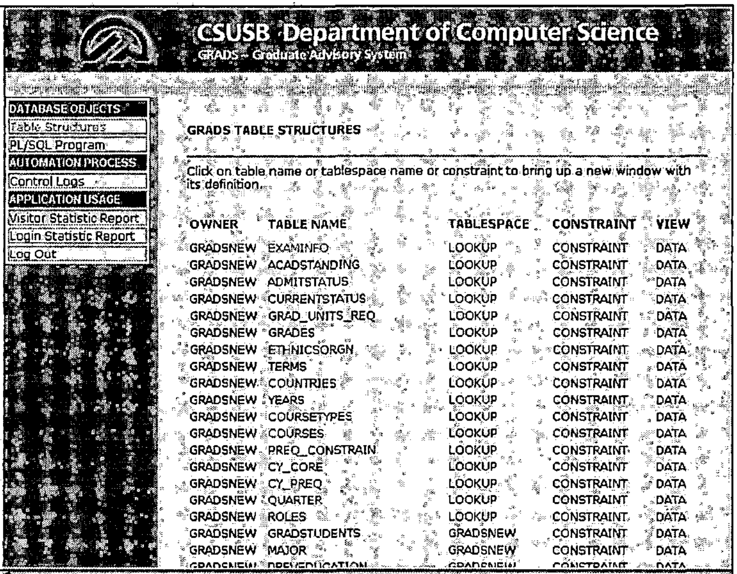
This set of functions allows the DBA/Developer to view application-specific table structures, comments, associated constraints, and data values.

User Input:

- None
- Can be resorted by clicking on "OWNER", "TABLE NAME", or "TABLESPACE".

Output:

- A list of tables and associated owner, and tablespace.
- Embedded links on table names, tablespace names, constrain text, and data text.



GRADS TABLE STRUCTURES

Click on table name or tablespace name or constraint to bring up a new window with its definition.

OWNER	TABLE NAME	TABLESPACE	CONSTRAINT	VIEW
GRADSNEW	EXAMINFO	LOOKUP	CONSTRAINT	DATA
GRADSNEW	ACADSTANDING	LOOKUP	CONSTRAINT	DATA
GRADSNEW	ADMITSTATUS	LOOKUP	CONSTRAINT	DATA
GRADSNEW	CURRENTSTATUS	LOOKUP	CONSTRAINT	DATA
GRADSNEW	GRAD_UNITS_REQ	LOOKUP	CONSTRAINT	DATA
GRADSNEW	GRADES	LOOKUP	CONSTRAINT	DATA
GRADSNEW	ETHNICSORGN	LOOKUP	CONSTRAINT	DATA
GRADSNEW	TERMS	LOOKUP	CONSTRAINT	DATA
GRADSNEW	COUNTRIES	LOOKUP	CONSTRAINT	DATA
GRADSNEW	YEARS	LOOKUP	CONSTRAINT	DATA
GRADSNEW	COURSETYPES	LOOKUP	CONSTRAINT	DATA
GRADSNEW	COURSES	LOOKUP	CONSTRAINT	DATA
GRADSNEW	PREQ_CONSTRAIN	LOOKUP	CONSTRAINT	DATA
GRADSNEW	CY_CORE	LOOKUP	CONSTRAINT	DATA
GRADSNEW	CY_PREQ	LOOKUP	CONSTRAINT	DATA
GRADSNEW	QUARTER	LOOKUP	CONSTRAINT	DATA
GRADSNEW	ROLES	LOOKUP	CONSTRAINT	DATA
GRADSNEW	GRADSTUDENTS	GRADSNEW	CONSTRAINT	DATA
GRADSNEW	MAJOR	GRADSNEW	CONSTRAINT	DATA
GRADSNEW	PREVEDUCATION	GRADSNEW	CONSTRAINT	DATA

User Input:

- Click on the embedded link of proceeding table name

Output:

- Table structure and comments for proceeding table

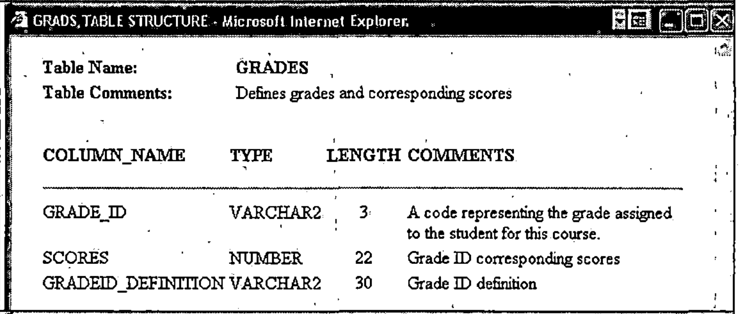


Table Name: GRADES

Table Comments: Defines grades and corresponding scores

COLUMN_NAME	TYPE	LENGTH	COMMENTS
GRADE_ID	VARCHAR2	3	A code representing the grade assigned to the student for this course.
SCORES	NUMBER	22	Grade ID corresponding scores
GRADEID_DEFINITION	VARCHAR2	30	Grade ID definition

User Input:

- Click on the embedded link of proceeding data text

Output:

- Data values for proceeding table

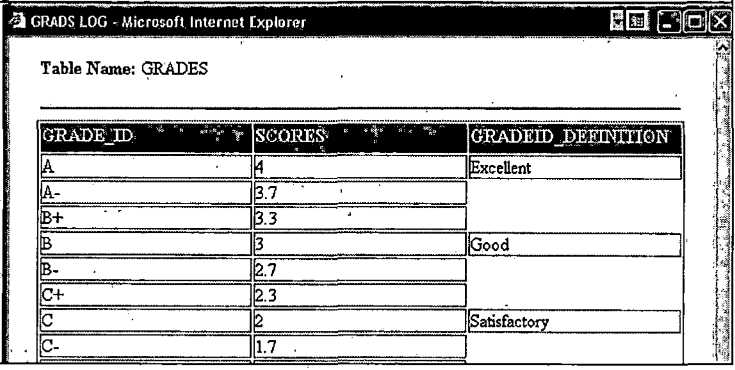


Table Name: GRADES

GRADE_ID	SCORES	GRADEID_DEFINITION
A	4	Excellent
A-	3.7	
B+	3.3	
B	3	Good
B-	2.7	
C+	2.3	
C	2	Satisfactory
C-	1.7	

Figure 32. Table Structures Page

4.23 Database Administrator/Developer: View Procedural Language/Structured Query Language Program

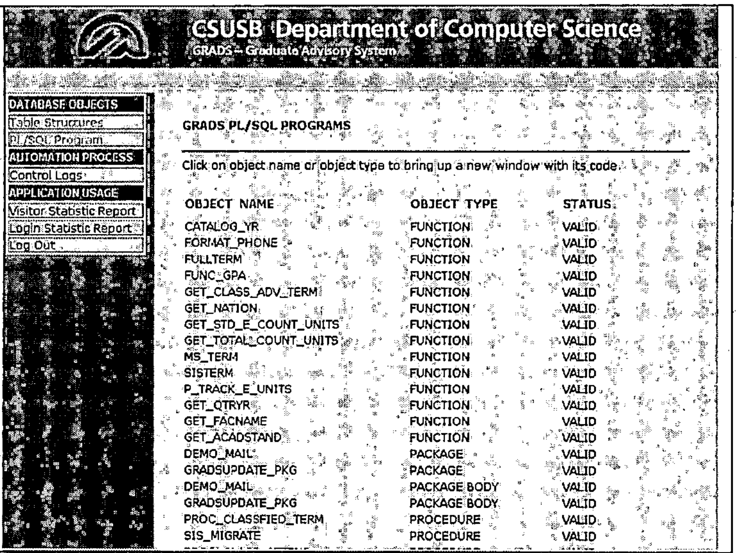
This set of functions allows the DBA/Developer to monitor the status (valid/invalid) and code on system triggers, functions, and procedures.

