5-2022

College of Education FileMaker Extraction and End-User Database Development

Andrew Tran

COLLEGE OF EDUCATION LEGACY FILEMAKER EXTRACTION AND END-USER DATABASE DEVELOPMENT

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

Andrew Tran

May 2022

COLLEGE OF EDUCATION LEGACY FILEMAKER EXTRACTION AND END-USER DATABASE DEVELOPMENT

A Project

Presented to the

Faculty of

California State University,

San Bernardino

---

by

Andrew Tran

May 2022

Approved by:

Dr. Haiyan Qiao, Committee Chair, Computer Science

Dr. Ernesto Gomez, Committee Member

Dr. Marita Mahoney, Committee Member

Mrs. Yen-Hsi Chiang, Committee Member

ABSTRACT

The College of Education (CoE) at the California State University San Bernardino (CSUSB) developed a system to keep track of both state and national accreditation requirements using FileMaker 5, a database system. This accreditation data is crucial for reporting and record-keeping for the CSU Chancellor's Office as well as the State of California. However, the database system was developed several decades ago and software support has long since been dropped, causing the CoE's legacy accreditation data to be at risk of being lost should the software or hardware suffer permanent failure. The purpose of this project was to perform extraction of the raw data in this legacy system, develop a new database to clean and house this raw data, as well as develop a newer and more efficient end-user interface for the CoE to retrieve the data. Raw data analysis, consultation, and extraction are performed first and are followed by database design and development using Oracle SQL and PL/SQL. At the same time, the end-user interface and reporting tool was developed using Oracle PeopleSoft. This project aimed to extract critical data from the outdated and no longer supported legacy database while providing the CoE with an efficient means to access the data for reporting needs.

ACKNOWLEDGEMENTS

First, I would like to acknowledge Mrs. Tiffany Chiang for her constant support and mentorship. Her invaluable efforts and guidance enabled me to have the foundation necessary to analyze, develop, and implement this project. Her mentorship ignited a drive for learning which has led me to where I am today.

I would also like to acknowledge Dr. Marita Mahoney for cheerful cooperation and mentorship as well. Without her hard work and willingness to assist, the project may have taken much longer to complete. Her expertise in the education field was also invaluable in documentation and completion of this manuscript.

I would like to extend gratitude to Dr. Haiyan Qiao and Dr. Ernesto Gomez for their guidance through the project and their insightful comments and suggestions. Their openness to help guided me through much of my career as a graduate student and I sincerely appreciate that.

Finally, I would like to thank Mr. Linh Tran, ACBI's Database Administrator. With his help and feedback, I was able to receive permissions to work on this project. His assistance with the technical components of the project ultimately led to the project's completion.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 Background

In the past, the College of Education (CoE) at the California State University, San Bernardino (CSUSB) needed a single solution to keep track of both state and national accreditation requirements. These requirements included a way to perform student tracking and progress monitoring, usually from a student's application until their program completion. The CoE also needed a method for data collection reporting for Strategic Planning, which the department would be able to use and reference in order to make "evidence-based" decisions.

Eventually, the CoE settled on designing databases in order to keep track of the data for their needs and requirements. Initially, the databases were both numerous and independent, with much of it simply being housed on individual desktops. Eventually, the CoE decided to compile all those databases into shared relational database systems. These database systems were developed using FileMaker 5's and FileMaker 6's End-User Desktop Software along with FileMaker 5's Server Software. All this information was essentially stored using a one server system, and it has been around since 1999.

When Dr. Marita Mahoney came on-board the CoE in 2006, the same system from 1999 was still in use. Despite considerable resources and time, efforts to develop a new system with third party vendors on two occasions was unsuccessful as neither vendor was able to produce a solution that could meet

the CoE's needs. However, in the Winter Quarter of 2016, the CoE was able to collaborate with CSUSB's Administrative Computing and Business Intelligence (ACBI), the Information Technology Services (ITS) department, and the Institutional Research and Analytics (IR) department in order to build a customized system using Oracle PeopleSoft. This system was dubbed the Admissions Module and it went live three quarters later in fall of 2016, with additional modules and sequential updates continuing since that time.

1.2 Significance

After the development of the customized PeopleSoft system project, the legacy FileMaker system has since been archived. This legacy system is now read-only, and its contents can only be accessed by personnel while they are on the CSUSB campus. However, when the project was moved to the PeopleSoft system, it only allowed the College of Education to enter data for current students starting from 2016, the year the PeopleSoft system was implemented and introduced. Initially, the CoE was able to get by and simply referenced the legacy FileMaker system for the archived data should it be needed; however, a new problem arose. As the FileMaker system was built and continued to use software from 1999, the system along with all the critical archived data being stored on it, was at risk to fail at any moment. This brought up a serious dilemma for the CoE as per California State Regulations, the CoE must retain all records for teacher credentialing and licensing information, including the archived "old" records from the FileMaker system, and they must have this information be always available

for retrieval. In other words, the database system was developed several decades ago on software whose support has long since been dropped. This caused a time-sensitivity issue for the CoE's legacy accreditation data to be at risk of being lost should the database system eventually permanently fail. Thus, the significance of this project is to develop a way to help the CoE migrate all their legacy FileMaker data onto a newly developed and stable database system. This data will then be crucial in reporting to the Chancellor's office as well as in keeping records for the State of California.

This project serves as an opportunity to perform data cleanup and promote data consistency on this legacy data. The FileMaker's reporting interface is relatively cluttered and complex, and Dr. Mahoney has requested that the new reporting interface be made simpler. This was done through data analysis and consultation. Doing this allowed for determining what data is strictly necessary for reporting, what data was already available in records from other departments, and what data were simply no longer needed. This analysis allowed for identifying priority key fields, allowing for production of a cleaner end-user reporting interface. Following data analysis, an opportunity to perform data clean-up and check data consistency can be realized. Prior to the creation of the FileMaker system, data was manually entered into individual database systems and data entry standards varied. Following the completion of the FileMaker system up until the archival in 2016, data entry standards became stricter causing data entry to become more uniform throughout the years. However,

since data was still largely entered manually, this still left room for human error. Small errors like typos, spacing, and formatting issues are relatively common and are simple to correct. Accuracy and verification are needed to correctly identify students. If a record is missing a Student ID, then it needed to be manually referenced in the complete FileMaker data in order to ensure the student's information is consistent and accurate. If a student's ID is not of the correct length, the same method had to be performed to verify consistency as well. This process held true for every other key field, or fields that are used to uniquely identify a student. By doing this, the data migration process is guaranteed to be consistent with the CoE's standards and needs.

1.3 Purpose

The purpose of this project was to extract and export all priority identified FileMaker legacy key data onto a newly developed Oracle database, one that is both easier to use and access, while keeping the integrity of the data being transferred. This project was developed at the request of Dr. Marita Mahoney of the College of Education with CSUSB ACBI and ITS providing resources and support. Due to the nature and sensitivity of the information being handled, Mrs. Tiffany Chiang and her team from CSUSB ACBI and ITS supervised the creation and development of this new Oracle Database. Prior to data extraction from the legacy system, data analysis and consultation were done with Dr. Mahoney in order to clearly identify the critical data needed for migration. Extraction and exportation of this legacy data helped ensure its longevity as well. All of this

along with the data cleanup will help comply with California State regulations. In addition to data migration and development of the database, a new reporting interface was also developed, allowing end-users to more quickly and easily retrieve data when compared to the FileMaker system. Training, resources and support for data analysis and extraction form the FileMaker system was provided by Dr. Mahoney. Training, resources and support for the end-user reporting interface was provided by Mrs. Tiffany Chiang, which was developed using Oracle's PeopleSoft Application Designer. Finally, the database system itself was developed using Oracle SQL and PL/SQL.

1.4 Organization of the Project Document

There are seven chapters in this document: (1) Introduction, (2) Software Requirements Specifications, (3) Database Design and Implementation, (4) PeopleSoft End-User Reporting Interface (5) Application Implementation, (6) Deployment, and (7) Conclusion. The appendices that follow these chapters provide technical details on the system's design, implementation, and methods used.

CHAPTER TWO

SOFTWARE REQUIREMENTS SPECIFICATION

2.1 Project Scope

This project will consist of three main phases: (1) data analysis and extraction; (2) database development and data cleanup; and (3) end-user reporting interface development.

The data analysis and extraction phase focus on working with the College of Education to identify what data needs to be migrated. As the FileMaker legacy database is quite large, a discussion will be made on what key fields from student records need to be extracted and exported to the new system.

The database development and data cleanup phase progress through distinct three stages. 1) *Import Stage*, used to import the raw extractions to the new Oracle database from the legacy system: 2) *Cleanup Stage*, used to correct typos and formatting issues and data correction for any faulty key fields: and 3) *Ready Stage*, used to house the cleaned data following agreement of data quality between client and developer.

The end-user reporting interface phase focuses on developing a user-friendly reporting interface for the client using Oracle PeopleSoft. This interface will connect to the Ready Stage table from the database in order to provide reporting services for Dr. Mahoney and her team at the College of Education.

2.2 Product Perspective

2.2.1 Use-Case Diagram

Figure 1 demonstrates the possible actor interaction with the new system. The College of Education will have access to surface level functions such as searching and exporting data. The developer and administrator will have the same actions as the College of Education with additional actions in system design, security, and code structure.

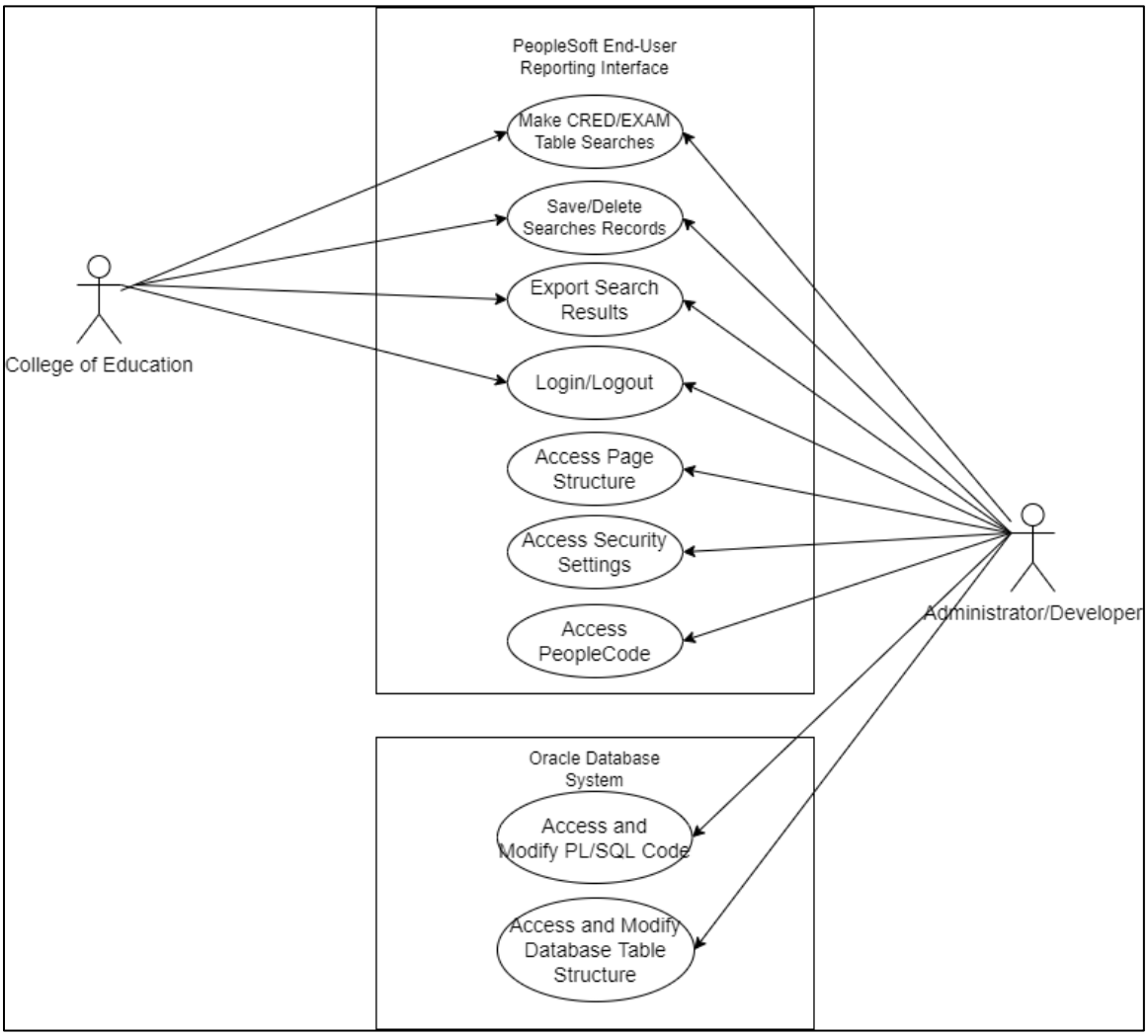


Figure 1. Use-Case Diagram for the College of Education Database Module

2.2.2 System Interfaces

Users will first need to connect to the Cisco AnyConnect virtual private network using the credentials provided by CSUSB's ITS department. Then, users will navigate through the College of Education's PeopleSoft Portal to reach the FileMaker archive project search page. Using the interface, users will input their search criteria, which will then send a request to the Oracle database. The database will process the search query and return the result. This will either redirect the user to a landing page or will print an error message stating that no records were found.

2.3 Users

For this project, there are two main classifications of users.

- College of Education: any authorized user from this department who needs to retrieve records for reporting to the State of California.
- Database Developer/Administrator: IT personnel responsible for development, maintenance, and administration over this module.

2.4 Product Functions

Each user will have the ability to login and logout of the system. Upon successful login, the user will be redirected to their respective PeopleSoft landing page. Similarly to the users, there are two main roles able to access this system.

Table 1. System Roles and Privileges

Role	Privileges
College of Education	• View records in the CRED and EXAM tables
Database Administrator/Developer	• View database structures • Access security settings • View stored programs

2.4.1 College of Education Component

This component will provide an authorized user from the College of Education with the following options:

- View a student's contact information
- View a student's Credential records
- View a student's Comprehensive Exam records
- Export each of the pages to a data file of their choice

2.4.2 Database Administrator/Developer Component

This component will provide a database administrator or developer with the following options:

- View and modify security settings for the system
- Access underlying page and table structures
- View and modify source code of PL/SQL programs
- Access database tables and fields
- Generate ad-hoc reports on table composition

CHAPTER THREE

DATABASE DESIGN AND DEVELOPMENT

3.1 Data Migration Introduction and Design Challenges

The term data migration can have a variety of meanings depending on the source. According to Morris, data migration is “the selection, preparation, extraction, transformation and permanent movement of appropriate data that is of the right quality … and the decommissioning of legacy data stores” [1]. As such, the first portion of this project deals with data migration, with the source being the College of Education’s legacy FileMaker system and the target is the new Oracle database server.

There are two main types of migration strategies: big bang and trickle. The big bang strategy involves a bulk migration from the source system to the target system with the goal of having a shorter completion time. The trickle method involves migrating data from the source to the target system in incremental batches with a drawback of requiring more complex design. [2]

For the purposes of this project, the trickle migration strategy was utilized in order to ensure data quality and consistency for the system migration process. Using the trickle method allowed for partitioning of the raw data into sets, analyzing the data, and generating a report back to the client for review. The ability to have constant feedback between developer and client ensured the migration process was as accurate as possible given the previous lack of standardization in data entry in the legacy database.

The partitioning of datasets also allowed for modular revision to the data clean-up source code. With this design, I was able to adjust the code and run quick test cases on the data if the client made any requests to how the legacy data needed to be transformed.

The database migration and data clean-up design occurred stepwise in sets of stages that act as tasks on how to approach the problem and address troubleshooting issues. The design loosely follows the general software development cycle as follows:

Table 2. Migration Development Cycles

Development Cycle	Deliverable(s)	Major Tasks
• Strategy	• Project Proposal	• Determining the purpose and scope of the project. • Identifying a database system to house data extractions. • Identifying cost-effective system for end-user reporting interface.
• Data Pre-Analysis	• Detailed Extraction and Database Schema Plan	• Considering which tables and data sets need to be extracted from FileMaker. • Devising a pipeline of stages for data import and data cleanup. • Generating a list of possible data types needed for each field. • Making uniform the field names from FileMaker on conversion

Development Cycle	Deliverable(s)	Major Tasks
		to Oracle fields.
• Data Analysis	• Data Migration Strategy Documentation	• Trimming data already readily available on other systems on campus. • Identifying key fields used for search indexing. • Identifying fields that need immediate cleanup. • Determining consistent and efficient data migration strategy.
• Design	• Documentation of Created Oracle Database Tables and Data Mapping	• Creating three-stage table structures for data import and data clean-up for each of the FileMaker table extractions. • Creating search indexes for query performance and record uniqueness. • Testing FileMaker extractions to match with database import tables for the Data Wizard.
• Development	• Documentation of PL/SQL Procedures and Packages Code	• Validate data batch count between FileMaker query and Oracle table imports. • Development of PL/SQL code for data import and data clean-up. • Unit and Integration testing for modular development of data transformation pipeline.
• Testing & Revisions	• Debugging Report and	• Creating reports on data import and data clean-up batches.

Development Cycle	Deliverable(s)	Major Tasks
	Documentation	• Identifying non-apparent fields that require specific clean-up modifications after consulting with client. • Identifying modifications to table structures needed for more efficient debugging and data quality checks. • Adjusting PL/SQL code based on testing and feedback from client.
• Implementation	• Finalized Data Tables for End-User Reporting Interface	• Migration of cleaned data batch to a production-ready table to prime for transferring to the PeopleSoft system. • Loading the new data in this table to PeopleSoft.
• Maintenance		• Validate scripts and script documentation to uphold coding standards. • Submit a finalized report for the count of total records and total unique students to the client. • Complete maintenance documentation for future projects and reference.

There were numerous challenges that came up during development of the data migration portion of this project. The first challenge was determining what data definitions were necessary for data migration. The process of determining

what to retain and what to omit took intensive review and analysis as well as approval from various members of the College of Education. This stage influenced the scope of the project as the number of data definitions to be migrated influences how each individual definition will be transformed and standardized. The original migration scope consisted of seven tables and was eventually reduced to two tables after data review (the CRED table and the COMP_EXAM table). A supplemental table for CRED called CRED_NOID was developed later to account for students without a CSUSB student ID.

The second challenge that came up was determining an effective migration strategy. Initially, the big bang strategy was employed in hopes of extracting all the necessary data at once and the performing analysis and clean-up within the new database system. However, it was quickly discovered that using such a strategy opened a large probability for data errors and data inconsistencies while making it incredibly difficult to back-reference the data in FileMaker should any issues with data quality arise. After realizing this, a trickle migration strategy was adopted with the data sets being partitioned year by year. In doing so, I was able to quickly identify any issues within each data set and have the data issues be cleared and verified in FileMaker.

The third challenge dealt with the data clean-up and data standardization process. Constant communication between client and developer is important in any software development process to ensure both parties are equally informed. As such, there was consistent and frequent communication between me and the

client on the final product. In turn, I would identify what was and was not possible within Oracle PL/SQL and we would plan accordingly. An additional challenge was to meet the client's specifications and balance these with Oracle PL/SQL parameters with the client's satisfaction and the end-result. This challenge was addressed through constant communication, review, and feedback.

In regard to data migration, Koletzke [3] states that "[u]sually the scope and difficulty of data migration is impossible to assess until the migration itself is almost complete". This was the case for this project as the exact data migration strategy was not finalized until well after the PL/SQL procedures and packages were completed. Fortunately, any revisions or modifications to the data migration process only required slight modifications to the underlying PL/SQL code.

3.2 FileMaker Student Information Data Extraction

The FileMaker extraction covers two data definitions, 48 core attributes per student, and a total of 35,495 rows of data. The data extracted was grouped as follows:

- Data Definitions
- Students' Credential Information
- Students' Credential Evaluation Data
- Subject Matter Program (SMP) and Exam Information
- Teaching Experience History
- Students' Comprehension Exam Information

- Comprehension Exam Question Results

The table below [4] provides a description of each of the data fields extracted from FileMaker for the scope of this project.