User Input:

- None
- Can be resorted by clicking on "OBJECT NAME", "OBJECT TYPE", or "STATUS".

Output:

- A list of PL/SQL programs and associated type and status.
- Embedded links on object name.



GRADS PL/SQL PROGRAMS

Click on object name or object type to bring up a new window with its code.

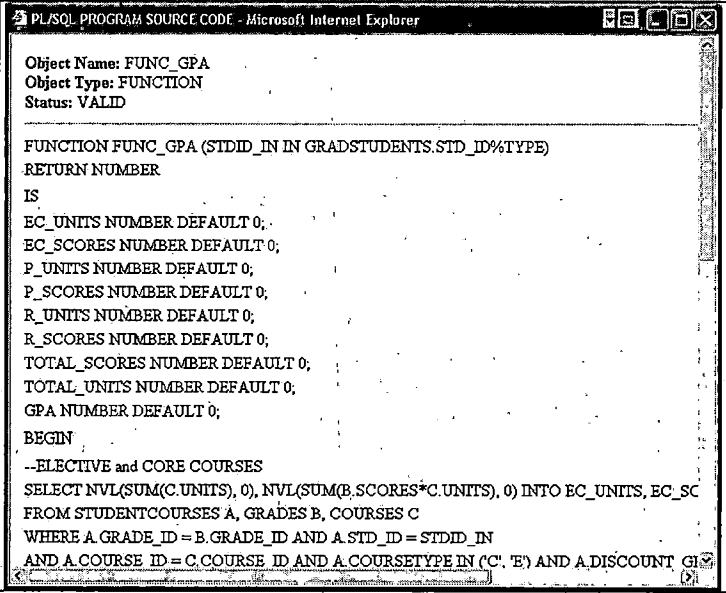
OBJECT NAME	OBJECT TYPE	STATUS
CATALOG_YR	FUNCTION	VALID
FORMAT_PHONE	FUNCTION	VALID
FULLTERM	FUNCTION	VALID
FUNC_GPA	FUNCTION	VALID
GET_CLASS_ADV_TERM	FUNCTION	VALID
GET_NATION	FUNCTION	VALID
GET_STD_E_COUNT_UNITS	FUNCTION	VALID
GET_TOTAL_COUNT_UNITS	FUNCTION	VALID
MS_TERM	FUNCTION	VALID
SISTERM	FUNCTION	VALID
P_TRACK_E_UNITS	FUNCTION	VALID
GET_QTRYR	FUNCTION	VALID
GET_FACNAME	FUNCTION	VALID
GET_ACADSTAND	FUNCTION	VALID
DEMO_MAIL	PACKAGE	VALID
GRADSUPDATE_PKG	PACKAGE	VALID
DEMO_MAIL	PACKAGE BODY	VALID
GRADSUPDATE_PKG	PACKAGE BODY	VALID
PROC_CLASSFIED_TERM	PROCEDURE	VALID
SIS_MIGRATE	PROCEDURE	VALID

User Input:

- Click on the embedded link of proceeding object name

Output:

- Program code for proceeding object



Object Name: FUNC_GPA
Object Type: FUNCTION
Status: VALID

```
FUNCTION FUNC_GPA (STDID_IN IN GRADSTUDENTS.STD_ID%TYPE)
RETURN NUMBER
IS
EC_UNITS NUMBER DEFAULT 0;
EC_SCORES NUMBER DEFAULT 0;
P_UNITS NUMBER DEFAULT 0;
P_SCORES NUMBER DEFAULT 0;
R_UNITS NUMBER DEFAULT 0;
R_SCORES NUMBER DEFAULT 0;
TOTAL_SCORES NUMBER DEFAULT 0;
TOTAL_UNITS NUMBER DEFAULT 0;
GPA NUMBER DEFAULT 0;
BEGIN
--ELECTIVE and CORE COURSES
SELECT NVL(SUM(C.UNITS), 0), NVL(SUM(B.SCORES*C.UNITS), 0) INTO EC_UNITS, EC_SC
FROM STUDENTCOURSES A, GRADES B, COURSES C
WHERE A.GRADE_ID = B.GRADE_ID AND A.STD_ID = STDID_IN
AND A.COURSE_ID = C.COURSE_ID AND A.COURSETYPE IN ('C', 'E') AND A.DISCOUNT GI
```

Figure 33. Procedural Language/Structured Query Language Program Page

4.24 Database Administrator/Developer: Visitor Statistic Report

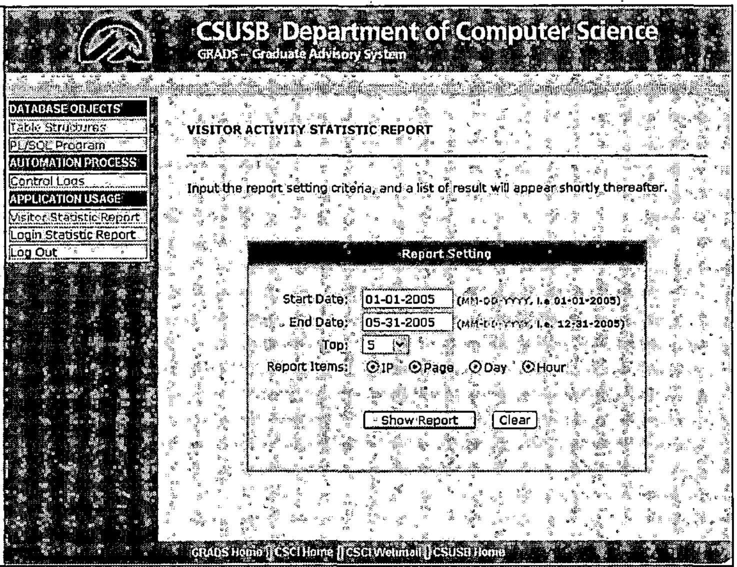
This set of functions provides the DBA/Developer an ad-hoc statistics report on visitors' activities, such as visited pages, day and time statistics.

User Input:

- Report timeframe setting
- Interested report items (visitor's IP, or visited pages, day, or hour)

Output:

- Statistic report



CSUSB Department of Computer Science
GRADS – Graduate Advisory System

VISITOR ACTIVITY STATISTIC REPORT

Input the report setting criteria, and a list of result will appear shortly thereafter.