Table 3. FileMaker Extraction Fields and Description

Extraction Field	Description
ID_New	• Student ID Number
LastName	• Student's Last Name
FirstName	• Student's First Name
CrObjective	• The credential program the applicant is enrolled in
CredentialObjective	• The California credential the applicant is being evaluated for
Crprogoption	• The content and subject area that the California credential applicant is being evaluated for
CrClass	• The data element supplements the term or classification of the credential for which the student has been processed. • The student is then assigned a credential class based on the credential that he or she is being recommended for and the total number of requirements that have been met for that credential

Extraction Field	Description
DS_SubjArea	• Designated-Subjects Subject Area. • For Designated-Subjects, Credential, Credential Content, and Subject Area data
CRSUPPLEMENTALAUTH	• Additional areas of credential content and subject area(s) to be added to the primary credential identified in Crprogoption
BCLAD_EMPHA	• Additional authorizations for: English Learner Authorization (CLAD) or instruction in student's native language (BCLAD)
LANGUAGE	• The language for additional authorizations in BCLAD_EMPHA
PROGRAM	• California Commission on Teacher Credentialing (CTC) Program Standard for evaluated or issued credential
PAPER_APP	• CTC submission via a paper application
ONLINE_APP	• CTC submission via an online application
EvalDate	• Date of Evaluation for credential Requirements and Recommendation
CTCSubmission	• Date that the Credential Recommendation was submitted to the CTC
ProgCompDate	• Date that the student completed all program requirements to be recommended for their credential

Extraction Field	Description
CrIssuanceDate	• Date that the CTC issued the recommended credential to the student
Evaluator	• Name of the CSUSB Credential Analyst who evaluated the credential application submitted by the student
SMP_Program	• Subject Matter Program (SMP) • Content area for students who completed an approved undergraduate program instead of the California Subject Examinations for Teachers (CSET) to demonstrate subject matter competency
SMP_Institution	• Institution where the student completed an approved SMP
SSATAREA	• Date that the student passed all sections of the CSET to demonstrate subject matter competency. • This field is for students who did not complete an approved SMP
CBEST_Date	• Date that the student passed all sections of the California Basic Educational Skills Test (CBEST)
RICA_Date	• Date that the student passed all sections of the California Reading Instruction Competency Assessment (RICA)
TPA_Date	• Date that the student passed all sections of the California Teaching Performance Assessment (TPA)
DS_TEACHING_EXP	• Designated Subjects Teaching Experience • For Designated Subjects students and their history of teaching experience

Extraction Field	Description
REG_TEACHING_EXP	• List of teaching experiences and demonstration of teaching experience
ADM_TEACHING_EXP	• Administrative Services Credentials and demonstration of teaching experience
Work_Experience	• Designated Subjects Credentials • Demonstration of work-related experience to DS_SubjArea
POST-SECONDARY UNITS COURSEWORK	• Designated Subject Credentials • Demonstration of post-secondary courses completed
DSPrelimDate	• Date that the Designated Subjects Preliminary Credential was issued by the CTC
DSClearDate	• Date that the Designated Subjects Clear Credential was issued by the CTC
Internship	• Marks whether the student was issued an INTERN credential by the CTC
Preliminary	• Marks whether the student was issued a PRELIMINARY credential by the CTC
ProfessionalClear	• Marks whether the student was issued a CLEAR credential by the CTC
Graduate_Degree_Major	• The content or subject area of graduate degree that the student has previously received or is currently completing

Extraction Field	Description
Institution	• Institution at which the student has or is completing a graduate degree
Certificate_of_Eligibility	• Marks whether the student was issued a CERTIFICATE OF ELIGIBILITY by the CTC
PROGRAMOPTION	• The subject or discipline area of graduate degree that the student is seeking
ProgObjectStatus	• Marks whether the student seeking a graduate degree only or a graduate degree and a credential simultaneously
CompElDate	• Date that the student has met all program requirements to be eligible to take their comprehensive exam
CompInelDate	• Date of evaluation which determined if the student has not met all program requirements to take their comprehensive exam
CompExamDate	• Date the student passed all sections of their comprehensive exam
Compq1	• Comprehensive Exam Question 1 • Content Area that this student completed or attempted. • Subject and content varies across programs
Compq2	• Comprehensive Exam Question 2 • Content Area that this student completed or attempted. • Subject and content varies across programs

Extraction Field	Description
Compq3	• Comprehensive Exam Question 3 • Content Area that this student completed or attempted. • Subject and content varies across programs
Compq4	• Comprehensive Exam Question 4 • Content Area that this student completed or attempted. • Subject and content varies across programs
Compq5	• Comprehensive Exam Question 5 • Content Area that this student completed or attempted. • Subject and content varies across programs

3.3 Oracle SQL Developer Database Migration

3.3.1 Data Migration, Extraction, and Import Process

Data migration, especially on legacy data, can be both a risky and time consuming process. As FileMaker 5 is a legacy system, data extraction and migration is trickier than migrating between two modern database systems. The pipeline proposed to the College of Education was:

1) Begin data extraction for the most recent year's data set to the oldest
2) Extract the data from each year and export it to a raw (.CSV) file

3) From the (.CSV) file, convert to an (.XLSX) file to add in headers to simplify data field matching with an Oracle Database
4) Create a set of import tables in Oracle SQL to house the extracted data from FileMaker
5) Import the extractions using SQL Developer's Data Import Wizard into the new import tables

With FileMaker being an older system, it did not have the functionality of allowing for a link to an Oracle database system. So, an alternative solution was to select the necessary fields and extract the data manually to a (.CSV) file. It was also discussed that the extractions should be done year by year using the trickle migration method rather than all at once. For example, if an extraction's lower bound was September 1st XXXX, then the upper bound would be August 31st XXXX+1 [5]. This would partition the large amount of data into smaller datasets, allowing for easier initial data quality checking. If something were to go wrong during this early stage of the project, then the offending dataset extraction could be found by simple looking at a flag for the execution year. After the (.CSV) files are converted to (.XLSX), I used SQL Developer's Data Import Wizard to assist in importing the files into the appropriate import tables.

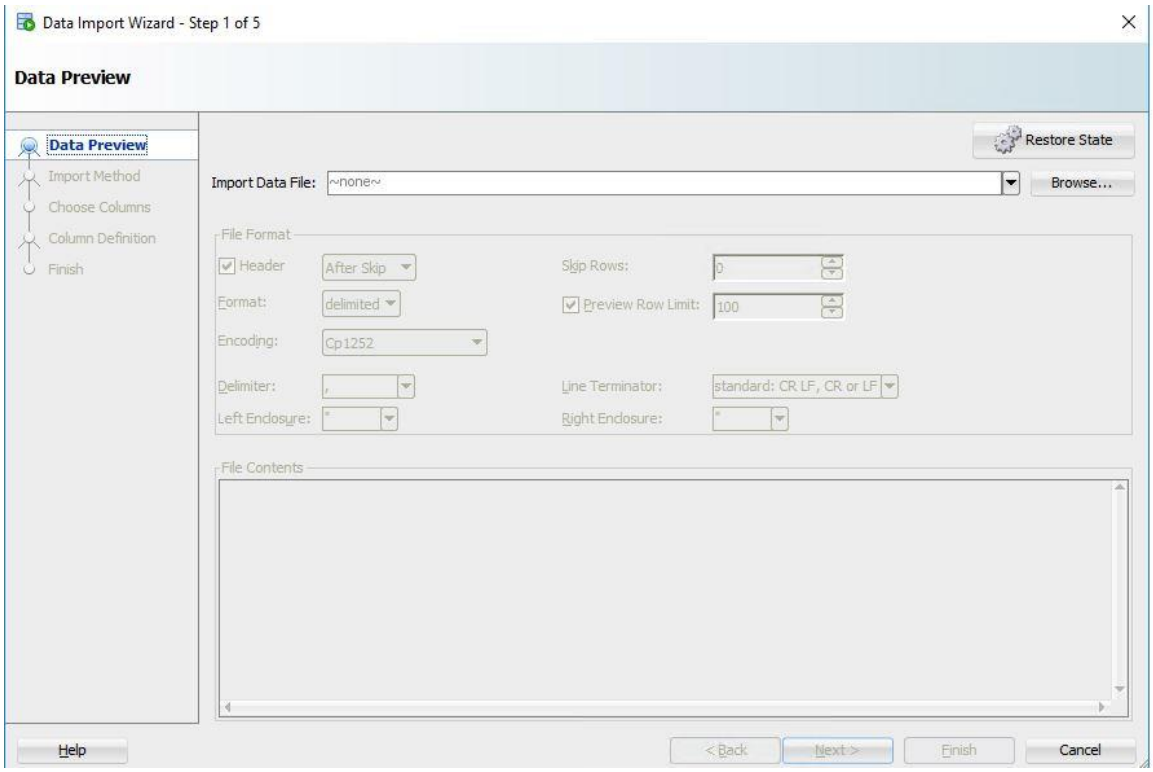


Figure 2. Data Import Wizard Interface

SQL Developer's Data Import Wizard is intuitive to use, allowing for easy one-to-one mapping of spreadsheet column names to the column names of the database's table. Figure 3 shows how column names are automatically matched if they are spelled the same. Otherwise, it would raise a warning, allowing for the developer to manually map columns that do not match up. If the data import is successful, the wizard will inform the user the data import is complete and the transaction was committed. Otherwise, it will prompt the user with an error, seen in Figure 4, and print a log of what insertions were successful and where the error occurred. Initially, these errors were quite common. The raw FileMaker data often had data with inconsistent spacing issues (e.g. like having blank spaces

that extended several thousand characters), and breaking out of the defined data types. The solution was to simply make the data types of the target table bigger whenever the problem came up. Doing this allowed for a much smoother transition between the legacy data and the initial import table.

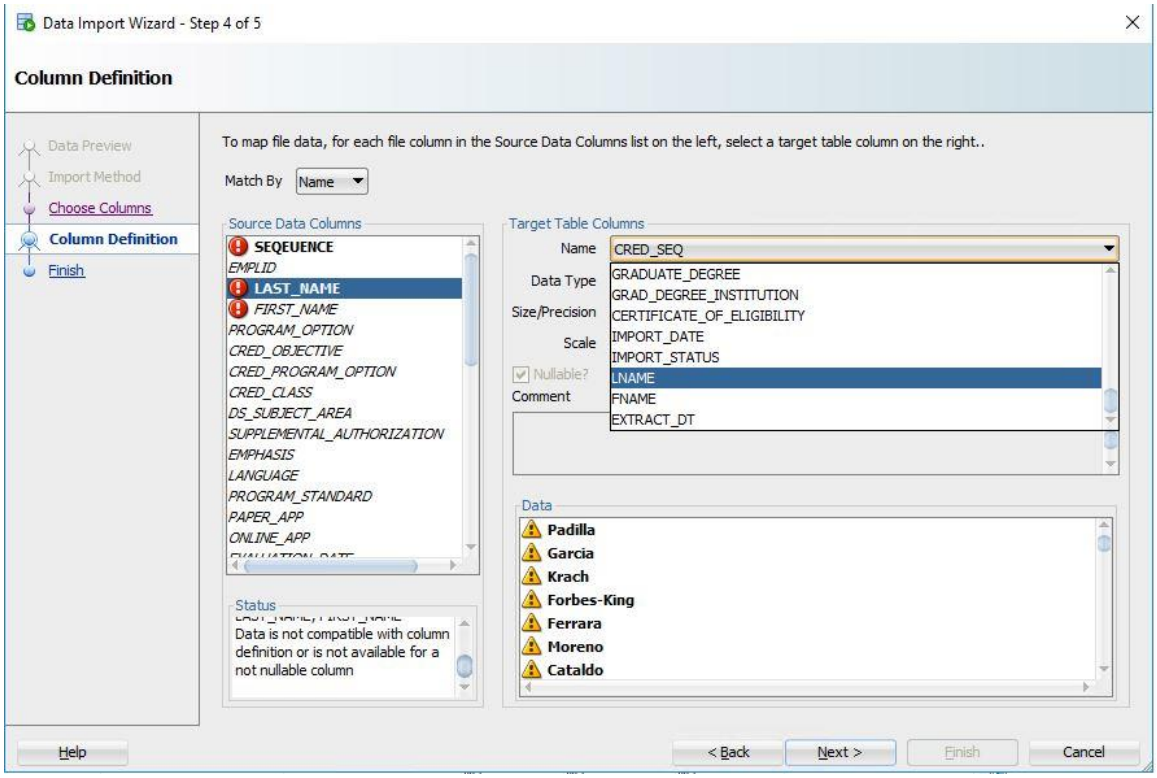


Figure 3. Data Import Wizard: Column Name Matching

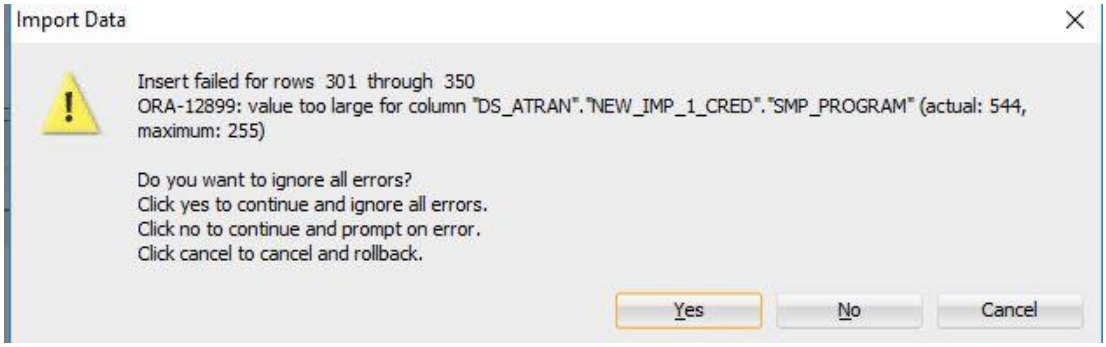


Figure 4. Data Import Wizard: Import Insertion Error Message

3.3.2 FileMaker Legacy Data to Oracle Data Field Mapping

Table 4. Mapping Legacy Data Elements to Target Table

Source Table & Column	Target Table & Column
CRED.ID_New	IMP_1_CRED.EMPLID
	IMP_1_CRED.CRED_SEQ
CRED.LastName	IMP_1_CRED.LNAME
CRED.FirstName	IMP_1_CRED.FNAME
CRED.CrObjective	IMP_1_CRED.PROGRAM_OPTION
CRED.CredentialObjective	IMP_1_CRED.CRED_OBJECTIVE
CRED.Crprogoption	IMP_1_CRED.CRED_PROGRAM_OPTION
CRED.CrClass	IMP_1_CRED.CRED_CLASS
CRED.DS_SubjArea	IMP_1_CRED.DS_SUBJECT_AREA
CRED.CRSUPPLEMENTALAUTH	IMP_1_CRED. SUPPLEMENTAL_AUTHORIZATION
CRED.BCLAD_EMPHA	IMP_1_CRED.EMPHASIS
CRED.LANGUAGE	IMP_1_CRED.LANGUAGE
CRED.PROGRAM	IMP_1_CRED.PROGRAM_STANDARD
CRED.PAPER_APP	IMP_1_CRED.PAPER_ONLINE_APP
CRED.ONLINE_APP	
CRED.EvalDate	IMP_1_CRED.EVALUATION_DATE
CRED.CTCSubmission	IMP_1_CRED.CTC_SUB_DATE
CRED.ProgCompDate	IMP_1_CRED. PROGRAM_COMPLETION_DATE
CRED.CrIssuanceDate	IMP_1_CRED.DATE_OF_ISSUANCE
CRED.Evaluator	IMP_1_CRED.EVALUATOR
CRED.SMP_Program	IMP_1_CRED.SMP_PROGRAM
CRED.SMP_Institution	IMP_1_CRED.SMP_INSTITUTION

Source Table & Column	Target Table & Column
CRED.SSATAREA	IMP_1_CRED.CSET_EXAM_SUBJECT
CRED.CBEST_Date	IMP_1_CRED.CBEST_EXAM_DATE
CRED.RICA_Date	IMP_1_CRED.RICA_EXAM_DATE
CRED.TPA_Date	IMP_1_CRED.TPA_DATE
CRED.DS_TEACHING_EXP	IMP_1_CRED.DS_TEACHING_EXP
CRED.REG_TEACHING_EXP	IMP_1_CRED.REG_TEACHING_EXP
CRED.ADM_TEACHING_EXP	IMP_1_CRED.ADM_TEACHING_EXP
CRED.Work_Experience	IMP_1_CRED.WORK_EXP
CRED.POST-SECONDARY_UNITS_COURSEWORK	IMP_1_CRED.POST_SECONDARY_UNITS_COURSEWORK
CRED.DSPrelimDate	IMP_1_CRED.DS_PRELIMISSUE_DATE
CRED.DSClearDate	IMP_1_CRED.DS_CLEAR_ISSUE_DATE
CRED.Internship	IMP_1_CRED.INTERNSHIP
CRED.Preliminary	IMP_1_CRED.PRELIMINARY
CRED.ProfessionalClear	IMP_1_CRED.PROF_CLEAR
CRED.Graduate_Degree_Major	IMP_1_CRED.GRADUATE_DEGREE
CRED.Institution	IMP_1_CRED.GRAD_DEGREE_INSTITUTION
CRED.Certificate_of_Eligibility	IMP_1_CRED.CERTIFICATE_OF_ELIGIBILITY
	IMP_1_CRED.EXTRACT_DT
MA.ID_New	IMP_2_COMP_EXAM.EMPLID
	IMP_2_COMP_EXAM.AY_RECORD_CREATED
MA.LastName	IMP_2_COMP_EXAM.LAST_NAME
MA.FirstName	IMP_2_COMP_EXAM.FIRST_NAME
MA.PROGRAMOPTION	IMP_2_COMP_EXAM.PROGRAM_OPTION

Source Table & Column	Target Table & Column
MA.ProgObjectStatus	IMP_2_COMP_EXAM.OBJECTIVE
MA.CompElDate	IMP_2_COMP_EXAM. COMP_ELIGIBILITY_DATE
MA.CompIneDate	IMP_2_COMP_EXAM. COMP_INELIGIBILITY_DATE
MA.CompExamDate	IMP_2_COMP_EXAM. CURRENT_COMP_DATE
MA.Compq1	IMP_2_COMP_EXAM.COMP_Q1
MA.Compq2	IMP_2_COMP_EXAM.COMP_Q2
MA.Compq3	IMP_2_COMP_EXAM.COMP_Q3
MA.Compq4	IMP_2_COMP_EXAM.COMP_Q4
MA.Compq5	IMP_2_COMP_EXAM.COMP_Q5

3.4 Oracle PL/SQL Package Design

Oracle databases use SQL to manipulate data. By extension PL/SQL, the programming language for SQL, is a powerful tool used to manipulate SQL. Morin [6] describes Oracle PL/SQL as a tool that combines the data-manipulating power of SQL with procedural language which promotes flexibility, portability, and security of code. As the systems at CSUSB centralize around Oracle-based applications like Oracle database and Oracle PeopleSoft, Oracle's PL/SQL is the preferred language for this endeavor.

This project utilizes a combination of PL/SQL-based stored procedures and packages to manipulate the extracted legacy data from FileMaker for clean-up. After that, they will also transform that data into standards suitable for

PeopleSoft. The import, clean-up, and staging of the legacy data from FileMaker 5 to the new Oracle database contain 17 PL/SQL procedures. An additional three procedures are used to load the PeopleSoft tables for the end-user reporting interface.

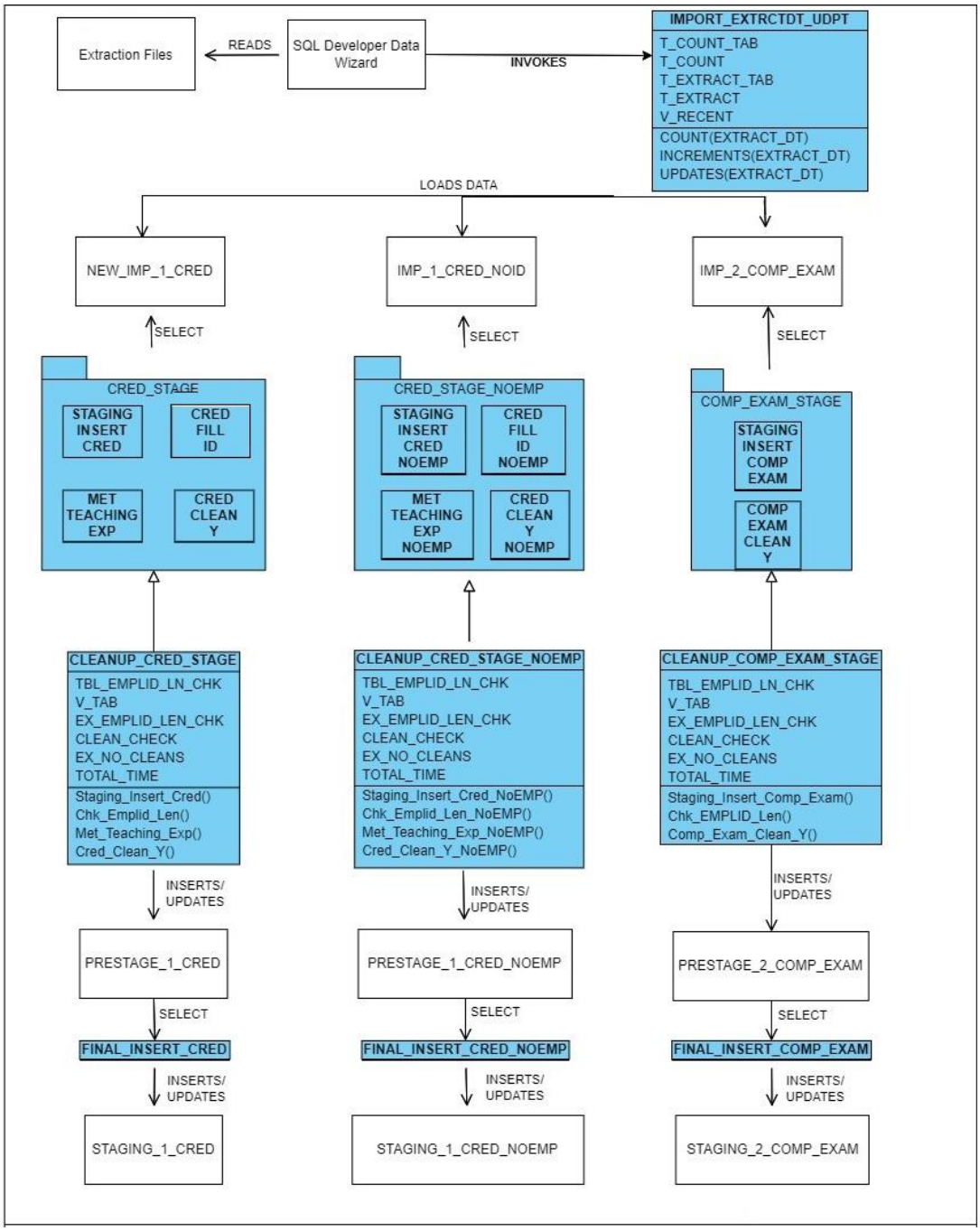


Figure 5. Migration Process: Invoking and Executing

3.4.1 Post-Import Clean-up Process

As mentioned in Section 3.3.1, the migration process between FileMaker and Oracle must be run manually. The legacy data is extracted year by year and then imported into the new database using SQL Developer's Data Import Wizard. Each of the final tables required primary keys to uniquely identify students. These primary keys are comprised of a composite set of fields, with the CRED table needing three fields for its keys (a student's extraction year, their student ID, and their CRED sequence number), whereas the COMP_EXAM only requires two fields for its composite key (the extraction year and student ID). It is also important during the cleanup process to make sure that none of these fields contain a null value so each record remains unique. Should there be any violations of the keys, the cleanup process will print a list of the offending records so that manual look-up and verification back to FileMaker can be done to retrieve the correct information and to ensure that the student's data is valid. The primary keys will also assist with indexing and searching for students once the data is moved to PeopleSoft. After referencing the legacy data using the list of search key violations, corrections must be manually completed before continuing.

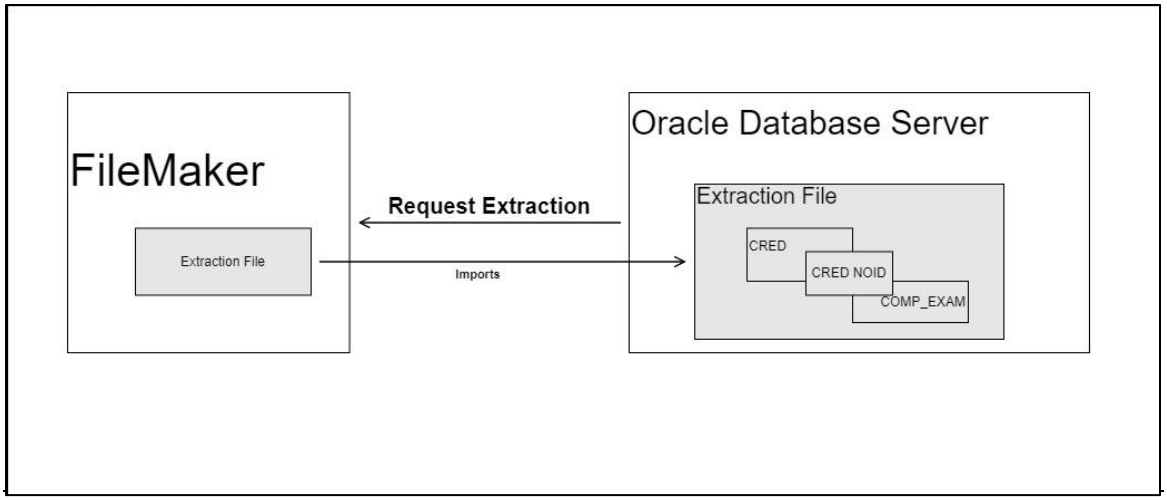


Figure 6. Request for Data Extraction.

Fields from the FileMaker data that are blank will be marked as null by default upon import. However, the PeopleSoft database system does not recognize or use null values, so fields with empty spaces are converted to a single blank character string. Fortunately, this makes logical comparisons straightforward due to how Oracle databases treat null values as non-values.

Once all post-import issues have been resolved, the data is then passed through a data clean-up process to ready it for reference in PeopleSoft.

3.4.2 Procedural Language/Structured Query Language (PL/SQL) Units

Table 5. Migration and Clean-up Program Units

• Program Name	• Purpose	• Debugging Output
• IMPORT_EXTRCT DT_UPDT	• Fills in the EXTRACT_DT field with the appropriate year extraction based on student ID and CRED_SEQ.	• Prints the list of current years in EXTRACT_DT. • Prints how many rows have a null value currently in EXTRACT_DT. • Prints new list of years after updating the null fields to the appropriate year range.
• STAGING_INSERT _CRED	• Modifies field using regular expressions for pattern matching. • Modifies field data contents according to requests from the College of Education. • Inserts fields from IMPORT CRED table to PRESTAGE CRED table. • Updates flag from the IMPORT table from “N” to “Y”.	• Prints number of rows inserted to PRESTAGE. • Prints number of rows that have the import flag updated from “N” to “Y”.
• STAGING_INSERT _COMP_EXAM	• Modifies field using regular expressions for pattern matching. • Modifies field data contents according to	• Prints number of rows inserted to PRESTAGE

• Program Name	• Purpose	• Debugging Output
	requests from the College of Education. • Inserts fields from IMPORT COMP_EXAM table to PRESTAGE COMP_EXAM table. • Updates flag from the IMPORT table from “N” to “Y”.	
• CLEANUP_CRED_STAGE	• Calls STAGING_INSERT_CRED, CRED_FILL_ID, MET_TEACHING_EXP, CRED_CLEAN_Y	• If any student ID falls outside the length of 9, an exception will be raised and the list of offending IDs will be printed. • If no rows need to be cleaned, then will raise an exception stating that no rows need to be processed. • Prints an Oracle error number and error message if some other error occurs.
• CLEANUP_COMP_EXAM_STAGE	• Calls STAGING_INSERT_COMP_EXAM, COMP_EXAM_CLEAN_Y	• If any student ID falls outside the length of 9, an exception will be raised and the list of offending IDs will be printed. • If no rows need to be cleaned, then

• Program Name	• Purpose	• Debugging Output
		will raise an exception stating that no rows need to be processed. • Prints an Oracle error number and error message if some other error occurs.
• FINAL_INSERT_CRED	• Inserts records from PRESTAGE CRED to STAGING CRED where the flag STAGING_IMP is “N” • Updates PRESTAGE CRED’s STAGING_IMP field from “N” to “Y”	• Prints the total number of rows inserted • Prints the total number of rows whose flags were updated
• FINAL_INSERT_COMP_EXAM	• Inserts records from PRESTAGE COMP_EXAM to STAGING COMP_EXAM where the flag STAGING_IMP is “N” • Updates PRESTAGE COMP_EXAM’s STAGING_IMP field from “N” to “Y”	• Prints the total number of rows inserted • Prints the total number of rows whose flags were updated
• COMP_EXAM_STAGE.COMP_EXAM_CLEAN_Y	• Marks flag for data clean-up as “Y” after process finishes.	• Prints total number of records updated when this program is called.
• CRED_STAGE.CRED_FILL_ID	• Fills in any blank student IDs with the most recent student ID.	• Prints total number of blank student ID records. • Prints list of student IDs that

• Program Name	• Purpose	• Debugging Output
		have been used to fill the blank fields.
• CRED_STAGE.MET_TEACHING_EXP	• Corrects the fields DS_TEACHING_EXP, REG_TEACHING_EXP, ADM_TEACHING_EXP based on specific criteria supplied by Dr. Mahoney	• Prints the total number of records affected by the program • Prints the list of primary key records affected.
• CRED_STAGE.CRED_CLEAN_Y	• Marks flag for data clean-up as “Y” after process finishes.	• Prints total number of records updated when this program is called.
• PS_CRED_DML	• Inserts records from STAGING CRED the PeopleSoft CRED table where PS_IMPORT = “N” • Updates PS_IMPORT to “Y” for these new inserts	• Prints the total number of rows inserted • Prints the total number of rows whose flags were updated
• PS_EXAM_DML	• Inserts records from STAGING COMP_EXAM the PeopleSoft EXAM table where PS_IMPORT = “N” • Updates PS_IMPORT to “Y” for these new inserts	• Prints the total number of rows inserted • Prints the total number of rows whose flags were updated

3.5 Database Table Structures

Several tables were created for this part of the project. As this new database process was designed around a three-stage pipeline (import, cleanup, and ready), three tables were created for each of the stages. A total of nine tables were created and a summary of each is detailed in the following sections, with design details provided in Appendix A.

3.5.1 The NOID/NOEMP Tables

Late into the project's life cycle, it was discovered that a handful of students did not have a CSUSB student ID. Upon further inspection, it was found that these students were evaluated and recommended by CSUSB for a credential but did not register as CSUSB students [5]. Registration as a CSUSB student was not always required for a credential evaluation to be completed. To ensure these records were included in the new database, the NOID tables were developed to supplement to the CRED tables. The CoE and I developed a numbering scheme to uniquely identify these students in order to preserve their records with the data migration process. The numbering scheme's notes are distributed to Mrs. Chiang of ACBI and Dr. Mahoney of the CoE for future reference. The NOID tables and programs closely follow the structure of the CRED tables and programs so the designs are nearly identical, except for the missing student IDs. Though similar, it was decided to keep the processes separate between CRED and NOID due to how the CoE works with each set of students.

3.5.2 Database Tables and Modifications from FileMaker

Table 6. Database Tables and Modifications

• Table Name	• Description	• Modification
• IMP_1_CRED	• Main table to store all the CRED extractions from FileMaker	• Added columns CRED_SEQ, IMPORT_DATE, IMPORT_STATUS, and EXTRACT_DT • Added a new column CRED_SEQ NUMBER to give each record an extra layer of indexing and to help distinguish between extraction years. • Added a new column IMPORT_DATE DATE to indicate the date when the extraction data was imported to the Oracle database. • Added a new column IMPORT_STATUS VARCHAR2(1) to indicate whether the record has been moved to the PRESTAGE stage. • Added a new column EXTRACT_DT VARCHAR2(9) to indicate the extraction year the record is from. • Combined PAPER_APP and ONLINE_APP into PAPER_ONLINE_APP. • Renamed CSET_EXAM_DATE to

• Table Name	• Description	• Modification
		CBEST_EXAM_DATE.
• IMP_1_CRED_NOEMP	• Main table to store all the CRED extractions with no traditional student IDs from FileMaker	• Same as IMP_1_CRED
• IMP_2_COMP_EXAM	• Main table to store all the EXAM extractions from FileMaker	• Added a new column AY_RECORD_CREATED VARCHAR(9) to indicate the year the extraction is from. • Added a new column IMPORT_DATE DATE to indicate the date when the extraction data was imported to the Oracle database. • Added a new column IMPORT_STATUS VARCHAR2(1) to indicate whether the record has been moved to the PRESTAGE stage.
• PRESTAGE_1_CRED	• Checkpoint table. Used to hold cleaned CRED data for review and revisions	• Adds a new column CLEAN_STATUS VARCHAR2(1) to indicate whether or not the data has finished the clean-up process • Adds a new column STAGING_IMP VARCHAR2(1) to indicate if the data has been migrated to the last stage.
• PRESTAGE_1_C	• Checkpoint table.	• Same as

• Table Name	• Description	• Modification
RED_NOEMP	Used to hold cleaned NOID data for review and revisions	PRESTAGE_1_CRED
• PRESTAGE_2_COMP_EXAM	• Checkpoint table. Used to hold cleaned EXAM data for review and revisions	• Same as PRESTAGE_1_CRED
• STAGING_1_CRED	• Final table. Used to hold cleaned CRED records ready for reference in PeopleSoft	• Added a new column PS_IMPORT VARCHAR2(1) to indicate whether or not the data has been moved to PeopleSoft
• STAGING_1_CRED_NOEMP	• Final table. Used to hold cleaned NOID records ready for reference in PeopleSoft	• Same as STAGING_1_CRED
• STAGING_2_COMP_EXAM	• Final table. Used to hold cleaned EXAM records ready for reference in PeopleSoft	• Same as STAGING_1_CRED

CHAPTER FOUR

PEOPLESOFT END-USER REPORTING INTERFACE

4.1 PeopleSoft Introduction and Design Challenges

PeopleSoft Campus Solutions is used for development and maintenance of a student system using PeopleTools. Within PeopleTools, the AppDesigner application allows developers to create PeopleSoft pages that can be accessed through a web-browser with Internet connection [7]. CSUSB uses PeopleSoft as its primary reporting tool for student records. The PeopleSoft database contains student information for every student who has been enrolled at CSUSB, making it the best tool to develop the end-user reporting interface for the College of Education. Extracted student records from the legacy FileMaker data are joined with PeopleSoft records using student ID. For the rare cases of students without a CSUSB student ID, a separate page was built, and a custom ID was assigned to each record for identification. The following table details the development cycle of the End-User Reporting Interface:

Table 7. Reporting Interface Development Cycle

Stage	Task
Strategy and Analysis	• Documentation on necessary PeopleSoft project objects.
Design	• Creation of necessary fields and record for data reference to the Oracle tables. • Creation of PeopleSoft page security settings.
Development	• Creation of the end-user pages. • Population of the PeopleSoft records with data from the STAGING Oracle database.
Testing	• Debugging end-user access to the system. • Demonstrating page structure and functionality to the end-user. • Modification of pages according to user feedback.
Implementation	• Finalization of reporting pages after full approval of the College of Education department.
Maintenance	• Modification and upkeep of the pages for better end-user

Stage	Task
	experience. • Tracking changes in security settings as needed.

The challenge in this part of the project was learning and understanding the workflow that ACBI's developers use to develop PeopleSoft pages. Each webpage required creation of several PeopleSoft objects (like fields, records, page structures, components, menus, etc.) to account for the page's design, security, structure, and access. Each of these objects was interconnected with the others and it was essential to keep track of the flow of object creation to create a stable and secure reporting interface. The workflow employed by ACBI uses database cloning and migration, so it was important the project had all the necessary requirements so that settings and data were not lost. Certain objects within AppDesigner required writing PeopleSoft's PeopleCode to configure a variety of properties. As PeopleCode structure and content is strictly regulated, reusability of existing code was made possible through direct permission and supervision from CSUSB's ACBI department [8]. ACBI's code templates and documentation were provided and the PeopleCode of this project was written with assistance from these resources to fit the specifications of the project.

4.2 End User Reporting Interface Development

4.2.1 PeopleSoft Fields and Records

Fields created within AppDesigner can be shared with other projects. However, fields shared from other projects are not able to be customized or else the modifications affect every other project that uses the modified field [7]. For the purposes of this project, it was determined that less risk would be encountered to create new fields specifically for each record to have full control of field management. If field properties, like size or name, needed to be changed during development then can be done so without the risk of unintentionally modifying another developer's records. The field mapping between PeopleSoft and the Oracle database is shown in the following table. The PeopleSoft tables (called records) are of the same structure as their Oracle database counterparts and are listed in the following figures.

Table 8. PeopleSoft to Oracle Database Field Mapping

PeopleSoft Field	Oracle Database Field
EMPLID (KEY)	IMP_1_CRED.EMPLID
SB_COE_I_SEQ (KEY)	IMP_1_CRED.CRED_SEQ
SB_COE_I_PROG_OPT	IMP_1_CRED.PROGRAM_OPTION
SB_COE_I_CREDOBJ	IMP_1_CRED.CRED_OBJECTIVE
SB_COE_I_CPOBJ	IMP_1_CRED.CRED_PROGRAM_OPTION
SB_COE_I_CRED_CLASS	IMP_1_CRED.CRED_CLASS
SB_COE_I_DSSUB	IMP_1_CRED.DS_SUBJECT_AREA
SB_COE_I_SUPPAUTH	IMP_1_CRED. SUPPLEMENTAL_AUTHORIZATION

PeopleSoft Field	Oracle Database Field
SB_COE_I_EMPHASIS	IMP_1_CRED.EMPHASIS
SB_COE_I_LANGUAGE	IMP_1_CRED.LANGUAGE
SB_COE_I_PROG_STD	IMP_1_CRED.PROGRAM_STANDARD
SB_COE_I_PAPERAPP	IMP_1_CRED.PAPER_ONLINE_APP
SB_COE_I_EVAL_DT	IMP_1_CRED.EVALUATION_DATE
SB_COE_I_CTC_SUBDT	IMP_1_CRED.CTC_SUB_DATE
SB_COE_I_CMPDT	IMP_1_CRED. PROGRAM_COMPLETION_DATE
SB_COE_I_DT_ISS	IMP_1_CRED.DATE_OF_ISSUANCE
SB_COE_I_EVALUATOR	IMP_1_CRED.EVALUATOR
SB_COE_I_SMP_PRG	IMP_1_CRED.SMP_PROGRAM
SB_COE_I_SMP_INST	IMP_1_CRED.SMP_INSTITUTION
SB_COE_I_CSET_SUBJ	IMP_1_CRED.CSET_EXAM_SUBJECT
SB_COE_I_CSET_DT (VISUALLY APPEARS AS CBEST)	IMP_1_CRED.CBEST_EXAM_DATE
SB_COE_I_RICA_DT	IMP_1_CRED.RICA_EXAM_DATE
SB_COE_I_TPA_DT	IMP_1_CRED.TPA_DATE
SB_COE_I_DS_EXP	IMP_1_CRED.DS_TEACHING_EXP
SB_COE_I_REG_EXP	IMP_1_CRED.REG_TEACHING_EXP
SB_COE_I_ADM_EXP	IMP_1_CRED.ADM_TEACHING_EXP
SB_COE_I_WORK_EXP	IMP_1_CRED.WORK_EXP
SB_COE_I_PST2_WRK	IMP_1_CRED.POST_SECONDARY_UNITS_ COURSEWORK
SB_COE_I_DS_PRELIM	IMP_1_CRED.DS_PRELIMISSUE_DATE
SB_COE_I_DS_CLEAR	IMP_1_CRED.DS_CLEAR_ISSUE_DATE
SB_COE_I_INTERNSHIP	IMP_1_CRED.INTERNSHIP