Report Setting

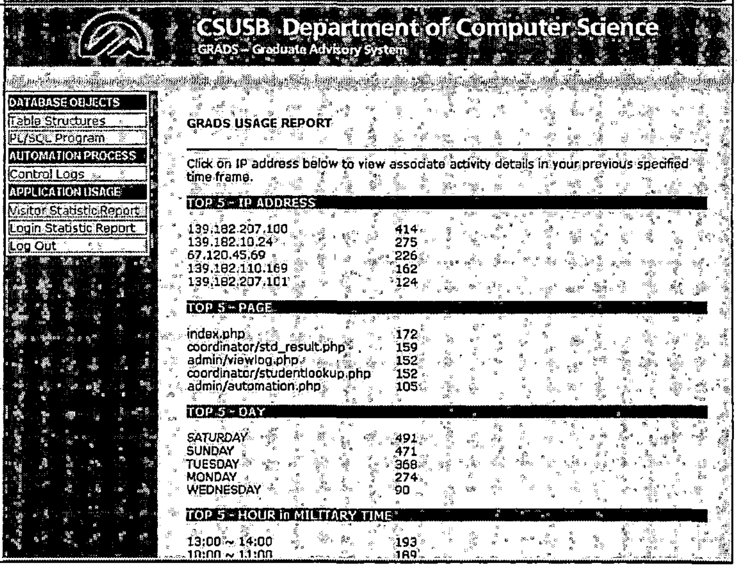
Start Date: 01-01-2005 (MM-DD-YYYY, i.e 01-01-2005)
End Date: 05-31-2005 (MM-DD-YYYY, i.e. 12-31-2005)
Top: 5
Report Items: ● IP ● Page ● Day ● Hour

User Input:

- Embedded link to query the log for specified IP address

Output:

- Top-N analysis report for proceeding items



CSUSB Department of Computer Science
GRADS – Graduate Advisory System

GRADS USAGE REPORT

Click on IP address below to view associate activity details in your previous specified time frame.

```
TOP 5 - IP ADDRESS

139.182.207.100             414
139.182.10.24               275
67.120.45.69                226
139.182.110.169             162
139.182.207.101             124

TOP 5 - PAGE

index.php                   172
coordinator/std_result.php  159
admin/viewlog.php           152
coordinator/studentlookup.php 152
admin/automation.php        105

TOP 5 - DAY

SATURDAY                    491
SUNDAY                      471
TUESDAY                     368
MONDAY                      274
WEDNESDAY                   90

TOP 5 - HOUR in MILITARY TIME

13:00 ~ 14:00               193
10:00 ~ 11:00               189
```

Figure 34. Visitor Statistic Report Page

4.25 System Logout

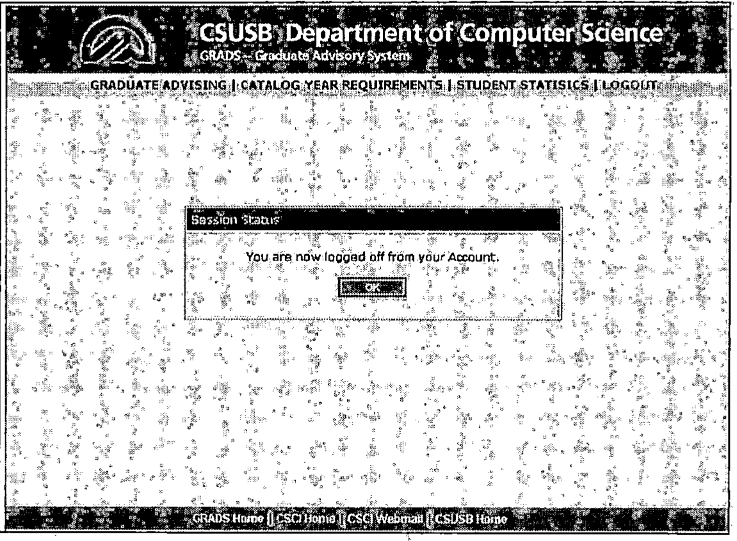
This function allows the system to close the user session.

User Input:

- Clicking on top-navigation-bar's Logout hyperlink

Output:

- Processing result message on screen
- Session closed



CSUSB Department of Computer Science
GRADS -- Graduate Advisory System
GRADUATE ADVISING | CATALOG YEAR REQUIREMENTS | STUDENT STATISICS | LOGOUT
Session Status

You are now logged off from your Account.

OK
GRADS Home || CSCI Home || CSCI Webmail || CSUSB Home

Figure 35. Logout Page

CHAPTER FIVE

DEPLOYMENT

This chapter provides a guide for deploying the web application and database. This guide details needed procedures to assist the departmental system administrator and Oracle database administrator to deploy.

5.1 System Requirements

This project needs two major components: database server and application server. The database server runs Oracle 9i Database Enterprise Edition while the application server runs Apache 1.3 as the web server and PHP 4.3 with OCI8 extension as the application engine.

5.2 Installation

5.2.1 Application Server

Process Overview

This process will change the configuration on the application server to run the current application. We assume that the ORACLE_HOME is /opt/oracle/ora92.

Process Steps

1. Create a shell script file called startgrads* and place it in the /usr/local/apache/bin/.

The script file is shown below. (NOTE: You may have to change the ORACLE_SID below based off of Oracle SID where the DB is located).

```
#! /bin/sh
ORACLE_HOME=/opt/oracle/ora92
ORACLE_SID=Your_Oracle_DB_SID
LD_LIBRARY_PATH=${ORACLE_HOME}/lib
export ORACLE_HOME ORACLE_SID LD_LIBRARY_PATH
#export > /tmp/envvars

# Using following line to start Apache
automatically with ssl
# You will need SSL configured prior to starting
it
/usr/local/apache/bin/apachectl startssl

# Uncomment the following line if you want to
start Appache w/o ssl
#/usr/local/apache/bin/apachectl start
```

2. Place the following lines at the end of the rc.local file located in the /etc directory. (Note that /usr/local/apache/bin/startgrads is the location of the script that was created in step 2)

is pointing to the correct server and ORACLE_SID. Also make sure that the Oracle username and password are correct for read/write permissions. The mail server also needs to be configured in this file.

5.2.2 Database Server

Process Overview

This process will create needed tablespaces, user acoounts, and tables in the database server.

Process Steps

5. Install Oracle by refering to Chapter 3 of the Oracle 9i Installation Guide.
6. Login as sys to run create_ts.sql to create needed tablespaces.
7. Run create_acct.sql to create role and user accounts.
8. Logout, then login as the new user you created in step 3.
9. Run lookup_create.sql to create lookup tables and comments.
10. Run grads_creat.sql to create data tables and comments.
11. Run automation.sql to create intermediate and log tracking tables and comments.

5.2.3 Database Migration Process

The following section lists processes that need to be performed by database developers to deploy the Oracle production environment.

5.2.3.1 Automate the Information Update Process.

Table 10. Information Update Process

Description	The following procedure details the process needed for scheduling the jobs to automate the data migration process in the Oracle database server.
Requirements	UNIX Account on development Oracle server. Oracle Account on development Oracle server with EXECUTE privileges on GRADSNEW procedures.
Supported OS Version	RHEL 3.0
Supported Oracle Version	Oracle9i Enterprise Edition
Notes	This process automates the extraction file download and batch-load process. If the database structure changes, then the process may need to be modified.

5.2.3.1.1 Schedule Extraction File Download and Batch-load.

Process Overview

This process will create UNIX crontab job to download extraction file to Oracle server, and then invoke Oracle SQL*Loader to load data into the Oracle database.

Process Steps

1. Create a new directry under /home/oracle/ directory and grant read, write, and execute permission on this new directory.
2. Copy the expect script, data loading control file, and load.sh to the newly created directory.
3. Use UNIX crontab utility to schedule extraction file download and batch-load task.
   - Type in command line crontab -e to place you in your default editor and load the current crontab file

   Example: 30 12 * * * /home/oracle/new_dir/load.sh

5.2.3.1.2 Schedule Oracle Jobs.

Process Overview

This process uses DBMS_JOB package to schedule a recurring job to invoke SIS+ data migrating procedure, GPA calculating procedure, and log procedure.

Process Steps

1. Schedule Oracle job to execute SIS+ data migrating task

```
DECLARE
   m_job number;
BEGIN
   dbms_Job.Submit(
   job        => m_job,
   what       => ‘SIS_MIGRATE;’,
   next_date  => trunc(sysdate)+1+3/24,
   interval   => ‘trunc(sysdate)+1+3/24’);
END;
/
```

2. Schedule Oracle job to invoke GPA procedure

```
DECLARE
   m_job number;
BEGIN
   dbms_Job.Submit(
   job        => m_job,
   what       => ‘PROC_GPA;’,
   next_date  => trunc(sysdate)+1+21/24,
   interval   => ‘trunc(sysdate)+1+21/24’);
END;
/
```

3. Schedule Oracle job to clean logs

```
DECLARE
   m_job number;
BEGIN
   dbms_Job.Submit(
   job        => m_job,
   what       => ‘PROC_NUKELOG;’,
    next_date => trunc(sysdate)+1+4/24,
    interval  => 'trunc(sysdate)+1+4/24');
END;
/
```

5.2.3.2 Microsoft Access Database Migration Phase I.

Table 11. Access Database Migration Specification I

Description	The following procedure details the migrating process from departmental Microsoft Access database to intermediate staging area of Oracle database.
Requirements	Oracle Account on development Oracle server with DBA privileges.
Supported MS Access Version	Microsoft Access 2000
Supported Oracle Version	Oracle9i Enterprise Edition Release 9.2.0.4.0
Supported Oracle Workbench Version	Oracle Migration Workbench Release 10.1.0.2.0
Notes	This process migrates the departmental Access database to development Oracle database. If the Access database structure changes, then the process may need to be modified.
Special Instructions	Download omwb.zip & MSACccess.jar from Oracle OTN, then place MSAccess.jar in omwb/Omwb/plugins before migration.

5.2.3.2.1 Accessing the Destination Database.

Process Overview

Create a user with the minimum required privileges required to migrate a database using Migration.

Process Steps

1. Create a new user:

   ```
   CREATE USER OMWB_USER IDENTIFIED BY
   omwb_user_password;
   ```

2. Grant role to this new user:

   ```
   GRANT CONNECT, RESOURCE, CREATE PUBLIC SYNONYM TO
   OMWB_USER WITH ADMIN OPTION;
   ```

3. Grant system priviledge to this new user:

   ```
   GRANT ALTER ANY ROLE, ALTER ANY SEQUENCE, ALTER
   ANY TABLE, ALTER TABLESPACE, ALTER ANY TRIGGER,
   COMMENT ANY TABLE, CREATE ANY SEQUENCE, CREATE ANY
   TABLE, CREATE ANY TRIGGER, CREATE ROLE, CREATE
   TABLESPACE, CREATE USER, DROP ANY SEQUENCE, DROP
   ANY TABLE, DROP ANY TRIGGER, DROP TABLESPACE, DROP
   USER, DROP ANY ROLE, GRANT ANY ROLE, INSERT ANY
   TABLE, SELECT ANY TABLE, UPDATE ANY TABLE TO
   omwb_user;
   ```

5.2.3.2.2 Configure Departmental Microsoft Access Database for Migration.

Process Overview

This process will turn off security from Microsoft Access database by copying the contents of the secured database into a new database to prepare a ready-to-migrate copy of the Access database.

Process Steps

Back up of the Microsoft Access database files.

Ensure the following is installed on the same system (your desktop etc.) as Migration Workbench:

- Microsoft Access 2000
- Microsoft ODBC driver release 4.00.4403.00 or greater

3. Turn off security by copying the contents of the secured database into a new database, as follows:

   a. From the File menu in Microsoft Access, select New Database.

   b. Select the Blank Database icon, then click OK.

   c. From the File menu within the new database, select Get External Data, then select Import.

   d. Select the back up of the Microsoft Access database files that you want to import, then click Import.

   e. A screen appears prompting you to enter database password. Type in your password for accessing the back up of MDB file then click OK.

   f. From the Import Objects dialog, click Options.

g. Select Definition and Data options.

h. From the Tables tab, choose MS Student or any table that you want to migrate.

i. Click OK.

5.2.3.2.3 Creating Microsoft Access XML Files

Process Overview

This process will use Oracle Migration Workbench Exporter for Microsoft Access to export departmental Microsoft Access database to an XML file.

Process Steps

1. Open the appropriate MDE file in Microsoft Access: Open the following file: OMWB_install_dir\Omwb\msaccess_exporter\omwb2000.mde An OMWB Exporter for MS Access screen appears prompting you to specify the Microsoft Access database file that you want to migrate to Oracle.
2. Click Export Database Schema.
3. In the MDB File Name field, enter the path of MDB file that you would like to migrate. If necessary, choose Browse to locate the Microsoft Access database.
4. In the XML File Name field, change the path and file name of the XML file, if necessary.

5. Click Export. The Microsoft Access database you selected is exported to an XML file.

6. After export database schema, go back to OMWB Exporter for MS Access screen then click Export Table Data and repeat steps 3 through 5 to export data. The Microsoft Access database data you selected is exported to DAT file.

5.2.3.2.4 Starting Migration Workbench on Windows.

Process Overview.

The process will start Oracle Migration Workbench and create a Repository account for Capture wizard to capture source database.

Process Steps

1. From the Start option, select Run.

   The Run dialog box appears.

2. Run the startup script by typing a command similar to the following:

   ```
   OMWB_install_dir\Omwb\bin\omwb.bat
   ```

3. In the previous command, OMWB_install_dir is the absolute directory where you installed Migration Workbench.

   The Oracle Migration Workbench Repository Login dialog box appears.

4. Click OK.

5. Select the Workbench repository that you want to use for the migration.

   Before you can capture the Source Model, Migration Workbench prompts you to specify the Workbench repository you want to use:

   - Select an Oracle database (Oracle Database 10g, Oracle9i, or Oracle8i), enter the user name, password, host, port, and SID for that database, then click OK.
   - You are prompted to create a specific user account on the database you want to use as the Workbench repository. Click OK.
   - The Select Migration Source screen appears.

6. Select the plug-in that you want to use from the list of installed plug-ins, then click OK.

   The Capture wizard appears.

5.2.3.2.5 Using the Capture Wizard

Process Overview

The process will use Oracle Workbench Capture Wizard to capture the source database to Workbench Repository.

Process Steps

1. To start the Capture wizard, from the Action menu, select Capture Source Database.

The Welcome page appears.

2. Click Next.

   The Select Databases to Capture page appears.

3. Select the database to capture as follows:

   Click Add XML File.

   A file selection dialog box appears.

   - Select the XML file that you created, then click OK.
   - Click Next.

   The Data Type Mapping page appears.

4. Review the data type mappings, change default memo datatype map from CLOB to VARCHAR2(4000) to save system resource since current memo datatype fields are no more than 500 charters.

   Click Next.

   The Create Oracle Model page appears.

5. Choose whether to automatically create the Oracle Model:

   Choose Yes.

6. Click Next.

   The Destination Database page appears. Choose the type of Oracle database options Oracle8i/Oracle9i/Oracle Database 10g.

7. Click Next. The Summary page appears. Review the summary information.

8. Click Finish to start the capture.

   A progress screen appears displaying information about the capture in progress.

   - If an error message appears, fix the error and start the Capture wizard again.
   - If a warning message appears, consider the problem described in the message and decide whether any action is required.

9. Click OK to dismiss the progress screen.

   The Creating Oracle Model screen appears displaying the total number of errors and warnings.

10. Click OK to return to the main Migration Workbench screen.

5.2.3.2.6 Using the Migration Wizard

Process Overview

The process will use Oracle Workbench Migration Wizard to migrate the source database to the destination Oracle database.

Process Steps

1. From the Action menu, select Migrate to Oracle to start the Migration wizard.

The Welcome page appears displaying the steps required to migrate the database.

2. Click Next.

The Destination Database Details page appears.

3. Enter the information required to connect to the destination database:

   - In the Username field, enter the newly created user omwb_user .
   - In the Password field, enter the password for the user.
   - Enter the appropriate values in the Host, Port, and SID fields.