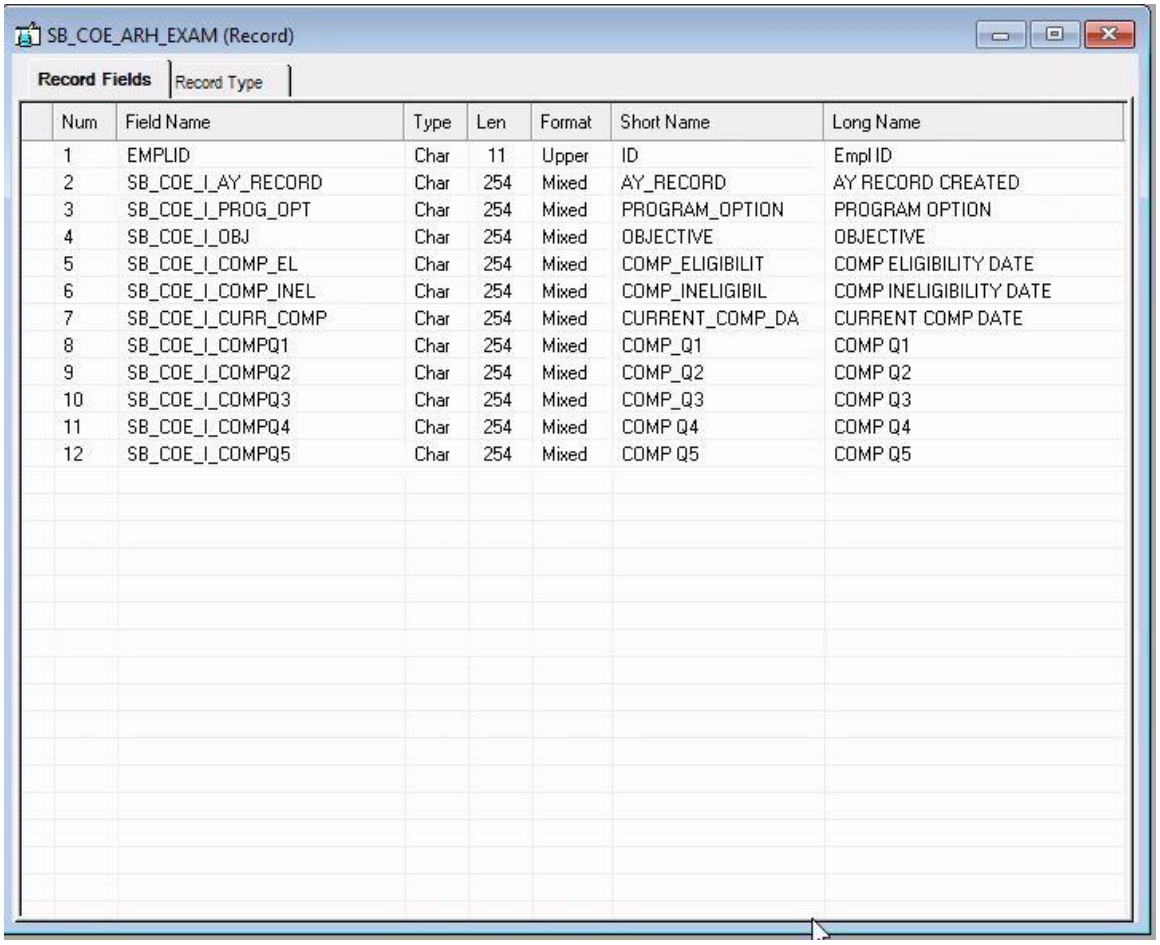
PeopleSoft Field	Oracle Database Field
SB_COE_I_PRELIM	IMP_1_CRED.PRELIMINARY
SB_COE_I_PROF_CLR	IMP_1_CRED.PROF_CLEAR
SB_COE_I_GRAD_DEG	IMP_1_CRED.GRADUATE_DEGREE
SB_COE_I_GRD_DEG_I	IMP_1_CRED. GRAD_DEGREE_INSTITUTION
SB_COE_I_CERT_ELIG	IMP_1_CRED. CERTIFICATE_OF_ELIGIBILITY
SB_COE_I_EXTRDT (KEY)	IMP_1_CRED.EXTRACT_DT
EMPLID (KEY)	IMP_2_COMP_EXAM.EMPLID
SB_COE_I_AY_RECORD (KEY)	IMP_2_COMP_EXAM. AY_RECORD_CREATED
SB_COE_I_PROG_OBJ	IMP_2_COMP_EXAM.PROGRAM_OPTION
SB_COE_I_OBJ	IMP_2_COMP_EXAM.OBJECTIVE
SB_COE_I_COMP_EL	IMP_2_COMP_EXAM. COMP_ELIGIBILITY_DATE
SB_COE_I_COMP_INEL	IMP_2_COMP_EXAM. COMP_INELIGIBILITY_DATE
SB_COE_I_CURR_COMP	IMP_2_COMP_EXAM. CURRENT_COMP_DATE
SB_COE_I_COMPQ1	IMP_2_COMP_EXAM.COMP_Q1
SB_COE_I_COMPQ2	IMP_2_COMP_EXAM.COMP_Q2
SB_COE_I_COMPQ3	IMP_2_COMP_EXAM.COMP_Q3
SB_COE_I_COMPQ4	IMP_2_COMP_EXAM.COMP_Q4
SB_COE_I_COMPQ5	IMP_2_COMP_EXAM.COMP_Q5

SB_COE_ARH_CRED (Record)

Record Fields | Record Type

Num	Field Name	Type	Len	Format	Short Name	Long Name
1	EMPLID	Char	11	Upper	ID	Empl ID
2	SB_COE_I_PROG_OPT	Char	254	Mixed	PROGRAM_OPTION	PROGRAM OPTION
3	SB_COE_I_CREDOBJ	Char	254	Mixed	CRED OBJECTIVE	CRED OBJECTIVE
4	SB_COE_I_CPOBJ	Char	254	Mixed	CRED PROGRAM OP	CRED PROGRAM OPTION
5	SB_COE_I_CRED_CLSS	Char	254	Mixed	CRED CLASS	CRED CLASS
6	SB_COE_I_DSSUB	Long	550		DS SUBJECT AREA	DS SUBJECT AREA
7	SB_COE_I_SUPPAUTH	Char	254	Mixed	SUPPLEMENTAL AU	SUPPLEMENTAL AUTHORIZATION
8	SB_COE_I_EMPHASIS	Char	254	Mixed	EMPHASIS	EMPHASIS
9	SB_COE_I_LANGUAGE	Char	254	Mixed	LANGUAGE	LANGUAGE
10	SB_COE_I_PROG_STD	Char	254	Mixed	PROGRAM STANDAR	PROGRAM STANDARD
11	SB_COE_I_PAPERAPP	Char	254	Mixed	PAPER APP	PAPER APP
12	SB_COE_I_EVAL_DT	Char	254	Mixed	EVALUATION DATE	EVALUATION DATE
13	SB_COE_I_CTC_SUBDT	Char	254	Mixed	CTC SUB DATE	CTC SUB DATE
14	SB_COE_I_PRG_CMPDT	Char	254	Mixed	PROGRAM COMPLET	PROGRAM COMPLETION DATE
15	SB_COE_I_DT_ISS	Char	254	Mixed	DATE OF ISSUANC	DATE OF ISSUANCE
16	SB_COE_I_EVALUATOR	Char	254	Mixed	EVALUATOR	EVALUATOR
17	SB_COE_I_SMP_PRG	Char	254	Mixed	SMP PROGRAM	SMP PROGRAM
18	SB_COE_I_SMP_INST	Char	254	Mixed	SMP INSTITUTION	SMP INSTITUTION
19	SB_COE_I_CSET_SUBJ	Char	254	Mixed	CSET EXAM SUBJE	CSET EXAM SUBJECT
20	SB_COE_I_CSET_DT	Char	254	Mixed	CBEST EXAM DATE	CBEST EXAM DATE
21	SB_COE_I_RICA_DT	Char	254	Mixed	RICA EXAM DATE	RICA EXAM DATE
22	SB_COE_I_TPA_DT	Char	254	Mixed	TPA DATE	TPA DATE
23	SB_COE_I_DS_EXP	Long	999		DS TEACHING EXP	DS TEACHING EXP
24	SB_COE_I_REG_EXP	Char	254	Mixed	REG TEACHING EX	REG TEACHING EXP
25	SB_COE_I_ADM_EXP	Char	254	Mixed	ADM TEACHING EX	ADM TEACHING EXP
26	SB_COE_I_WORK_EXP	Long	999		WORK EXP	WORK EXP
27	SB_COE_I_PST2_WRK	Char	254	Mixed	POST SECONDARY	POST SECONDARY UNITS CRSEWRK
28	SB_COE_I_DS_PRELIM	Char	254	Mixed	DS PRELIM ISSUE	DS PRELIM ISSUE DATE
29	SB_COE_I_DS_CLEAR	Char	254	Mixed	DS CLEAN ISSUE	DS CLEAN ISSUE DATE
30	SB_COE_I_INTERNSHP	Char	254	Mixed	INTERNSHIP	INTERNSHIP
31	SB_COE_I_PRELIM	Char	254	Mixed	PRELIMINARY	PRELIMINARY
32	SB_COE_I_PROF_CLR	Char	254	Mixed	PROF CLEAR	PROF CLEAR
33	SB_COE_I_GRAD_DEG	Char	254	Mixed	GRADUTE DEGREE	GRADUATE DEGREE
34	SB_COE_I_GRD_DEG_I	Char	254	Mixed	GRAD DEGREE INS	GRAD DEGREE INSTITUTION
35	SB_COE_I_CERT_ELIG	Char	254	Mixed	CERTIFICATE OF	CERTIFICATE OF ELIGIBILITY
36	UPDATE_DT	Date	10		Update	Last Updated
37	SB_COE_I_SEQ	Nbr	5		CREDSEQ	Cred Sequence
38	SB_COE_I_EXTRDT	Char	9	Upper	Extract Date	Extract Date

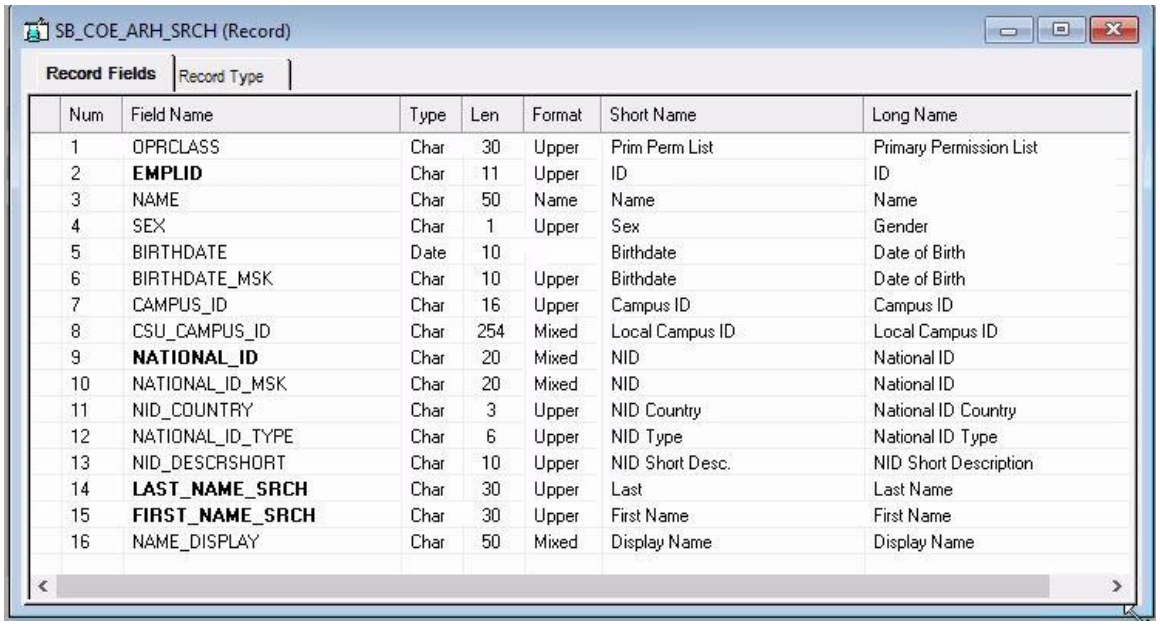
Figure 7. PeopleSoft Credential Record Structure



Num	Field Name	Type	Len	Format	Short Name	Long Name
1	EMPLID	Char	11	Upper	ID	Empl ID
2	SB_COE_I_AY_RECORD	Char	254	Mixed	AY_RECORD	AY RECORD CREATED
3	SB_COE_I_PROG_OPT	Char	254	Mixed	PROGRAM_OPTION	PROGRAM OPTION
4	SB_COE_I_OBJ	Char	254	Mixed	OBJECTIVE	OBJECTIVE
5	SB_COE_I_COMP_EL	Char	254	Mixed	COMP_ELIGIBILIT	COMP ELIGIBILITY DATE
6	SB_COE_I_COMP_INEL	Char	254	Mixed	COMP_INELIGIBIL	COMP INELIGIBILITY DATE
7	SB_COE_I_CURR_COMP	Char	254	Mixed	CURRENT_COMP_DA	CURRENT COMP DATE
8	SB_COE_I_COMPQ1	Char	254	Mixed	COMP_Q1	COMP Q1
9	SB_COE_I_COMPQ2	Char	254	Mixed	COMP_Q2	COMP Q2
10	SB_COE_I_COMPQ3	Char	254	Mixed	COMP_Q3	COMP Q3
11	SB_COE_I_COMPQ4	Char	254	Mixed	COMP Q4	COMP Q4
12	SB_COE_I_COMPQ5	Char	254	Mixed	COMP Q5	COMP Q5

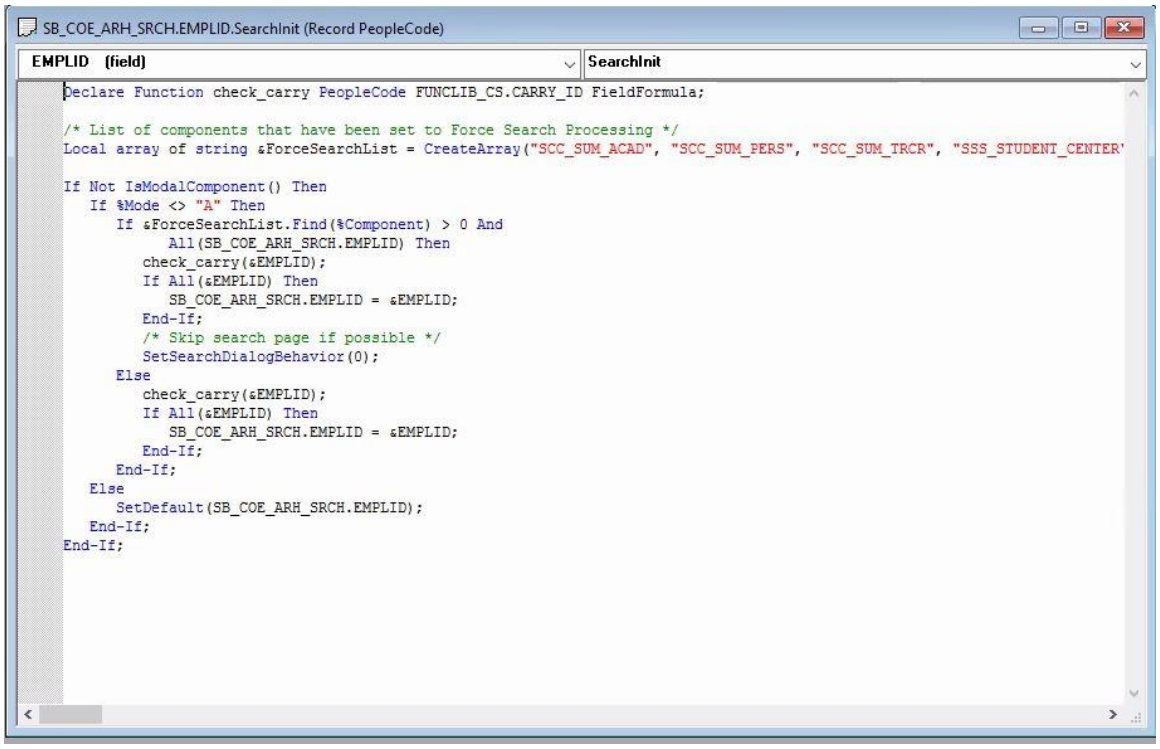
Figure 8. PeopleSoft Comprehensive Exam Record Structure

4.2.1.2. The SRCH Record. The SRCH record is derived from CSUSB's student search record and enables the creation of a search page, allowing users to search for students in the campus's PeopleSoft database by the fields identified in bold in the following figure. Fields not in bold have their search capabilities disabled as they are outside the scope of the project. The following figures detail the SRCH record along with the PeopleCode used for each of the search fields.



Num	Field Name	Type	Len	Format	Short Name	Long Name
1	OPRCLASS	Char	30	Upper	Prim Perm List	Primary Permission List
2	**EMPLID**	Char	11	Upper	ID	ID
3	NAME	Char	50	Name	Name	Name
4	SEX	Char	1	Upper	Sex	Gender
5	BIRTHDATE	Date	10		Birthdate	Date of Birth
6	BIRTHDATE_MSK	Char	10	Upper	Birthdate	Date of Birth
7	CAMPUS_ID	Char	16	Upper	Campus ID	Campus ID
8	CSU_CAMPUS_ID	Char	254	Mixed	Local Campus ID	Local Campus ID
9	**NATIONAL_ID**	Char	20	Mixed	NID	National ID
10	NATIONAL_ID_MSK	Char	20	Mixed	NID	National ID
11	NID_COUNTRY	Char	3	Upper	NID Country	National ID Country
12	NATIONAL_ID_TYPE	Char	6	Upper	NID Type	National ID Type
13	NID_DESCRSHORT	Char	10	Upper	NID Short Desc.	NID Short Description
14	**LAST_NAME_SRCH**	Char	30	Upper	Last	Last Name
15	**FIRST_NAME_SRCH**	Char	30	Upper	First Name	First Name
16	NAME_DISPLAY	Char	50	Mixed	Display Name	Display Name

Figure 9. PeopleSoft SRCH Record




```
Declare Function check_carry PeopleCode FUNCLIB_CS.CARRY_ID FieldFormula;

/* List of components that have been set to Force Search Processing */
Local array of string &ForceSearchList = CreateArray("SCC_SUM_ACAD", "SCC_SUM_PERS", "SCC_SUM_TRCR", "SSS_STUDENT_CENTER'

If Not IsModalComponent() Then
   If %Mode <> "A" Then
      If &ForceSearchList.Find(%Component) > 0 And
            All(SB_COE_ARH_SRCH.EMPLID) Then
         check_carry(&EMPLID);
         If All(&EMPLID) Then
            SB_COE_ARH_SRCH.EMPLID = &EMPLID;
         End-If;
         /* Skip search page if possible */
         SetSearchDialogBehavior(0);
      Else
         check_carry(&EMPLID);
         If All(&EMPLID) Then
            SB_COE_ARH_SRCH.EMPLID = &EMPLID;
         End-If;
      End-If;
   Else
      SetDefault(SB_COE_ARH_SRCH.EMPLID);
   End-If;
End-If;
```

Figure 10. SRCH Record's Student ID PeopleCode

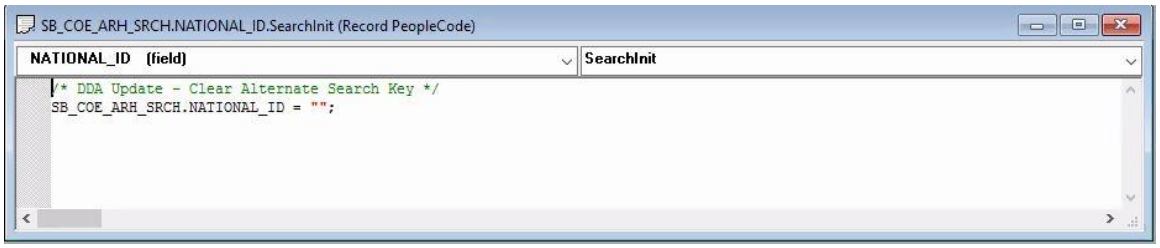


Figure 11. SRCH Record's National ID PeopleCode

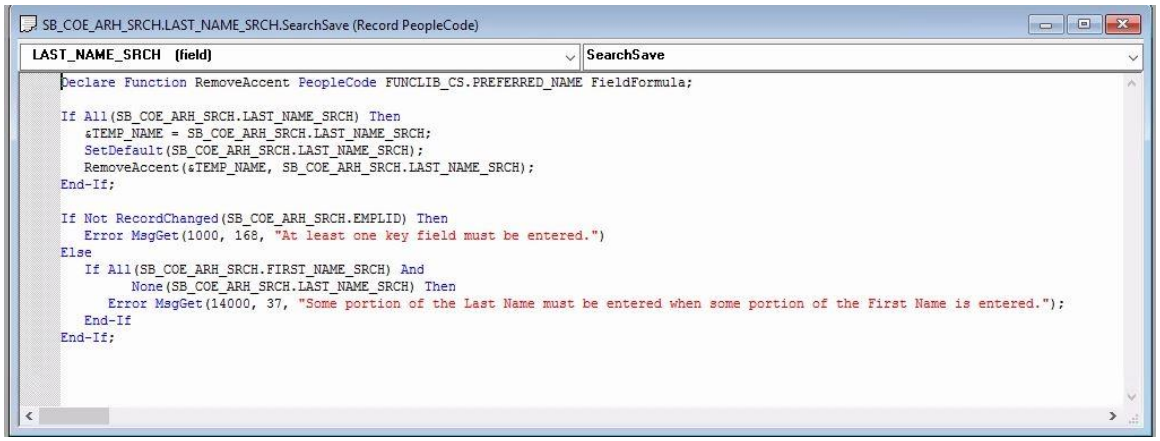


Figure 12. SRCH Record's Last Name PeopleCode

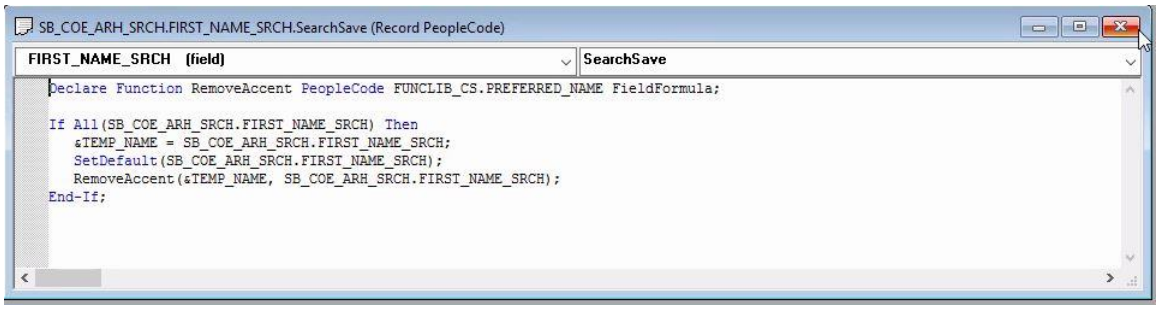


Figure 13. SRCH Record's First Name PeopleCode

4.2.2 Page Development and Design

The purpose of developing a new end-user reporting interface was to provide the College of Education with a more focused reporting tool based on Credentials Recommended information and Comprehensive Exam information. Development of the reporting interface was done using the grid design of PeopleSoft pages. The following figures demonstrate the old FileMaker interface compared to the new PeopleSoft interface.