4. Click Next. The Migrate Tablespaces, Users, and User Tables page appears.

5. Review the information about the Migrate Tablespaces, Users, and User Tables page, then click Next. The Migrate Table Data page appears.

6. Specify whether you want to transfer the data during this Migration wizard session. Choose Yes to transfer the data now.

7. Click Next. The Select Schema Objects to Migrate page appears.

8. Select the schema objects to migrate, then click Next. The Summary page appears.

9. Review the summary information:

   - If you need to change a selection, click Back to return to a previous page.
   - Click Finish to start the migration.

10. A progress screen appears, displaying information about the migration progress.

11. Review the messages that appear on the progress screen, and take the following action:

    - If an error message appears, you must fix the error and return to step 1.
    - If a warning message appears, consider the problem outlined in the message and decide whether any action is required.

    After the migration is completed, a screen appears displaying the total number of a errors and warnings.

5.2.3.3 Microsoft Access database Migration Phase II.

Table 12. Access Database Migration Specification II

Description	The following procedure details the migrating process to migrate data from the intermediate staging area to the newly designed Oracle database.
Requirements	Oracle Account on development Oracle server with all privileges in gradsnew and db3 schemas.
Supported Oracle Database Version	Oracle9i Enterprise Edition Release 9.2.0.4.0

Notes	This process migrates data from intermediate tables in staging area to the newly designed database in the development Oracle database. If the departmental Access database structure or data value definition changes from a point-in-time, then the PL/SQL programs may need to be revised.

5.2.3.4 Execute PL/SQL Programs.

Process Overview

This process migrates data from the intermediate table MS_STUDENT to data tables GRADSTUDENTS, COMMITTEE, PROJECT, THESIS, COMEXAM, EXAM, and PROBATION. If any migrating transaction fails, the failed transaction data will be written to STDINFO_EXCP, MASTEROPTION_EXCP, TOEFL_EXCP, COMMITTEE_EXCP, and PROB_EXCP.

Process Steps

1. Create a temp table named STAFF_TEMP for staff ID mapping purpose.

   ```
   CREATE TABLE STAFF_TEMP (STAFF_ID CHAR(9),
   NAME VARCHAR2(35), LAST VARCHAR2(30), FIRST
   VARCHAR2(30),
   MIDDLE VARCHAR2(15), ISFACULTY CHAR(1));
   ```

2. Insert data into STAFF_TEMP table by selecting "other" advisor/committee members from the intermediate table MS_STUDENT which was created by Oracle Workbench in phase I migration process.

```
INSERT INTO STAFF_TEMP (NAME) SELECT DISTINCT
ADVISOROTHER FROM MS_STUDENT;
INSERT INTO STAFF_TEMP (NAME) SELECT DISTINCT
OTHERMEMBER1 FROM MS_STUDENT;
INSERT INTO STAFF_TEMP (NAME) SELECT DISTINCT
OTHERMEMBER2 FROM MS_STUDENT;
DELETE FROM STAFF_TEMP WHERE NAME IS NULL;
```

3. Run procedure to generate staff ID for “other” advisor/committee members in STAFF_TEMP table.

```
EXECUTE PROC_STAFF_TEMP;
```

4. Copy the new advisor/committee members from STAFF_TEMP table to STAFF.

```
INSERT INTO STAFF (STAFF_ID, ISFACULTY) SELECT
STAFF_ID, ISFACULTY FROM STAFF_TEMP;
```

5. Run procedure to migrate data from intermediate tables in staging area to the newly designed database.

```
EXECUTE PROC_MSDB_MIGRATE;
```

6. Validate the migrated data in both data tables and exception handling tables.

CHAPTER SIX

CONCLUSIONS

6.1 Project Accomplishments

Interfacing our department database with the SIS+ helps capture a more complete set of information through coverage of additional data attributes. For example, previously, only courses students had already taken were available; today, data includes students' current registered/dropped courses information. The new interface also shortens the data updating cycle from a quarterly basis to an one-week period. It further increases the overall value of data. While out-of-date data may aid with historical aspects, it fails to assist greatly with further planning. Finally, the new interface system reduces the general workload in regard to manual data entry for the Program Assistant.

The useful applications provided by this new system are also numerous. In general, it enhances the service quality of communications for both prospective and current students. The system, specifically, provides advice to prospective students concerning necessary coursework and related prerequisites. Moreover, it collects prospective students' relevant survey information and prior academic background. Analyzing survey data aids the recruitment strategy effort.

Additionally, the system assists continuing students with study plans and ongoing accomplishment report. Thus, faculty can more readily review student performance in term of grades, project/thesis topics, and committee information. Efficiency is greatly improved for both continuing students and project/thesis advisor.

The system also aids the Program Coordinator in tracking the status and overall performance of graduate students in Computer Science. It reduces the need for paper storage by Graduate Coordinator.

6.2 Personal Skills Enhancement and Learning

Overall, I have learned a great deal from having participated in this project. This has included the learning of Oracle technologies used in designing, creating, and maintaining Oracle database. I learned to utilize Oracle Designer in designing and documenting a database. During the implementation phase, I learned to write database programs using Oracle PL/SQL programming language. This permits the embedding of crucial business rules within Oracle database.

In preparation for deploying the system, I learned to function as an efficient database administrator and to tune the database itself within the UNIX server environment. As a result, wait time is now reduced, user productivity is improved, and database performance is maximized.

Undertaking this project allowed me to obtain hands-on practice with Oracle products. Thereby, I received a deeper understanding of Oracle, and was aided in obtaining certification on Oracle. This certification validated my knowledge and understanding of Oracle products and skills necessary for a successful project outcome.

Broadly speaking, integration of SIS+ with Oracle enabled a linking, coordination, and expansion of available data between an original mainframe and a web-enabled application system. I learned how to coordinate and implement the coordination for best results, covering advisors, coordinators, students, and concerned faculty. My capability and skills immeasurably increased in the process.

6.3 Impact of Common Management System Project

In next twenty-two months, the SIS+ will be converted to new PeopleSoft Student Administration 8.9. Impacts of the conversion to the GRADS system should be minimums. The current GRADS system receives data from SIS+ using extract file in text format. It should continue the same process when new student-record-system is in production. During the CMS migration, some of translations in SIS+ will be changed (such as term code, country code, etc…). The GRADS database should apply the same changes to reflect new PeopleSoft SA 8.9. The cross-reference table for changes in SIS+ can be obtained from the CMS project office or the Office of Record, Registration, and Evaluation.

6.4 Future Directions

Results from this project have included the creation of a relational database and storing academic information for all students in the Computer Science graduate program. Interestingly, this project is the first web-enabled advising system among all graduate programs on the CSUSB campus, though undergraduates have used DARS/DARwin (the client/server version of Degree Audit Reporting System) for several years to perform transfer articulation, advising and certification. Hopefully, the outcome of this project may prompt some graduate departments to reconsider building advising rules in DARS/DARwin.

Furthermore, this application may be revised to directly interface with Common Management System HCM applications in the future, providing more precise information in real-time for Computer Science graduate students. It is my hope that, having worked with this new system, I will have assisted in encouraging additional applications at CSUSB, and in helping generates a focus upon future research and analysis. Thus providing a potential for further expansion of technology and rapid, efficient data access. With expended research comes greater knowledge, and with greater knowledge, broader applications.

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.

6. CURRENTSTATUS. Defines current status of students -- Active, Incoming, Graduated, Inactive Attended, On Leave, Dismissed, Never Attended, Academic Probation, Withdraw.

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

CUR_STATUS [VARCHAR2 (30)]: Description of current status of student -- Active, Incoming, Graduated, Inactive Attended, On Leave, Dismissed, Never Attended, Academic Probation, Withdraw.

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 perquisite 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_LOOKUP. Store 4 digits year value for application drop-down list reference.

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

A.2 Data Tables

26. 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

27. 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

28. 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)

29. 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. 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

31. 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
RES_ADDRS	[VARCHAR2 (100)]: Street address where the student is residing
CITY	[VARCHAR2 (20)]: City
STATE	[CHAR (2)]: State abbreviation
ZIP	[VARCHAR2 (9)]: Zip code
COUNTRY	[VARCHAR2 (50)]: Country name
PERM_ADDRS	[VARCHAR2 (200)]: Whole address extracted from SIS+
ADMIT_TERM	[VARCHAR2 (5)]: Term code which represents admitted quarter and year
ADMIT_STATUS	[VARCHAR2 (10), FK]: Admission status which has referential integrity constraint with ACADSTANDING table
ACAD_STAND	[VARCHAR2 (10), FK]: Academic standing status which has referential integrity constraint with ACADSTANDING table
CUR_STATUS	[VARCHAR2 (10), FK]: Current Status which has referential integrity constraint with CURRENTSTATUS table

CUR_START [VARCHAR2 (5)]: Term code which represents the year and quarter the current status started

CUR_END [VARCHAR2 (5)]: Term code which represents the year and quarter when current status will end

TERM_LASTATD [VARCHAR2 (5)]: Term code which represents the last year and quarter attended

CLASSIFIED_TERM [VARCHAR (5)]: Term code which represents classified quarter and year

CANDIDACY_TERM [VARCHAR (5)]: Term code which represents the year and quarters that student is advanced to candidacy

TOEFL_WAIVED [CHAR (1)]: Y if the TOEFL test requirement has been waived

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

MASTER_OPTION [CHAR (1)]: Type of graduation track option – P for project, T thesis, C for comprehensive exam.

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

32. 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.

33. MAJOR. Defines student's undergraduate education major

MAJOR_ID	[VARCHAR2 (4), PK]: Undergraduate education major identification number
MAJOR_NAME	[VARCHAR2 (30)]: Description of major

34. 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.

35. 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

36. 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

37. SIS_CLASS_ATTEND.

38. 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

39. 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.

40. 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.

41. 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

42. THESIS. Store master thesis track information.

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

TITLE [VARCHAR2 (300)]: Thesis title

PRESENT_DATE [DATE]: Thesis presentation date

THESIS_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

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

USER_ID [VARCHAR2 (10), PK]: User identification number

PWORD [VARCHAR2 (15)]: Password for authentication

ROLE_ID [CHAR (1), FK]: User's role which has referential integrity constraint with ROLES table.

STAFF_STD_ID [CHAR (9)]: Staff identification number of staff or student identification number of Student

ISSTAFF [CHAR (1)]: S if the user is a staff member

FIRSTPAGE [NUMBER (3)]: User's customized home page

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

44. 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

45. 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

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

AUTOPROC_IDSEQ	[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

47. 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

LOG_DATE [DATE]: Modification date

48. 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

49. 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

50. 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

51. 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

52. NUKELOG_CTLTB. Store PLSQL automation associated update logs keep days.

LOGNAME [VARCHAR2 (30), PK]: Log name

KEEPDAY [VARCHAR2 (5)]: Days log keeps

53. 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

54. 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
COL15	[VARCHAR2 (50)]: Generic column
COL16	[VARCHAR2 (10)]: Generic column
COL17	[VARCHAR2 (10)]: Generic column
COL18	[VARCHAR2 (10)]: Generic column
COL19	[VARCHAR2 (10)]: Generic column
COL20	[VARCHAR2 (10)]: Generic column
COL21	[VARCHAR2 (10)]: Generic column
COL22	[VARCHAR2 (10)]: Generic column
COL22	[VARCHAR2 (10)]: Generic column

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

STD_ID	[CHAR (9)]: Student Identification Number

COURSE_ID	[VARCHAR2 (15)]: Course Identification Number
COURSE_TERM	[VARCHAR2 (15)]: Course taken term
GRADE_ID	[VARCHAR2 (15)]: Letter grade
UNITS	[VARCHAR2 (15)]: Course units
LOG_DATE	[DATE]: Data fail loading date

56. STDCOURSES_NOSTORE. A log table stores student's non-stored course taken information.

STD_ID	[CHAR (9)]: Student Identification Number
COURSE_ID	[VARCHAR2 (15)]: Course Identification Number
COURSE_TERM	[VARCHAR2 (15)]: Course taken term
GRADE_ID	[VARCHAR2 (15)]: Letter grade
UNITS	[VARCHAR2 (15)]: Course units
LOG_DATE	[DATE]: Data loading date

57. 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_ADDRS [VARCHAR2 (200)]: Address extracted from SIS+, includes street address, city, state, zip

COUNTRY [VARCHAR2 (50)]: Country, part of PERM_ADDRS

ADMIT_TERM [VARCHAR2 (5)]: Student admits term

58. 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

59. 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

60. VISIT_LOG. Store application visitor's activity information

VISIT_SEQ [VARCHAR2 (8)]: A sequence number which generates from VISITLOG_SEQ to distinct each row

VISIT_DATETIME [DATE]: Visitor's visiting date and time

VISITOR_IP [VARCHAR (15)]: Visitor's IP address

PAGE_NAME [VARCHAR2 (100)]: Visited page (file) name

61. USER_LOG. Store application usage information

LOGIN_SEQ [VARCHAR2 (8)]: A sequence number which generates from USERLOG_SEQ to distinct each row

USERNAME [VARCHAR2 (10)]: User's login username

LOGIN_DATETIME [DATE]: User's login date and time

LOGIN_IP [VARCHAR (15)]: user's IP address

IS_SUCCESS [VARCHAR (1)]: A flag to indicate user's login process whether success

APPENDIX B

APPLICATION PROGRAM FILES

B.1 Login

Function Name	LOGIN

Associate File	include ("objects/login.obj");
	include("gradsHeader.php");
	include ("gradsFooter.php");

Processing Flow	/index.html ⇒ /_login.php ... → /student/student_main.php ... → /coordinator/coordinator_main.php ... → /admin/admin_main.php ... → errormsg.php

B.2 Password Recovery

Function Name	PASSWORD RECOVERY

Associate File	include ("objects/pwdLookUp.obj");
	include("funcs/mail_funcs.php");
	include("gradsHeader.php");
	include ("gradsFooter.php");

Processing Flow	/forget.php ⇒ /_forget.php

B.3 Grads Access Request

Function Name	GRADS ACCESS REQUEST

Associate File	include("funcs/mail_funcs.php");
	include("gradsHeader.php");
	include ("gradsFooter.php");

Processing Flow	/register.php ⇒ /_register.php

B.4 Logout

Function Name	LOGOUT
Associate File	/student/ logout.php /coordinator/ logout.php

B.5 Prospective Student: Survey and Evaluation

Function Name	STUDENT SURVEY AND EVALUATION

Associate File	include (“inqStudentObject.php”);
	include(“gradsHeader.php”);
	include (“gradsFooter.php”);
	include (“mail_report.php”);

Processing Flow	/inquireStudent/studSurvey.php ⇒ /inquireStudent/courseCheckList.php ⇒ /inquireStudent/courseInfo.php ⇒ /inquireStudent/courseEval.php

B.6 Prospective Student: Course Catalog

Function Name	VIEW COURSE INFORMATION
Associate File	include(“../objects/course.obj”) ; include(“gradsHeader.php”); include “gradsFooter.php”;
Processing Flow	/inquireStudent/courses.php ⇒ /inquireStudent/list_courses.php ⇒ /inquireStudent/view_course.php