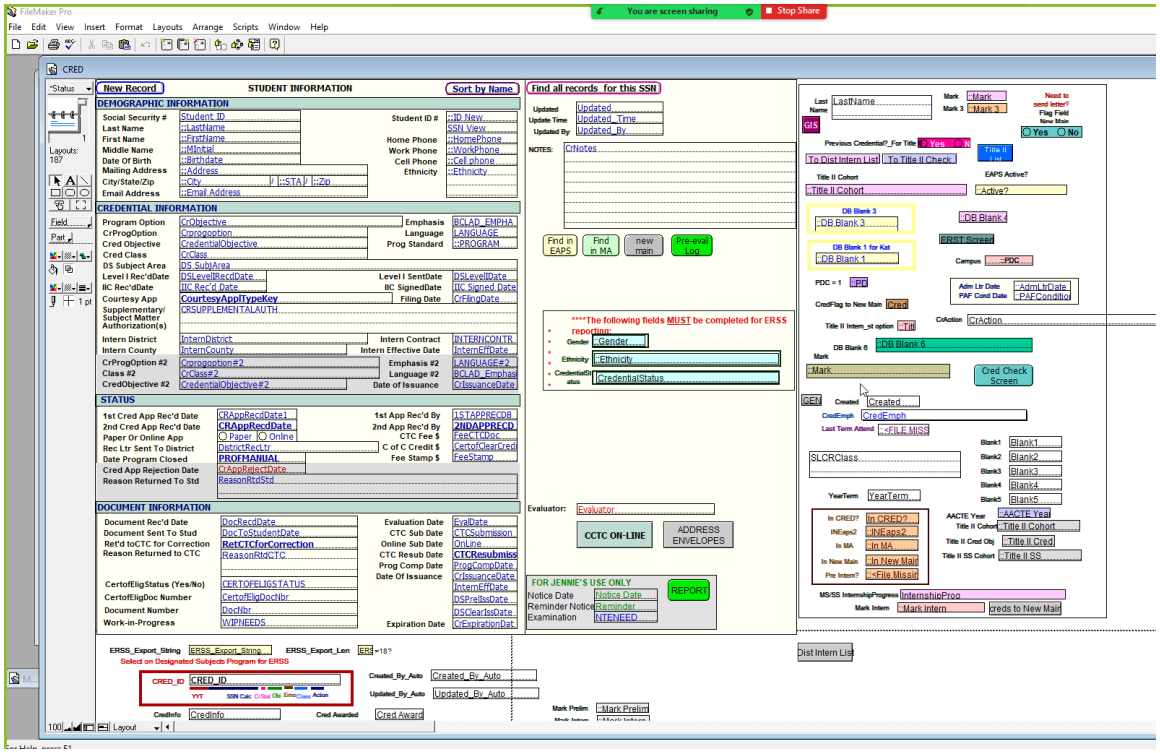


Figure 14. FileMaker CRED Reporting Page Template

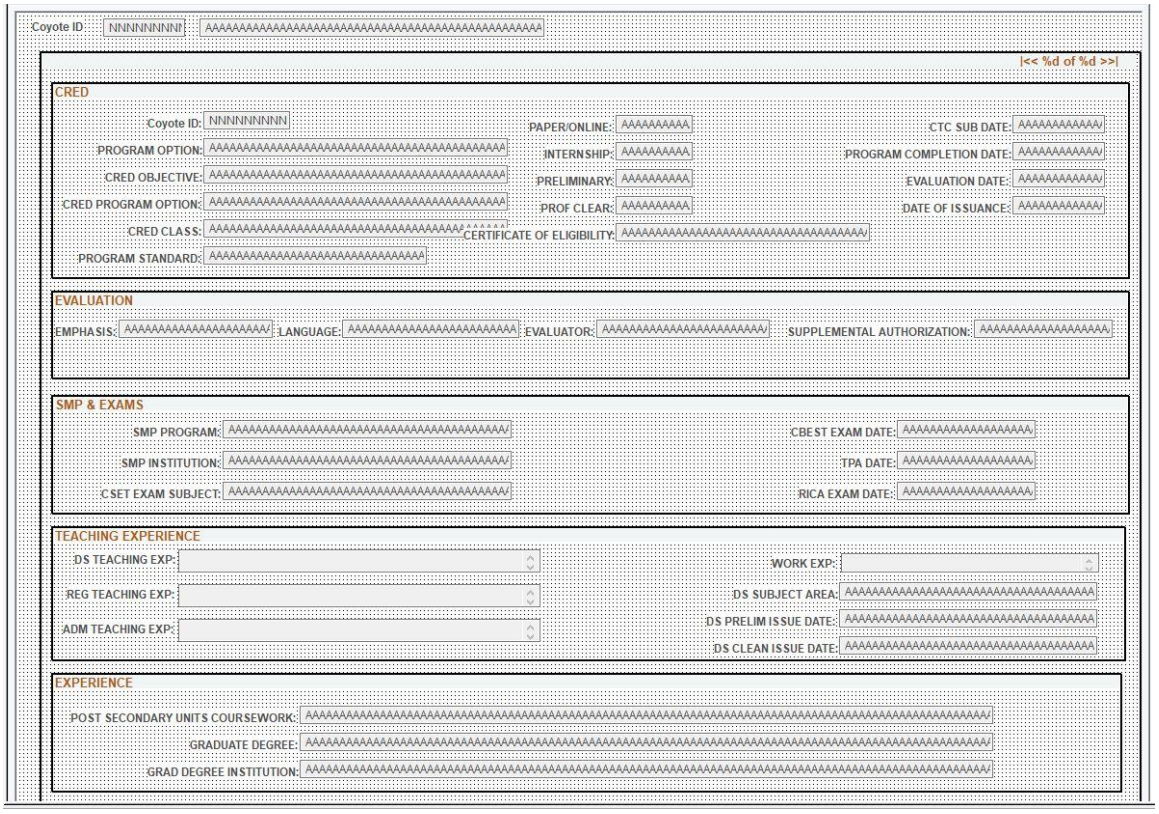


Figure 15. PeopleSoft CRED Page Development

4.2.3 Component Design

In PeopleSoft, the function of components is to logically group a page or a set of pages together for a specific purpose. Components are normally comprised of a set of pages, a search record that is used for retrieving data for the pages, and navigation links for moving through pages [7]. This project contains two components: 1) the search and reporting of the CRED and EXAM tables; and 2) the search and reporting for the NOID table.

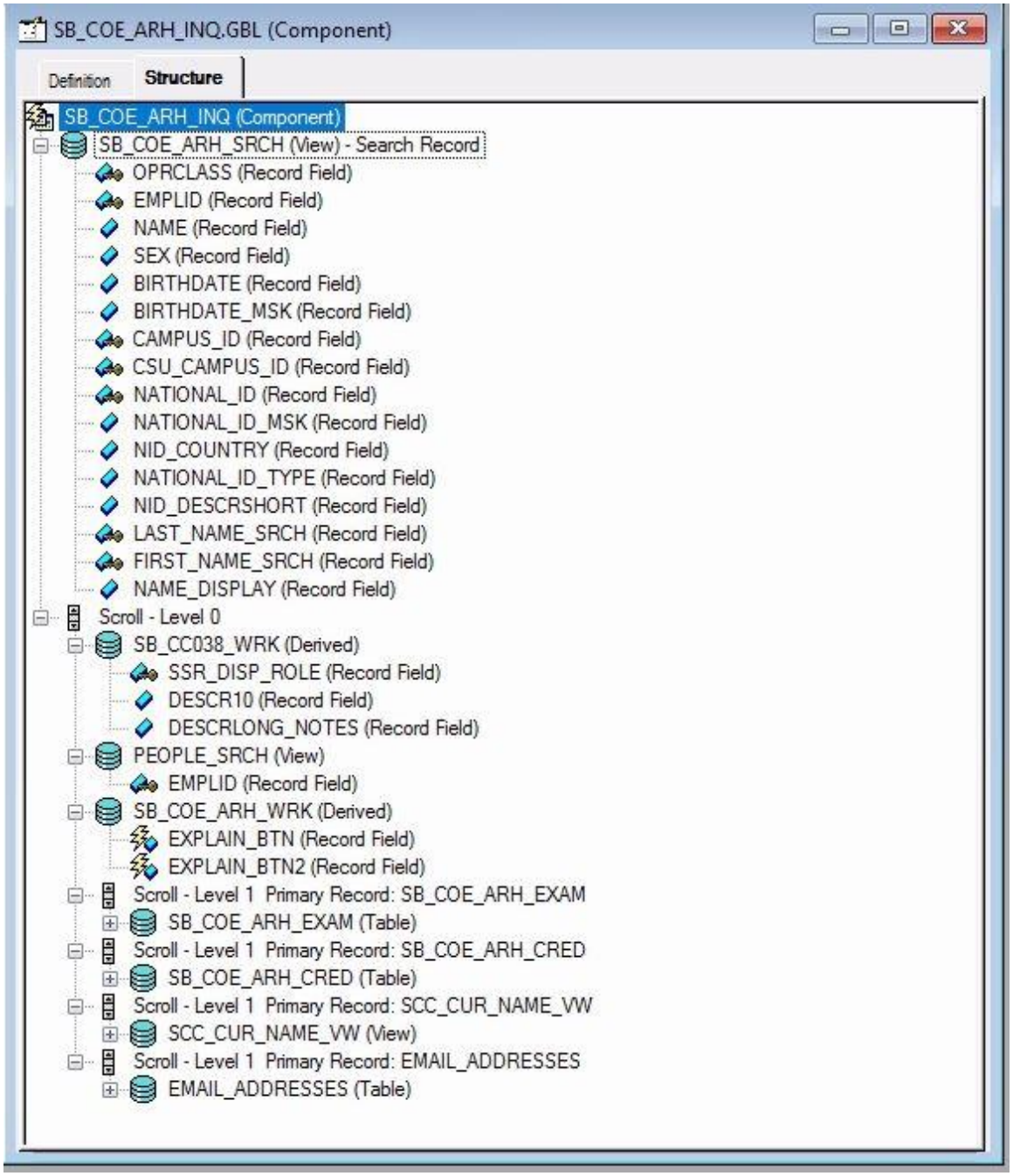


Figure 16. CRED and EXAM Component Structure

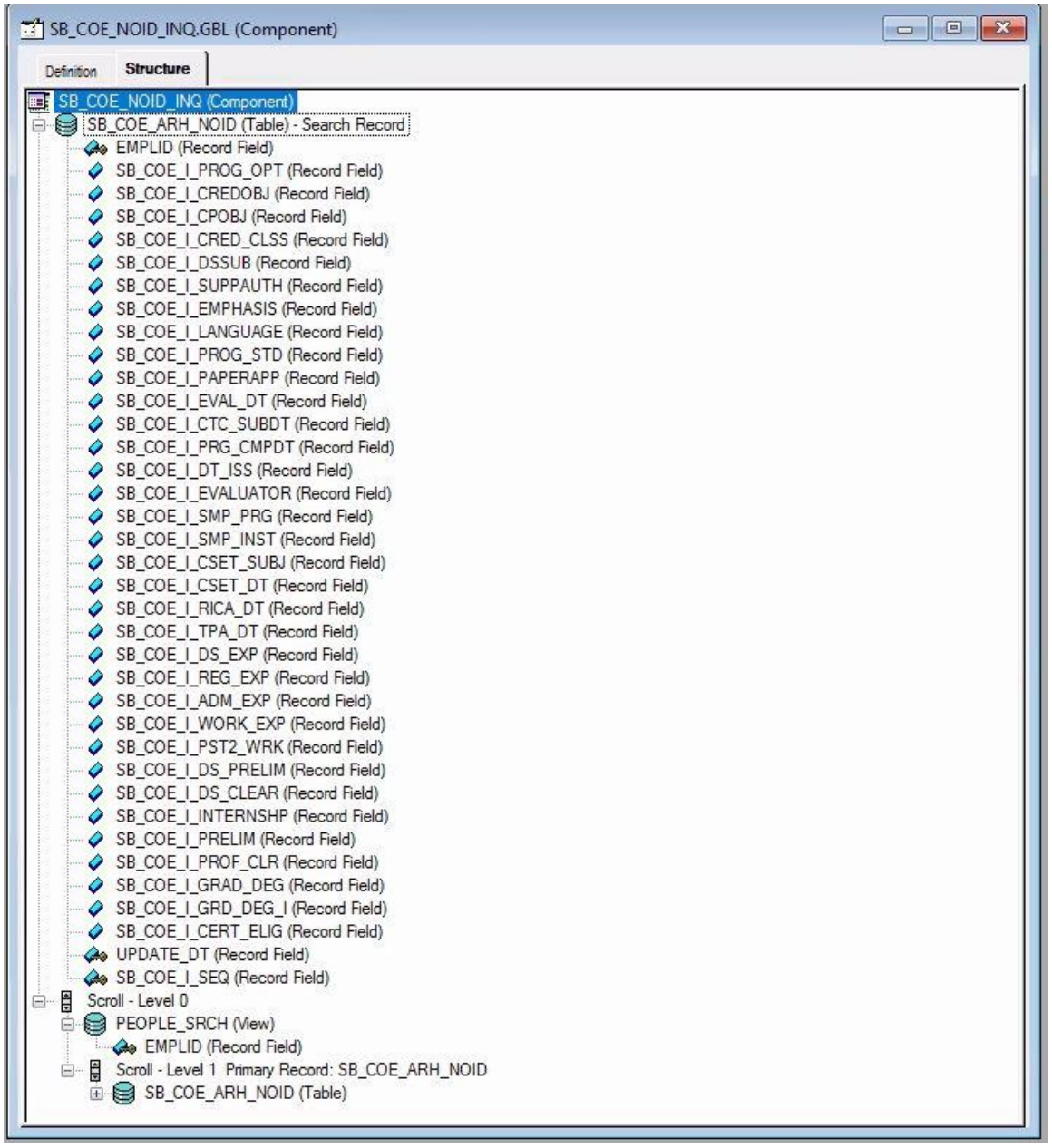


Figure 17. NOID Component Structure

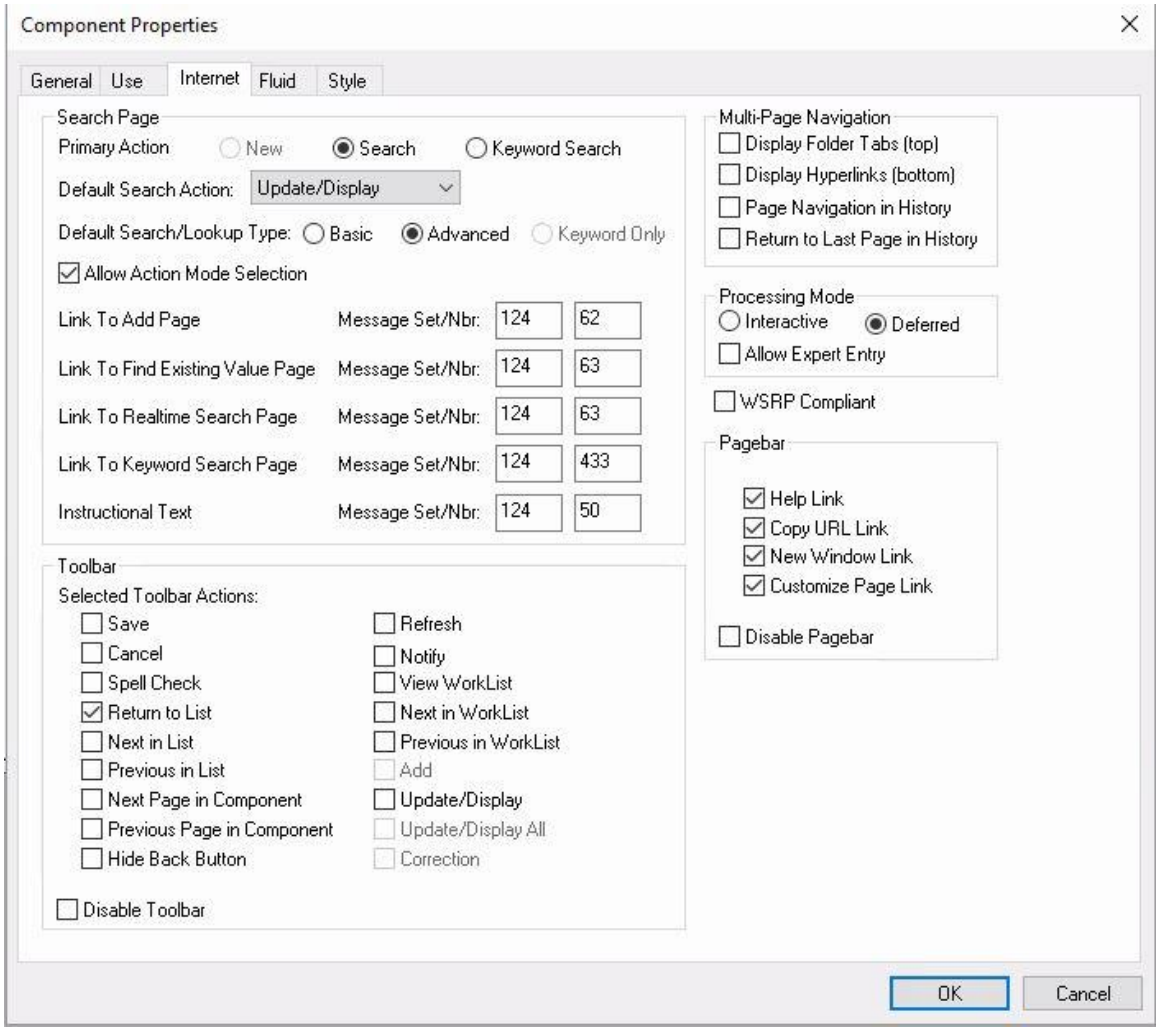


Figure 18. Component Connection Settings

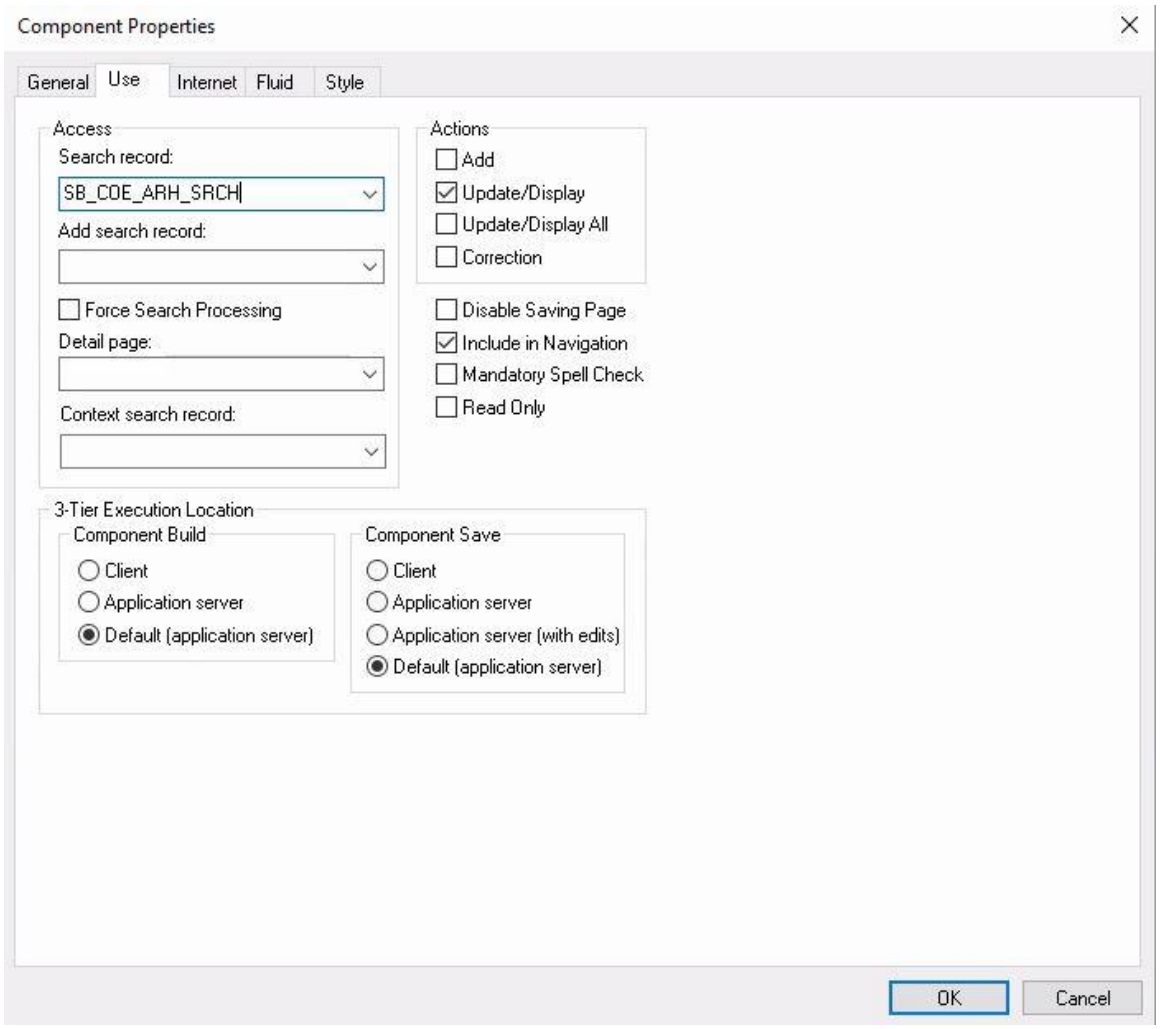


Figure 19. Component Usage and Search Settings

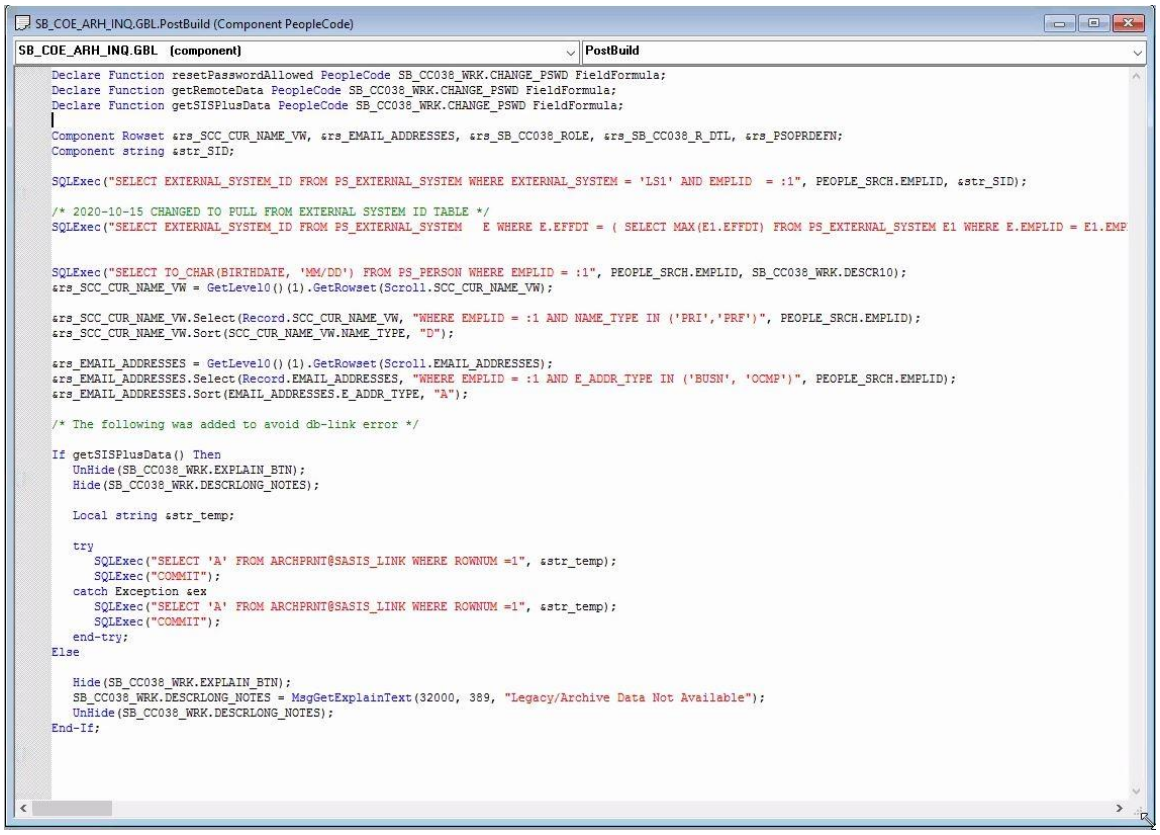


Figure 20. Component PeopleCode

4.2.4 Menu Design

Just as a component is used to logically group pages together, a menu is used to logically group components together. The purpose of this additional layer of grouping is for ease of assigning privileges and maintaining security between users [7]. This project contains a single menu, used to group the two components (COMP and EXAM, and NOID) together.

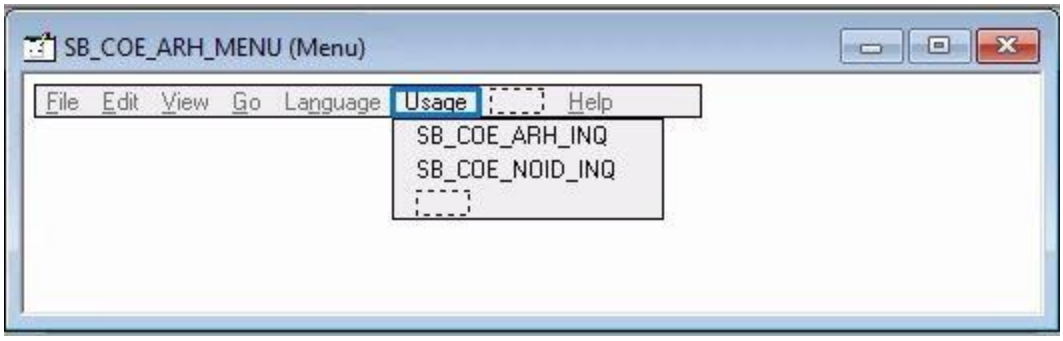


Figure 21. Menu Structure

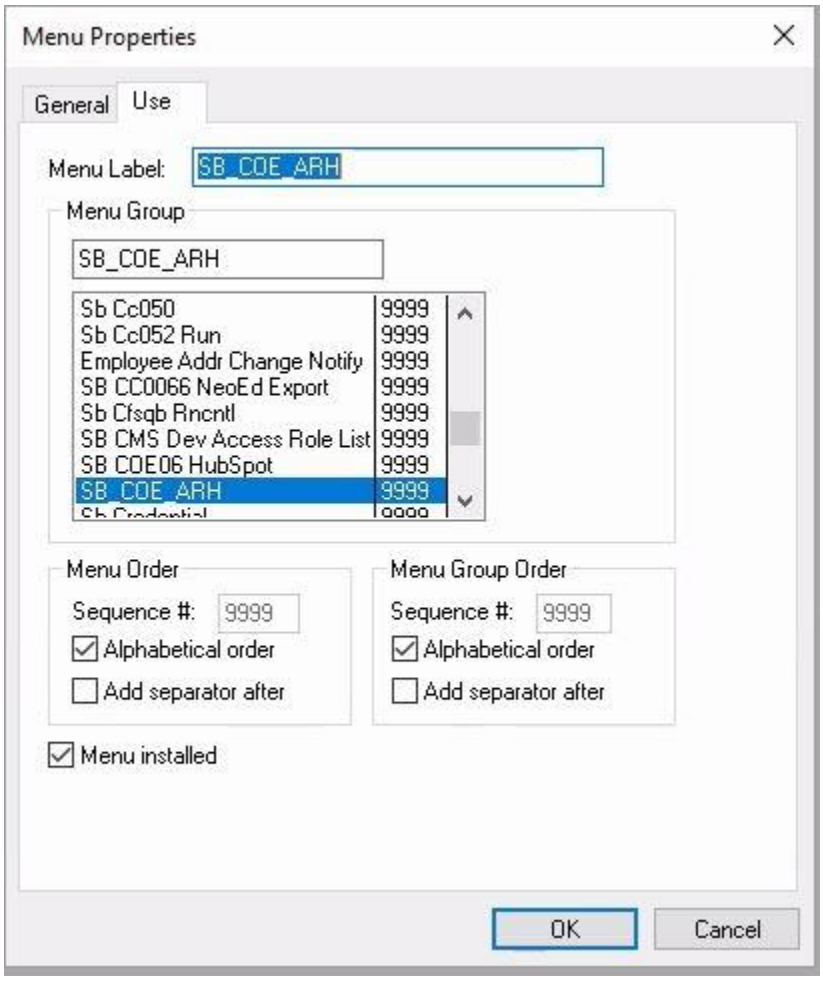


Figure 22. Configuring Menu Use Properties

4.2.5 Setting Upgrade Settings with Portal Registry

Portal Registry Structures must be created for PeopleSoft database migration and cloning. This process ensures all items within the end-user reporting interface are packaged together and all page settings are saved.

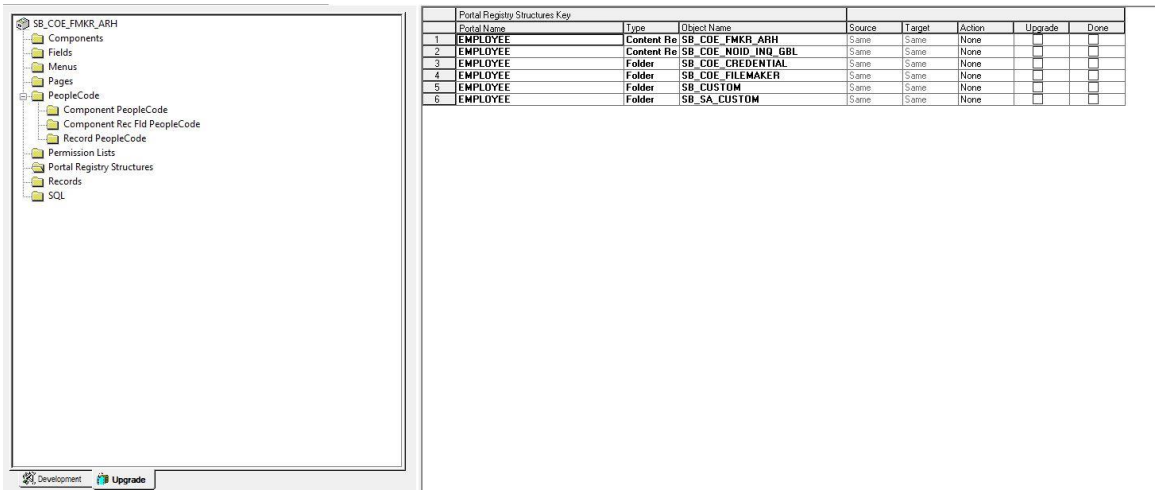


Figure 23. AppDesigner Portal Registry Structures

After configuring Portal Registry, a content reference must be declared. According to Oracle [7], “a content reference is simply a reference to a URL”. A content reference is considered registered once the content registry is created within the portal registry. Whenever users access a URL, the URL is looked up in the portal registry, which then retrieves the target content reference.

4.2.6 Security Design and Settings

The process of setting up security within PeopleSoft is straightforward. A designer must first designate a list of permissions to be granted to a user, create a role using that list of permissions, and then finally grant the role to specific user

IDs. The following figure was retrieved from Oracle and describes the relationship between user, roles and permissions lists [7].

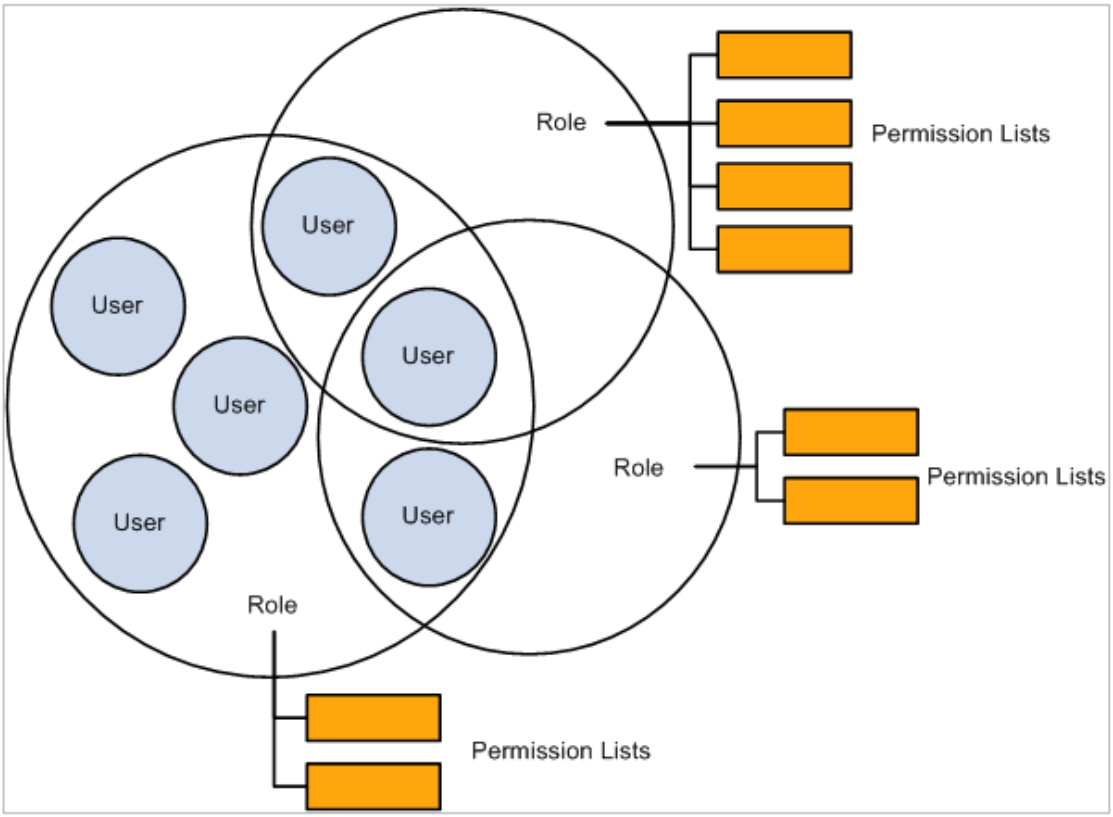


Figure 24. Relationship between User, Role, and Permissions

Once component and menu creation is finished and the Portal Registry has been set up in AppDesigner, the developer or administrator must log-in to CSUSB's Campus Solution Portal to set up page security. The first step is to define a security of permission list prior to creating any roles. The next step is to create a security role for users to access the reporting interface pages. A developer or administrator can specify what actions the role is able to perform and what the role is able to view and modify.

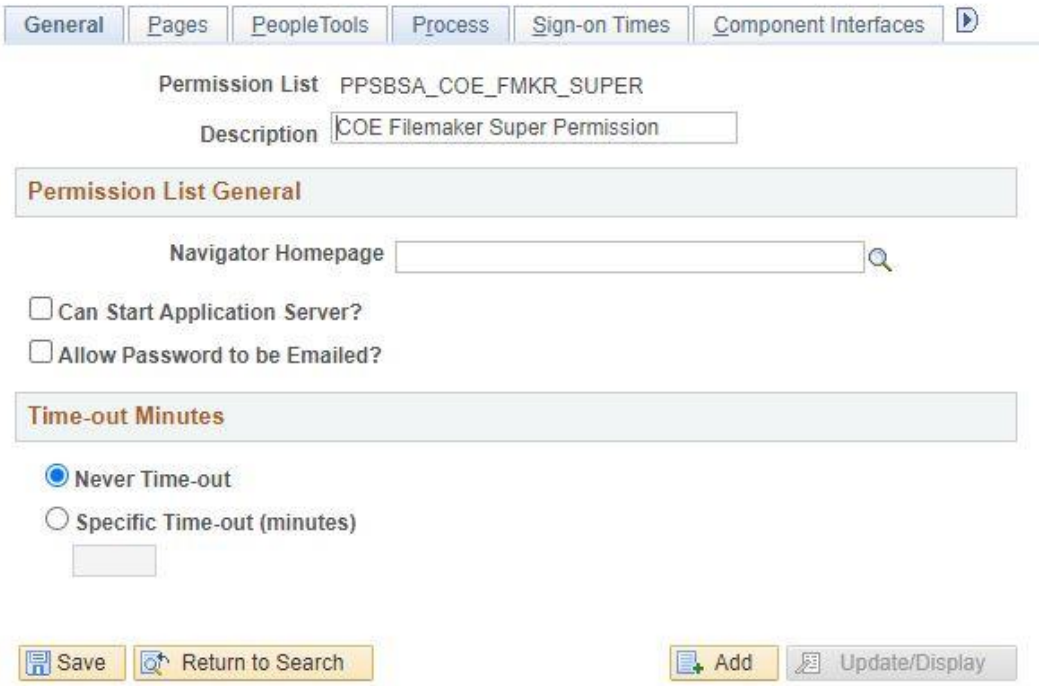


Figure 25. PeopleSoft Role Creation

After creation of the role, the developer may grant the appropriate actions and permissions to each specific page. Each page can be granted authorization access and can be marked to be editable or display only. Page actions include: adding records; updating or displaying page records; and performing page corrections. The last step is to grant the role to specific users within the PeopleSoft system.

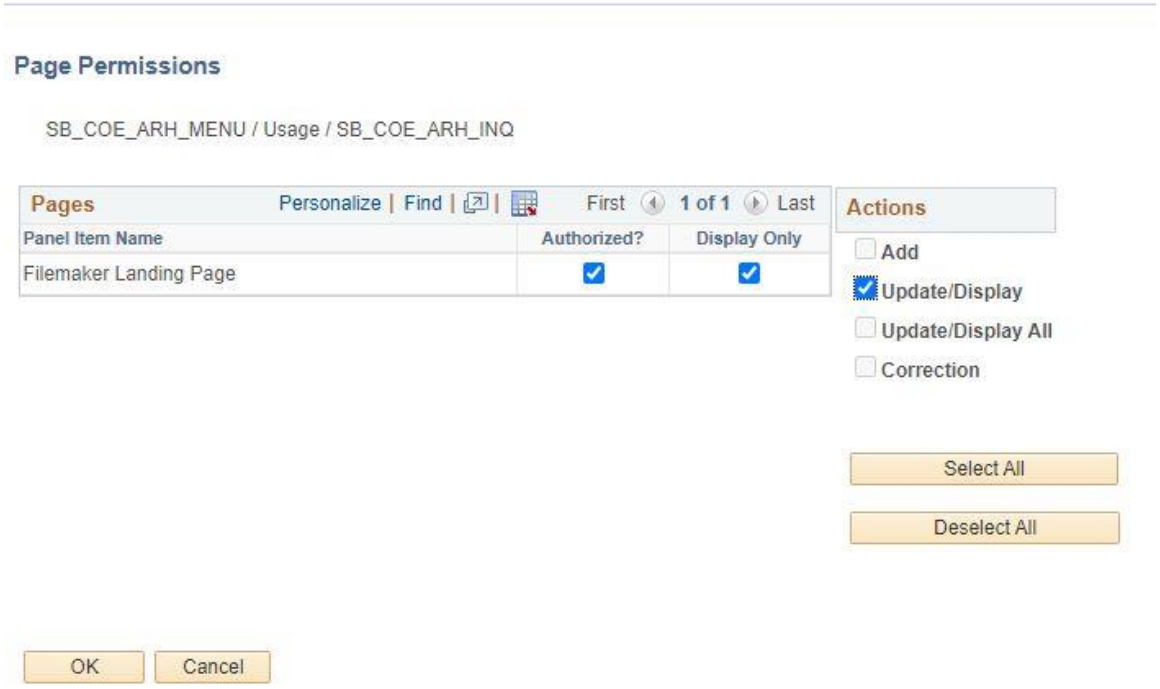


Figure 26. PeopleSoft Setting Page Permissions

4.2.7 Navigation Path and Settings

Following creation of the role and the appropriate permissions, the next step is to create the navigation path to the pages. Once the path has been created, the permissions settings will appear.

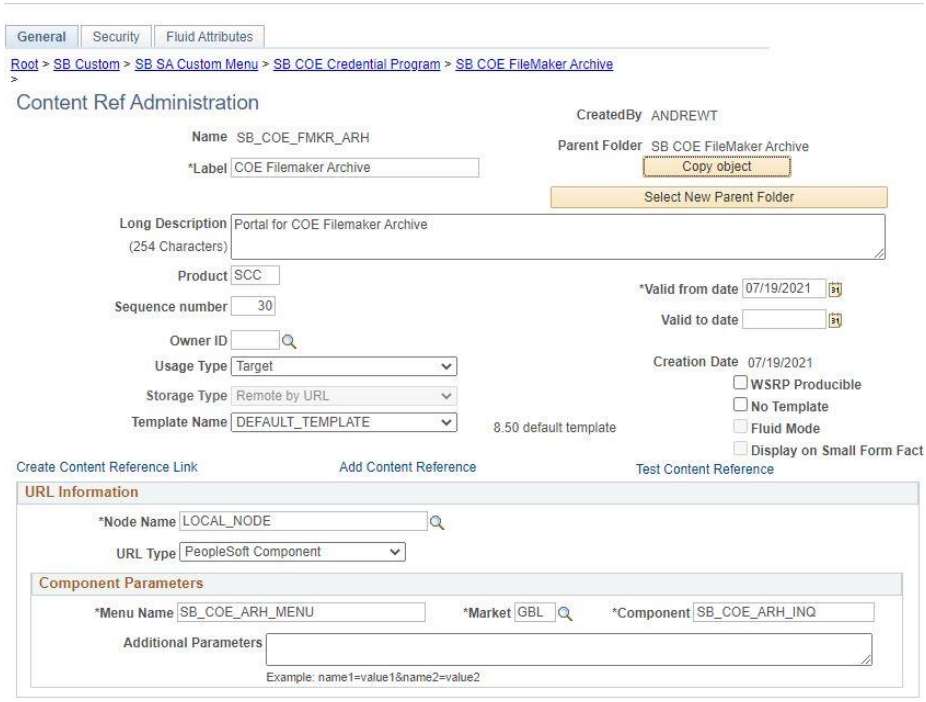


Figure 27. Navigation Path Setup

Figure 28. Content Reference Navigation Path Security

4.2.8 The Query Manager

The last step is to create queries for each table (CRED, EXAM, and NOID) within the Query Manager. The Query Manager allows users with administrator or developer privileges to directly run a query on each of the tables through the PeopleSoft interface. The project has the ability to search by student ID, and additional search criteria can be added in the future (e.g. by year, by cohort, etc.) at the request of the CoE. With the query manager, administrator and developer users can run quality assurance and data validation on the underlying Oracle database tables without requiring special permissions to access the main Oracle database connection. The following figures demonstrate the query manager.