B.7 Continuing Student: Current Status

Function Name	VIEW CURRENT STATUS

Associate File	include(“student.php”);
	include(“gradsHeader.php”);
	include (“gradsFooter.php”);

Processing Flow	student/coursetaken.php

B.8 Continuing Student: STUDY PLAN

Function Name	STUDY PLAN

Associate File	include (“student.php”);
	include(“gradsHeader.php”);
	include (“gradsFooter.php”);

Processing Flow	/student/studyplan.php ... → /student/_unitupdate.php ... → / pop_course.php

B.9 Continuing Student: COURSE CATALOG

Function Name	VIEW COURSE CATALOG

Associate File	include (“../objects/course.obj”) ;
	include(“gradsHeader.php”);
	include (“gradsFooter.php”);

Processing Flow	/student/courses.php /student/list_courses.php ⇒ /student/view_course.php

B.10 Continuing Student: MASTER OPTION

Function Name	VIEW MASTER OPTION

Associate File	include “student.php”;
	include(“gradsHeader.php”);
	include (“gradsFooter.php”);
	include (“../funcs/mail_funcs.php”);

Processing Flow	/student/masteroption.php … → /student/master_update.php ⇒ /student/master_update.php ⇒ /student/_mupdate.php

B.11 Continuing Student: ACCOUNT INFO

Function Name	STUDENT GRADS ACCOUNT INFO

Associate File	include (“student.php”);
	include (“../funcs/mail_funcs.php”);
	include(“gradsHeader.php”);
	include (“gradsFooter.php”);
	include (“mail_report.php”);

Processing Flow	/student/pinfo.php … → /student/info_update.php ⇒ /student/_pupdate.php

B.12 Continuing Student: Current Status

Function Name	VIEW STUDENT CURRENT STATUS

Associate File	include ("../student/student.php");
	include("advisingfuns.php");
	include("gradsHeader.php");
	include ("gradsFooter.php");

Processing Flow	student/coursetaken.php

B.13 Coordinator: Graduate Advising Main

Function Name	STUDENT PERFORMANCE LOOKUP

Associate File	include ("coordObject.php") ;
	include("gradsHeader.php");
	Include ("gradsFooter.php");

Processing Flow	/coordinator/studentlookup.php ⇒ /coordinator/std_result.php ⇒ /coordinator/advising_main.php

B.14 Coordinator: Study Plan

Function Name	VIEW STUDENT STUDY PLAN

Associate File	include ("../student/student.php");
	include("advisingfuns.php");
	include("gradsHeader.php");
	include ("gradsFooter.php");

Processing Flow	/coordinator/std_studyplan.php

B.15 Coordinator: Master Option

Function Name	VIEW STUDENT MASTER OPTION

Associate File	include ("../student/student.php");
	include("advisingfuns.php");
	include("gradsHeader.php");
	include ("gradsFooter.php");

Processing Flow	/coordinator/std_masteroption.php

B.16 Coordinator: Contact Info

Function Name	VIEW STUDENT CONTACT INFO

Associate File	include ("../student/student.php");
	include("advisingfuns.php");
	include("gradsHeader.php");
	include ("gradsFooter.php");

Processing Flow	/coordinator/advising_main.php

B.17 Coordinator: Add / Waive Prerequisites

Function Name	ADD/WAIVE STUDENT PREREQUISITES

Associate File	include ("coordObject.php") ;
	include("advisingfuns.php");
	include("gradsHeader.php");
	include ("gradsFooter.php");

Processing Flow	/coordinator/add_p.php ⇒ /coordinator/_addp.php

B.18 Coordinator: Add / Delete Recommends

Function Name	ADD/DELETE STUDENT RECOMMEND COURSES

Associate File	include ("coordObject.php") ;
	include("advisingfuns.php");
	include("gradsHeader.php");
	include ("gradsFooter.php");

Processing Flow	/coordinator/add_recommend.php ⇒ /coordinator/_addrecommend.php

B.19 Coordinator: CSCI Graduate Advising Sheet

Function Name	GRADUATE ADVISING SHEET

Associate File	include ("coordObject.php") ;
	include("advisingfuns.php");
	include("gradsHeader.php");
	include ("gradsFooter.php");

Processing Flow	/coordinator/advisingsheet.php /coordinator/print_advisingsheet.php ...→ /coordinator/add_note.php ⇒ /coordinator/_addnote.php

B.20 Coordinator: Preliminary Advancement to Candidacy Form

Function Name	PRELIMINARY ADVANCEMENT to CANDIDACY FORM

Associate File	include ("../student/student.php");
	include ("coordObject.php") ;
	include("advisingfuns.php");
	include("gradsHeader.php");
	include ("gradsFooter.php");

Processing Flow	/coordinator/preliminary.php /coordinator/preliminary_2.php ⇒ /coordinator/printpre.php

B.21 Coordinator: Advancement to Candidacy Form

Function Name	ADVANCEMENT to CANDIDACY FORM

Associate File	include ("../student/student.php");
	include ("coordObject.php") ;
	include("advisingfuns.php");
	include("gradsHeader.php");
	include ("gradsFooter.php");

Processing Flow	/coordinator/adv_candidacy.php ⇒ /coordinator/print_adv.php → /coordinator/ add_comment.php ⇒ /coordinator/_addcomment.php

B.22 Coordinator: Core Courses Requirements

Function Name	ADD/DELETE CATALOG YEAR CORE COURSES REQUIREMENTS

Associate File	include ("catalogYearObj.php") ;
	include("catafuns.php");
	include("gradsHeader.php");
	include ("gradsFooter.php");

Processing Flow	/coordinator/cat_corelookup.php ⇒ /coordinator/cat_core.php ⇒ /coordinator/_catcore.php

B.23 Coordinator: Prerequisites Requirements

Function Name	ADD/DELETE CATALOG YEAR PREREQUISITE COURSES REQUIREMENTS

Associate File	include ("catalogYearObj.php") ;
	include("catafuns.php");
	include("gradsHeader.php");
	include ("gradsFooter.php");

Processing Flow	/coordinator/cat_preqlookup.php ⇒ /coordinator/cat_preq.php ⇒ /coordinator/_catpreq.php

B.24 Coordinator: Prerequisites Constraints

Function Name	ADD/DELETE PREREQUISITES COURSE CONSTRAINTS

Associate File	include ("catalogYearObj.php") ;
	include("catafuns.php");
	include("gradsHeader.php");
	include ("gradsFooter.php");

Processing Flow	/coordinator/preq_constrains.php ⇒ /coordinator/ _preqconstrains.php

B.25 Coordinator: View Core Courses Requirement

Function Name	CATALOG YEAR CORE COURSES REQUIREMENTS LOOK UP

Associate File	include ("catalogYearObj.php") ;
	include("catafuns.php");
	include("gradsHeader.php");
	include ("gradsFooter.php");

Processing Flow	/coordinator/corelookup.php ⇒ /coordinator/corelist.php

B.26 Coordinator: View Prerequisite Courses Requirement

Function Name	CATALOG YEAR PREREQUISITE COURSES REQUIREMENTS LOOK UP

Associate File	include ("catalogYearObj.php") ;
	include("catafuns.php");
	include("gradsHeader.php");
	include ("gradsFooter.php");

Processing Flow	/coordinator/preqlookup.php ⇒ /coordinator/preqlist.php

B.27 Coordinator: View Prerequisite Constraints

Function Name	PREREQUISITES COURSE CONSTRAINTS LOOK UP

Associate File	include (“catalogYearObj.php”) ;
	include(“catafuns.php”);
	include(“gradsHeader.php”);
	include (“gradsFooter.php”);