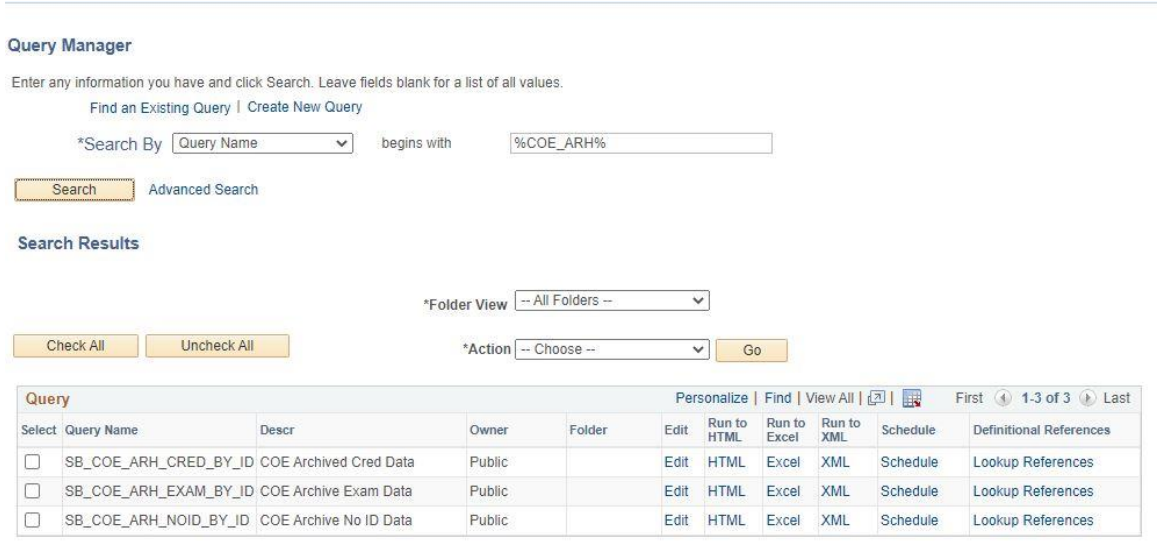


Figure 29. Query Creation with Query Manager

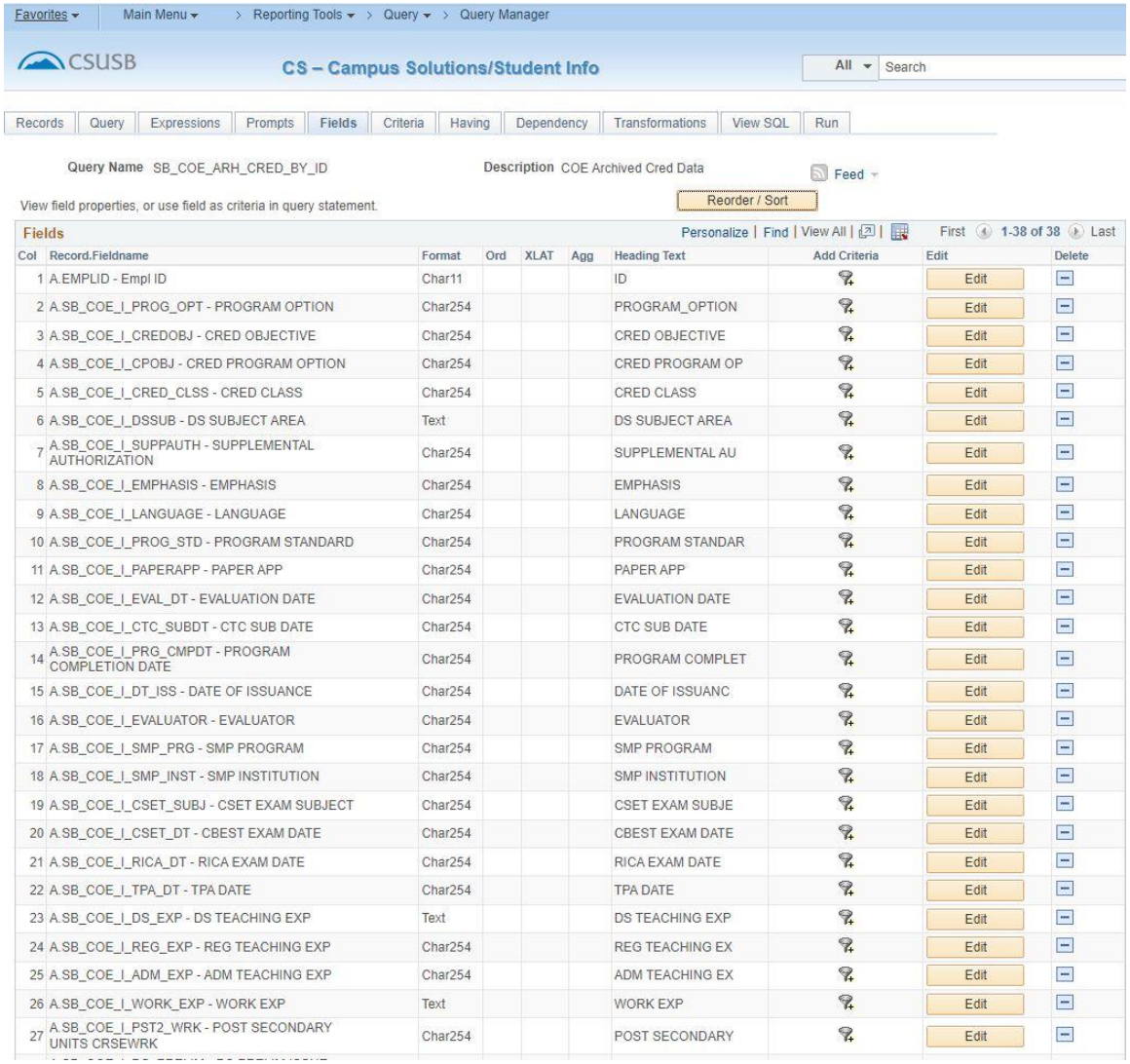


Figure 30. Viewing Underlying Table Structure with Query Manager

Figure 31. Running a Query with Query Manager

4.3 PeopleSoft AppDesigner Items

The End-User Reporting Interface development comprises of 51 fields, five records, four pages, two components, and one menu. The fields have the same properties as their Oracle database counterparts and the remaining AppDesigner objects work as follows:

Table 9. AppDesigner Items and Descriptions

AppDesigner Item	Description
[Record] SB_COE_ARH_CRED	• Record used to store CRED data from the Oracle database
[Record] SB_COE_ARH_EXAM	• Record used to store COMP_EXAM data from the Oracle database
[Record] SB_COE_ARH_NOID	• Record used to store CRED_NOID data from the Oracle database
[Record] SB_COE_ARH_SRCH	• Record used to query student information from the campus's PeopleSoft database.
[Record] SB_COE_ARH_WRK	• Record used as a supplement to SB_COE_ARH_SRCH. • Hosts fields needed to create and customize buttons on the pages
[Page] SB_COE_CRED_CRED	• Page that displays the results of a CRED record search
[Page] SB_COE_CRED_EXAM	• Page that displays the results of an EXAM search
[Page] SB_COE_CRED_LNDN	• Landing page, will display student basic student information if a match is found. • Gives the user an option of viewing the student's CRED data or EXAM data if applicable
[Page] SB_COE_NOID_CRED	• Page that displays the results of a NOID search.

[Component] SB_COE_ARH_INQ	• Packages SB_COE_ARH_SRCH, SB_COE_ARH_CRED, SB_COE_ARH_EXAM, and SB_COE_ARH_LNDN together
[Component] SB_COE_NOID_INQ	• Packages SB_COE_ARH_SRCH, SB_COE_NOID_CRED, and SB_COE_ARH_LNDN together
[Menu] SB_COE_ARH_MENU	• Packages the two components SB_COE_ARH_INQ and SB_COE_NOID_INQ together

CHAPTER FIVE

APPLICATION IMPLEMENTATION

The project was built into the College of Education's specific PeopleSoft portal, acting as an additional module that authorized users can access at any time. This chapter outlines the pages and functions found in the End-User Reporting Interface along with supplemental screenshots of the page layouts. It also describes end-user interaction with the pages, such as possible actions and outcomes, for all logical scenarios the user can expect from the system.

5.1 Main User Login

This page is the entry portal into the new system. Due to the nature and sensitivity of the data being used in the system, it is expected only a few select personnel from the CoE and ACBI departments will have access. Prior to accessing the login page, authorized users will have to sign-in using Cisco AnyConnect, a specialized VPN available through CSUSB's ITS department. Without the VPN, users will have their connections time-out and the page will fail to load. Once the VPN has been setup, authorized users will then have access to the login page.

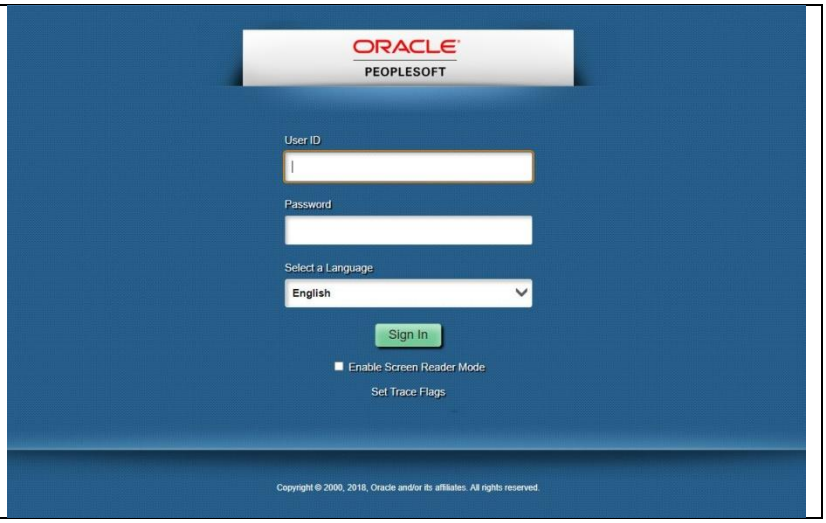
User Action: • User ID • Password Result: • Verifies input credentials • On success, redirect the user to the main page. • On failure, prints a login error message	

Figure 32. User Login Page

5.2 Navigation Folder Path

To access the search page, authorized users will navigate through the file path as shown in the screenshot.

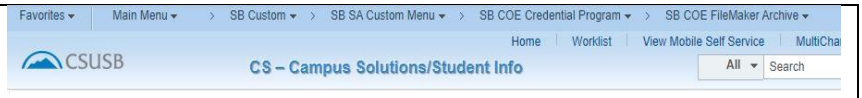
User Action: 1. Main Menu 2. SB Custom 3. SB COE Credential Program 4. SB COE FileMaker Archive Result: • User will be able to navigate to the reporting interface.	

Figure 33. Project Navigation Page

5.3 Search Page

The search page acts as a hub that allows for a user to search for a specific student based on the search criteria inputted. If the search criteria input does not match any student records within the scope of this project, an error message will appear. Users will also have the option to save their searches for future use as well.

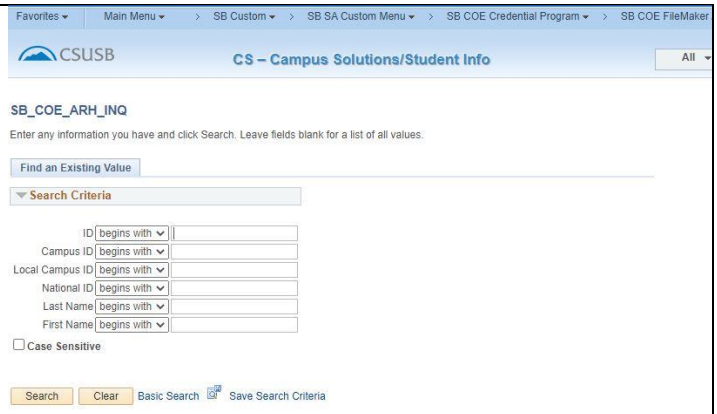
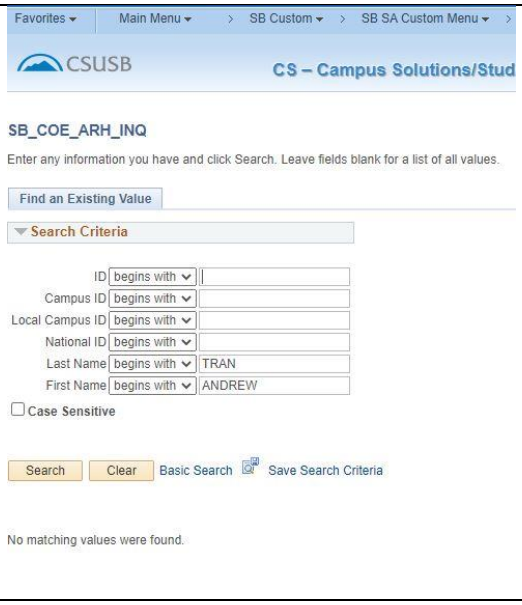
<table>
<tr>
<td>User Input:
ID
Campus ID
Local Campus ID
National ID
Last Name
First Name
User Action:
Basic Search
Save Search Criteria
Delete Saved Search</td>
<td> 
</td>
</tr>
<tr>
<td>User Input Result:
Moves user to landing page on search success
Print error message if the search returns no results
Print error if both first and last names aren’t entered
User Action Result:
Compresses the page
Redirects user to save search page
Deletes all user saved searches</td>
<td> 
</td>
</tr>
</table>

Figure 34. Student Search Page

5.4 Save Search Criteria Page

The function of this page is to save user searches. If the user enters a search criterion on the search page and clicks “Save Search Criteria”, they will be redirected to this page. This page will display the criteria entered and allow the user to name their search to be saved.

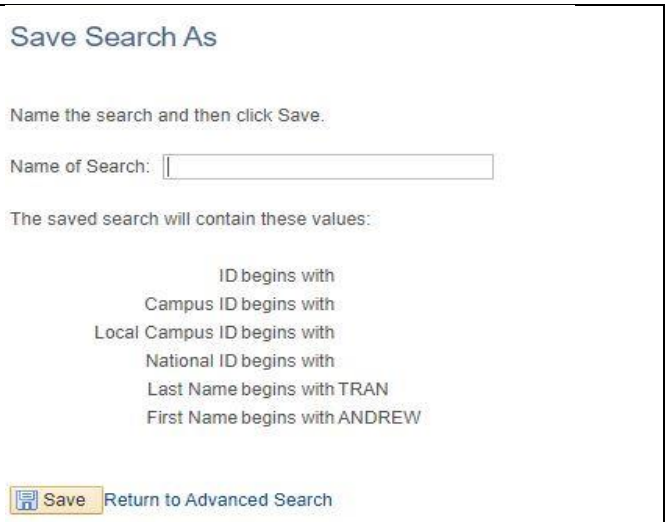
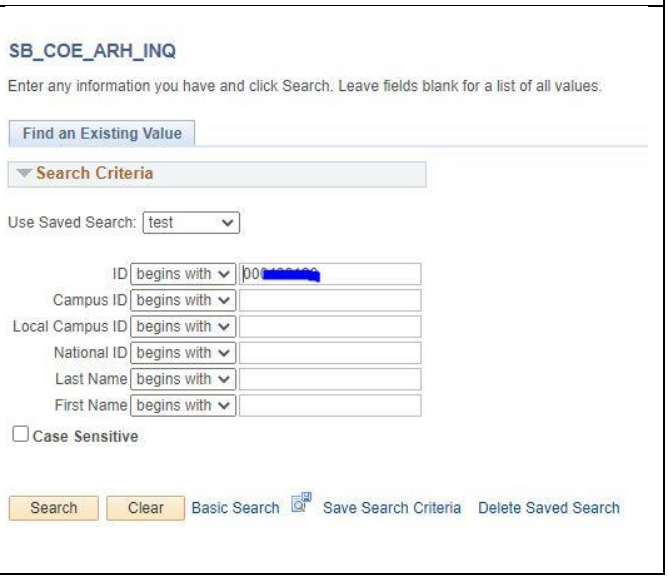
<table>
<tr>
<td>User Action:
• Name of Search</td>
<td> 
</td>
</tr>
<tr>
<td>Result:
• Search is saved, the user will be redirected to the main search page and their saved search will appear in a drop-down menu</td>
<td> 
</td>
</tr>
</table>

Figure 35. Save Search Criteria Page

5.5 Landing Page: Credentials and Comprehensive Exam

This page is landing page once an identified student has been found by the Search Page function in the previous section. It will list the student's Coyote ID, Date of Birth, Names, and Email.

User Action: • "CRED Archive" Button • "EXAM Archive" Button • "Return to Search" Button Result: • System will redirect user to the appropriate page • Allows user to return to the search page	

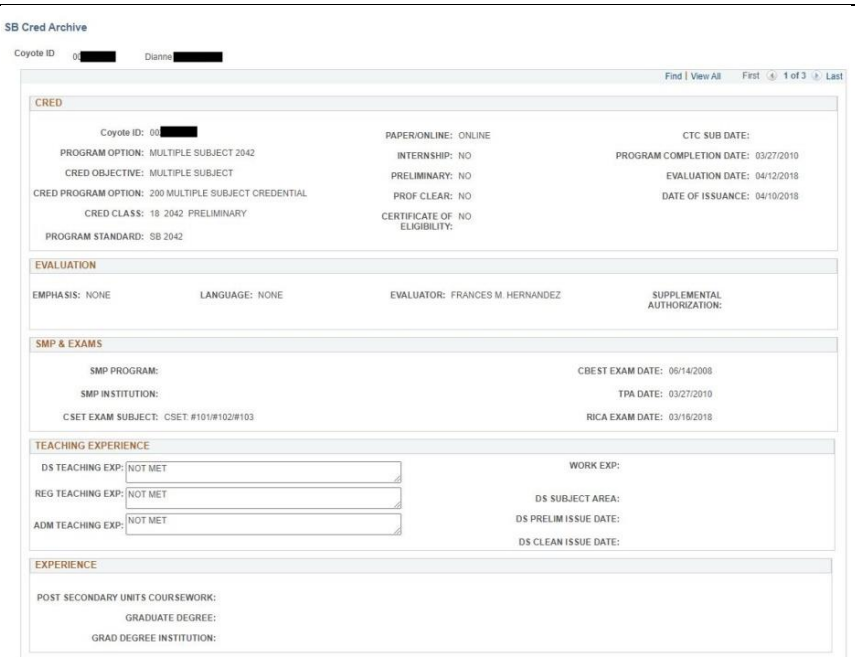
Figure 36. Student Record Landing Page

5.6 Credentials (CRED) Page

This page will display the results of the CRED table. The page is organized so every field within the CRED database will be grouped according to the specifications of the CoE. If the student has multiple distinct records in the CRED table, a set of navigation arrows will allow the user to scroll through each record. This page will be blank if the student has data in the COMP_EXAM table but not the CRED table.

User Actions:

- (If Applicable) GUI Page Navigation Arrows
- “View All” Button
- “Find” Button
- “Return” Button



Result:

- The arrows will allow users to navigate through each distinct student record
- Allows the user to view multiple distinct records (if any) on a single page
- Allows users to search for any string within the page
- Allows users to return to the landing page

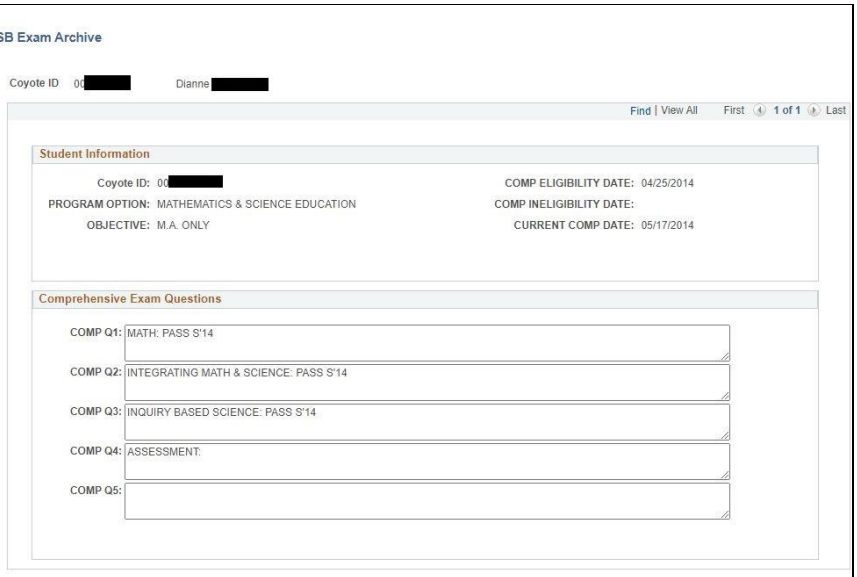
Figure 37. Credentials Reporting Page

5.7 Comprehensive Exam (COMP_EXAM) Page

This page will display the results of the COMP_EXAM table. The page is organized so every field within the COMP_EXAM database will be grouped according to the specifications of the CoE. This page will be blank if the student has data in the CRED table but not the COMP_EXAM table.