Processing Flow	/coordinator/constrainlist.php

B.28 Coordinator: Add Degree Units Requirement

Function Name	ADD NEW CATALOG YEAR DEGREE UNITS REQUIREMENT

Associate File	include (“catalogYearObj.php”) ;
	include(“catafuns.php”);
	include(“gradsHeader.php”);
	include (“gradsFooter.php”);

Processing Flow	/coordinator/add_require.php ⇒ /coordinator/_addunitreq.php

B.29 Coordinator: Delete Degree Units Requirement

Function Name	VIEW/DELETE DEGREE UNITS REQUIREMENT

Associate File	include (“catalogYearObj.php”) ;
	include(“catafuns.php”);
	include(“gradsHeader.php”);
	include (“gradsFooter.php”);

Processing Flow	/coordinator/unitlist.php ⇒ /coordinator/delete_units.php

B.30 Coordinator: Prospective Students' Statistic Chart

Function Name	INQUIRING STUDENT SURVEY: STASTISTIC CHART

Associate File	include("inqReport.obj");
	include_once("bar_graph.php");

Processing Flow	/coordinator/timeframe_chart.php ⇒ /coordinator/eva_chart.php

B.31 Coordinator: Prospective Students' Survey Summary

Function Name	INQUIRING STUDENTS SURVEY: SUMMARY REPORT

Associate File	include("inqReport.obj");
	include("evafuns.php");
	include("gradsHeader.php");
	include ("gradsFooter.php");

Processing Flow	/coordinator/timeframe_chart.php ⇒ /coordinator/eva_summary.php ⇒ /coordinator/print_evasum.php

B.32 Coordinator: Prospective Students' Gender & Country Summary

Function Name	INQUIRING STUDENTS SURVEY: GENDER & COUNTRY SUMMARY REPORT

Associate File	include("inqReport.obj");
	include("evafuns.php");
	include("gradsHeader.php");
	include ("gradsFooter.php");

Processing Flow	/coordinator/timeframe_inqcty.php ⇒ /coordinator/country_sum_inq.php ⇒ /coordinator/print_countrysuminq.php

B.33 Coordinator: Prospective Students' Contact Information

Function Name	INQUIRING STUDENTS SURVEY: CONTACT INFORMATION REPORT

Associate File	include("inqReport.obj");
	include("evafuns.php");
	include("gradsHeader.php");
	include ("gradsFooter.php");

Processing Flow	/coordinator/timeframe_cont.php ⇒ /coordinator/eva_cont.php ⇒ /coordinator/print_evacont.php

B.34 Coordinator: Continuing Students' Gender & Country Summary

Function Name	CONTINUING STUDENTS GENDER & COUNTRY SUMMARY

Associate File	include("inqReport.obj");
	include("evafuns.php");
	include("gradsHeader.php");
	include ("gradsFooter.php");

Processing Flow	/coordinator/timeframe_country.php ⇒ /coordinator/country_summary.php ⇒ /coordinator/ print_countrysum.php

B.35 Coordinator: Continuing Students' Gender & Ethnic Region Summary

Function Name	CONTINUING STUDENTS GENDER & ETHNIC REGION SUMMARY

Associate File	include("inqReport.obj");
	include("evafuns.php");
	include("gradsHeader.php");
	include ("gradsFooter.php");

Processing Flow	/coordinator/timeframe_ethnic.php ⇒ /coordinator/ethnic_summary.php ⇒ /coordinator/ print_ethnicsum.php

B.36 Database Administrator/Developer: Control Logs

Function Name	CONTROL LOGS

Associate File	Include ("../objects/visitlog.obj");
	Include ("gradslog.obj");
	Include ("adminfuns.php");
	Include ("gradsHeader.php");
	Include ("gradsFooter.php");

Processing Flow	/admin/automation.php ...→ /admin/_addNukelog.php ...→ /admin/editlog.php ⇒ /admin/_editlog.php ...→ /admin/viewlog.php ⇒ /admin/_viewlog.php

B.37 Database Administrator/Developer: Table Structures

Function Name	VIEW TABLE STRUCTURE

Associate File	Include ("../objects/visitlog.obj");
	include ("gradstable.obj") ;
	Include ("adminfuns.php");
	Include ("gradsHeader.php");
	Include ("gradsFooter.php");

Processing Flow	/admin/admin_table.php ...→ /admin/table.php ...→ /admin/tablespace.php ...→ /admin/constraint.php ...→ /admin/viewTable.php

B.38 Database Administrator/Developer: PL/SQL Program

Function Name	VIEW PL/SQL PROGRAM

Associate File	Include ("gradsFooter.php");
	Include ("../objects/visitlog.obj");
	Include ("gradsplsql.obj") ;
	Include ("adminfuns.php");
	Include ("gradsHeader.php");
	Include ("gradsFooter.php");

Processing Flow	/admin/admin_plsql.php ⇒ /admin/source.php

B.39 Database Administrator/Developer: Visitor Statistic Report

Function Name	VIEW VISITOR STATISTIC REPORT

Associate File	Include ("../objects/visitlog.obj");
	Include ("gradslog.obj");
	Include ("adminfuns.php");
	Include ("gradsHeader.php");
	Include ("gradsFooter.php");

Processing Flow	/admin/admin/timeframe_visitor.php ⇒ /admin/visitorlog.php ⇒ /admin/visitip_detail.php

B.40 Database Administrator/Developer: Login Statistic Report

Function Name	VIEW LOGIN STATISTIC REPORT

Associate File	Include ("../objects/visitlog.obj");
	Include ("gradslog.obj");
	Include ("adminfuns.php");
	Include ("gradsHeader.php");
	Include ("gradsFooter.php");

Processing Flow	/admin/timeframe_user.php ⇒ /admin/userlog.php ⇒ /admin/userip_detail.php

REFERENCES

[1] "Apache Jakarta-Tomcat." Goodwill. Apress, 2002.

[2] Bakken and Auldbach. PHP Manual http://www.php.net/manual/en/

[3] Burleson, Donald "Oracle 9i Unix Administration Handbook", Oracle Press, 2002.

[4] Converse, Tim and Park, Joyce "PHP Bible 2nd Edition", Wiley, 2002.

[5] Chiang and Rotte "GRADS Advising Component" Independent Study Project, Summer 2004.

[6] Dawes, Chip And Thomas Biju "Introduction to Oracle 9i SQL", Sybex, 2002.

[7] Feurstein, Steven and Pribyl, Bill "Oracle PL/SQL", O'Reilly, 2002.

[8] Heinle and Pena "Designing with JavaScroipt" Second Edition.. O'Reilly, 2001

[9] IEEE Std. 830-1998 IEEE Recommend Practice of Software Requirements Specifications.

[10] Koletzke, Peter and Dorsey, Paul "Oracle Designer Handbook 2nd Edition", Oracle Press, 1999.

[11] Kyte, Thomas "Expert One-on-One Oracle", Wrox Press, 2001.

[12] Lerdorf and Tatroe "Programming PHP". O'Reilly, 2002.

[13] Libes, Don "Exploring Expect", O'Reilly, 1996.

[14] Loney, Kevin and Theriault, Marlene "Oracle 9i DBA Handbook", Oracle Press, 2002.

[15] Software Requirement Specification for Class, Group 2, in CSCI655 class at CSUSB, Fall 2003.

[16] Stuns, Doug and Weishan, Matthew "Oracle 9i DBA Fundamentals II", Sybex, 2002.

[17] Thomas, Biju and Bryla, Bob "Oracle 9i DBA Fundamentals I", Sybex, 2002.

[18] Vu "The Web-Based Database Management System for the Computer Science Graduate program" CSUSB MSCS Graduate Project, Winter 2004.

[19] Wainwright. Wrox "Professional Apache.", 1999.