User Actions:

- (If Applicable) GUI Page Navigation Arrows
- “View All” Button
- “Find” Button
- “Return” Button



Result:

- The arrows will allow users to navigate through each distinct student record
- Allows the user to view multiple distinct records (if any) on a single page
- Allows users to search for any string within the page
- Allows users to return to the landing page

Figure 38. Comprehensive Exam Reporting Page

5.8 Search Page (NOID)

To access this search page, authorized users will navigate through the folder path seen at the top of the screenshot. As the NOID table is relatively small, this specific search page only has a single field for user input. Users also have the option to save their searches for future use in similar fashion to the other search page.

User Input: • ID User Action: • Basic Search • Save Search Criteria • Delete Saved Search User Input Result: • Moves user to NOID page on search success • Print error message if the search returns no results User Action Result: • Compresses the page • Redirects user to save search page • Deletes all user saved searches	

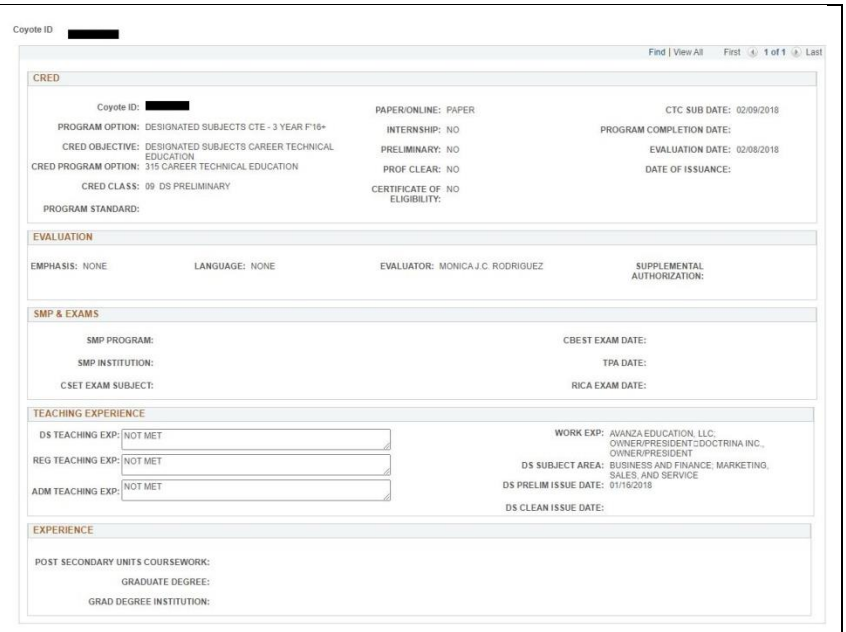
Figure 39. NOID Search Page

5.9 College of Education: CRED without Student ID (CRED_NOID)

This page will display the results of the CRED_NOID table after the user navigates to it. The page is organized so every field within the CRED_NOID database will be grouped according to the CoE's specifications.

User Actions:

- (If Applicable) GUI Page Navigation Arrows
- "View All" Button
- "Find" Button
- "Return" Button



Result:

- The arrows will allow users to navigate through each distinct student record
- Allows the user to view multiple distinct records (if any) on a single page
- Allows users to search for any string within the page
- Allows users to return to the landing page

Figure 40. Credentials with No Student ID

5.10 Logout Page

This function allows the user to logout of their current session. The session will also terminate after a certain amount of page inactivity.

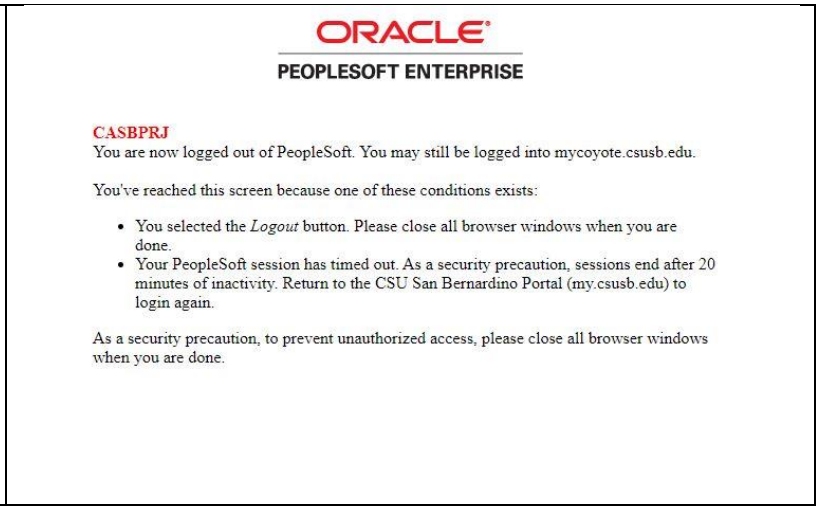
User Action: • User will interact with the “Sign Out” button at the top of the page Result: • User’s session will be closed and this result screen will be displayed	

Figure 41. User Logout Page

CHAPTER SIX

DEPLOYMENT

This chapter is a guide for deploying the new Oracle Database as well as steps needed to develop the End-User Reporting Interface through Oracle PeopleSoft. As per development protocol, this project was developed on a non-production PeopleSoft system. Once the project is approved by CSUSB's ACBI and CoE departments, it will then be migrated to a production system. The project's maintenance and any possible future development will then be taken over by ACBI. Thus, this chapter serves as a reference, detailing steps to assist CSUSB's ACBI developers and database administrator in deployment of this project.

6.1 System Requirements

This project requires two major components, a database system, and an interface development system. The database system runs on Oracle 12c Enterprise Edition and the interface development system runs on PeopleTools 8.57.11.

6.2 Installation

6.2.1 Oracle Database

This section will detail the initialization of database access and the creation of the tables in the database server.

1) Install Oracle referring to the Oracle 12c Installation Guide, which will include Oracle SQL Developer [9].
2) Request ACBI's database administrator (DBA) to create an account with tablespace, request connection credentials, and grant user access to PeopleSoft specific tables.
3) Login to SQL Developer using the user account and connection credentials provided by the DBA in the previous step.
4) Open IMPORT_TABLES.sql to view documentation comments and run the file to create the import tables.
5) Open PRESTAGE_TABLES.sql to view documentation comments and run the file to create the pre-stage tables.
6) Open STAGING_TABLES.sql to view documentation comments and run the file to create the final staging tables
7) Open and run P_STGE_CRED_INSERT.sql, P_STGE_EXAM_INSERT.sql, and P_STGE_NOID_INSERT.sql to view the documentation for each of the procedures that insert from the import tables to the pre-stage tables.
8) Open and run the supplemental procedure file IMP_EXTRDT_UPD.sql that will automatically fill in the EXTRACT_DT field for each import
9) Open and run EXAM_STAGE.sql, CRED_STAGE.sql, CRED_STAGE.sql, and NOID_STAGE.sql to view documentation

for each of the packages that clean up the newly inserted fields in the pre-stage tables.

10) Open and run CLEANUP_EXAM.sql, CLEANUP_CRED.sql, and CLEANUP_NOID.sql to create the procedures that combine the table insertion and field clean-up process into a single procedure
11) Open and run FINAL_EXAM_INSERT.sql, FINAL_CRED_INSERT.sql, and FINAL_NOID_INSERT.sql to create the procedure that performs insertion into the final tale once the pre-staging tables are accepted. These tables act as the source tables that the PeopleSoft tables will pull data from for the End-User Reporting Interface.

6.2.2 Database Migration Process

This section details the steps needed for extraction, import, clean-up, and finalization of the FileMaker data into the new Oracle database. Though the data for the Comprehensive Exam table and the Credentials table have already been completed, these steps can be followed if any additional tables need to be extracted and processed.

6.2.2.1 Extraction Process Steps.

1) Select the date range for the data set to be queried using this format: 01/01/XXXX to 12/31/XXXX+1
2) Once the data set has been found, select the fields needed for extraction and export it to a (.CSV) file

3) From there, convert the (.CSV) file to an (.XLSX) file while adding in the necessary column names for easy source-to-target field matching.
4) Create a sequence number for each record in the (.XLSX) file and drop and records that do not have any identifying information (like Student ID, Last Name, First Name, etc.) as part of initial data clean-up [5].

6.2.2.2 Import Process Steps.

1) Once the (.XLSX) files are ready, open SQL Developer and navigate to the schema list which should be populated with the tables created in the previous section.
2) Right-click on the target import table, and click “Import…”. This will bring up the Data Import Wizard.
3) From the Wizard, the first screen will allow the user to browse to select a source file. Select the file and click “Next”.
4) Continue clicking “Next” through the screens until the page detailing column matching appears. Make sure all the columns are matched correctly and continue through the Wizard until completion and the data is successful inserted.
5) If an error occurs upon finishing the Wizard, is highly likely an issue with the target field’s data-type size. And error message will pop-up indicating the size of the attempted insertion along

with the size of the target column. In these cases, alter the corresponding target column to increase the size.

6) Check the total number of rows inserted and make sure the count in the import table matches the count in the (.XLSX) file.
7) If this is the first import year, the EXTRACT_DT field must be manually updated to match the year. Otherwise, execute IMP_EXTR_DT to automatically update the most recent insertion's EXTRACT_DT field.
8) Repeat the steps above for the import process until all the year-partitioned extraction files have been imported to the database's import tables.

6.2.2.3 Cleanup and Final Insertion Process Steps.

1) Once import is finished, execute CLEANUP_CRED_STAGE, CLEANUP_CRED_STAGE_NOEMP, and CLEANUP_COMP_EXAM_STAGE to initiate data clean-up
2) If successful, data is cleaned and inserted into the respective PRESTAGE tables.
3) Upon failure, an error log will be printed along with student lookup information. The log may then be sent to the CoE for data validation.
4) Once clean-up is finished, execute FINAL_INSERT_CRED, FINAL_INSERT_CRED_NOEMP, and

FINAL_INSERT_COMP_EXAM to move the cleaned data into the final checkpoint before being sent to PeopleSoft

6.2.3 PeopleSoft Application Designer

This section details installation of AppDesigner and initialization of access to CSUSB's PeopleSoft system. PeopleTools, the installation package that includes AppDesigner is licensed to CSUSB, requires a user to be granted clearance through ACBI [10].

1) Request a PeopleSoft account from CSUSB ACBI.
2) Install Cisco AnyConnect using connection credentials after account has been set up
3) Navigate to H:\csusb_shortcuts and double click on INSTALL_sa85711_916.bat
4) Once prompted about the security warning, click "Run".
5) The Setup Wizard will appear on screen. Click "Next"
6) Continue clicking "Next" until you are prompted for your ADMIN credentials, provided by ACBI
7) Click "Finish" to complete installation.
8) Find the pscfg.exe-sa85711_916 shortcut and change the Connect Password to the one the sys admin provided.
9) Click "OK" and AppDesigner is now ready to use.

6.2.4 End-User Reporting Interface Process

This section details the steps needed for creation of the End-User Reporting Interface. Any additional tables added to the Oracle database can be linked to PeopleSoft and their pages can be developed using the following methods.

1) Login to AnyConnect .
2) Open AppDesigner; then open the project using the search string ‘%FMKR%’.
3) Create the necessary fields for matching and add them to the project.
4) Create the necessary records using the fields that were added to the project in step 2. Designate the fields to be used as keys for the table. Add the records to the project.
5) Build the project’s tables. It should be noted that tables built will not appear until the following day due to CSUSB’s system being refreshed nightly.
6) Create the necessary pages using the records created in step 3. The pages are best designed using a “Long Scroll Area” along with “Long Text Box”. Add the pages to the project.
7) Disable update and modification to the pages by other users. Modify the SQL script to join the table with

SB_COE_ARH_SRCH. This will allow the new table to have the student IDs be searchable using CSUSB's student records

8) Create the component that combines all relevant pages. The pages will now be able to communicate and work in tandem with each other. Add the component to the project.
9) Create a menu that combines all components into a single navigation group. Add the menu to the project.
10) Request ACBI's DBA to set up a DB_LINK between the Oracle database and the new PeopleSoft tables.
11) Once the DB_LINK is ready, migrate the data from the STAGING Oracle tables onto the PeopleSoft Tables by executing the procedures: PS_CRED_DML and PS_EXAM_DML
12) Log in to the PeopleSoft web browser from the credentials provided by ACBI
13) Navigate to PeopleTools > Security > Permissions & Roles
14) Add or modify roles and permissions for the new page(s)
15) Navigate to PeopleTools > Portal > Structure and Content
16) Add or Modify the navigation folders and path for the new pages
17) Return to AppDesigner and create a Portal Registry Structure for migration. Add the structure to the project.

CHAPTER SEVEN

CONCLUSIONS

7.1 Project Accomplishments and Impact

With the completion of this project, the College of Education no longer needs to worry about the legacy data stored on FileMaker system becoming lost or corrupted. Credentials Recommended information and Comprehensive Examination information, each critical data that is required to be regularly available by the State of California, was successfully extracted for each student record. As a result, the project served as a cost and time effective solution for the CoE.

This new system also provides better security and accessibility as well. Previously, the legacy data within the FileMaker system was only able to be accessed while the user was on campus. Any authorized user can now access and view the legacy data through the CoE's portal on the campus's PeopleSoft system following the successful migration of the data into the Oracle database. The PeopleSoft system also has access to better security tools and access controls as it is housed within the campus system. At the same time, if the legacy data ever needed to be edited then any authorized users from CSUSB's ACBI department will be able to access to the underlying Oracle database in order to perform data manipulation.

The end-user interface developed using Oracle's AppDesigner allows for easier migration between the CSUSB's various database systems. AppDesigner

packages each component, like data fields, records, page structures, security settings, etc., of the reporting interface into “Projects”. Any potential migration only requires a duplication of the “Project” which can then be imported into the target system.

Development of the end-user interface allowed for development of a clearer and more cohesively grouped interface. Data reports in this new system are cleanly displayed and end-users can quickly find data points. In addition to this, the data cleanup process allowed for a more uniform appearance of the data being reported. Data formats have been standardized and have been made to be more consistent and accurate than the data formats in FileMaker.

Finally, the project allows for openness for modular add-ons. Future developers on this project can replicate the extraction steps, run the Oracle scripts, and follow the AppDesigner instructions to add any supplemental table extractions from FileMaker.

7.2 Personal Skills Development

I have learned and improved my professional database skills working on this project. This includes deepening my understanding of Oracle technologies like PeopleSoft as well as reinforcing my understanding of database design, data migration, and data maintenance. Sharpened my Oracle SQL skills, developed my Oracle PL/SQL skills using real-world data. Development of these two skills has taught me valuable lessons as a future database developer.

In addition, I learnt and studied Oracle's AppDesigner and subsequently, Oracle PeopleSoft. This experience proved to be valuable as many systems, especially higher education campuses, use PeopleSoft. I developed an end-user interface with these tools, which improved my understanding of how prospective end-users may want to interact with a system.

Working on this project also gave me the hands-on experience and practice I needed to pursue Oracle's certifications. Through my experience, my confidence grew, and I was successful in completing a certification from Oracle on database design and data manipulation. This certification, along with my work on this project, has been a motivating factor in me developing my skills further. I look forward to successfully completing additional Oracle certifications soon.

Finally, this project taught me a valuable lesson in developer to client communication. Designing software for a class or for personal use is vastly different from developing software for a client and this project provided a perfect opportunity to develop this skill. While managing the project, I made sure to keep the client consistently updated which allowed the opportunity for constant feedback and input to ensure the end project met expectations. This constant stream of coordination and communication has proved to be beneficial professionally, as it allows both client and developer to always be at or near the mutual levels of understanding.

7.3 Future Directions

Following completion of this project and finalization of the manuscript, management of this project will be overtaken by CSUSB's ACBI department. The department will migrate the project from the non-production system it was developed on onto a production system that will allow the College of Education to fully access it. The data will then be readily available for the CoE should the State of California require a report.

Any potential supplemental tables from FileMaker can be extracted, cleaned, and pushed to the end-user interface using the steps described in the deployment chapter. Any modifications to this project, whether to the database tables or to the end-user reporting interface, can be done by referencing the respective chapters as well. It is my hope that this project will assist the CoE in their analysis and reporting needs as well as providing a system that is both easy to use and access.

APPENDIX A

DATA DICTIONARY

A.1 Import Tables

1. IMP_1_CRED and IMP_1_CRED_NOEMP [4]. These tables are used to hold the raw data from FileMaker's CRED tables.

EMPLID	[VARCHAR2(100 BYTE)]: Student ID Number
LNAME	[VARCHAR2(255 BYTE)]: Student's Last Name
FNAME	[VARCHAR2(255 BYTE)]: Student's First Name
PROGRAM_OPTION	[VARCHAR2(255 BYTE)]: The credential program the applicant is enrolled in
CRED_OBJECTIVE	[VARCHAR2(255 BYTE)]: The California credential the applicant is being evaluated for
CRED_PROGRAM_OPTION	[VARCHAR2(255 BYTE)]: The content and subject area that the California credential applicant is being evaluated for
CRED_CLASS	[VARCHAR2(255 BYTE)]: The data element supplements the term or classification of the credential for which the student has been processed. The student is then assigned a credential class based on the credential that he or she is being recommended for and the total number of requirements that have been met for that credential
DS_SUBJECT_AREA	[VARCHAR2(550 BYTE)]: Designated-Subjects Subject Area. For Designated-Subjects,

	Credential, Credential Content, and Subject Area data
SUPPLEMENTAL_AUTHORIZATION	[VARCHAR2(255 BYTE)]: Additional areas of credential content and subject area(s) to be added to the primary credential identified in CRED_PROGRAM_OPTION
EMPHASIS	[VARCHAR2(255 BYTE)]: Additional authorizations for: English Learner Authorization (CLAD) or instruction in student's native language (BCLAD)
LANGUAGE	[VARCHAR2(255 BYTE)]: The language for additional authorizations in EMPHASIS
PROGRAM_STANDARD	[VARCHAR2(255 BYTE)]: California Commission on Teacher Credentialing (CTC) Program Standard for evaluated or issued credential
PAPER_APP	[VARCHAR2(255 BYTE)]: CTC submission via a paper application
ONLINE_APP	[VARCHAR2(255 BYTE)]: CTC submission via an online application
EVALUATION_DATE	[VARCHAR2(255 BYTE)]: Date of Evaluation for credential Requirements and Recommendation
CTC_SUB_DATE	[VARCHAR2(255 BYTE)]: Date that the Credential Recommendation was submitted to the CTC
PROGRAM_COMPLETION_DATE	[VARCHAR2(255 BYTE)]: Date

	that the student completed all program requirements to be recommended for their credential
DATE_OF_ISSUANCE	[VARCHAR2(255 BYTE)]: Date that the CTC issued the recommended credential to the student
EVALUATOR	[VARCHAR2(255 BYTE)]: Name of the CSUSB Credential Analyst who evaluated the credential application submitted by the student
SMP_PROGRAM	[VARCHAR2(550 BYTE)]: Subject Matter Program (SMP) Content area for students who completed an approved undergraduate program instead of the California Subject Examinations for Teachers (CSET) to demonstrate subject matter competency
SMP_INSTITUTION	[VARCHAR2(255 BYTE)]: Institution where the student completed an approved SMP
CSET_EXAM_SUBJECT	[VARCHAR2(255 BYTE)]: Date that the student passed all sections of the CSET to demonstrate subject matter competency. This field is for students who did not complete an approved SMP
CBEST_EXAM_DATE	[VARCHAR2(255 BYTE)]: Date that the student passed all sections of the California Basic

	Educational Skills Test (CBEST)
RICA_EXAM_DATE	[VARCHAR2(255 BYTE)]: Date that the student passed all sections of the California Reading Instruction Competency Assessment (RICA)
TPA_DATE	[VARCHAR2(255 BYTE)]: Date that the student passed all sections of the California Teaching Performance Assessment (TPA)
DS_TEACHING_EXP	[VARCHAR2(550 BYTE)]: Designated Subjects Teaching Experience For Designated Subjects students and their history of teaching experience
REG_TEACHING_EXP	[VARCHAR2(255 BYTE)]: List of teaching experiences and demonstration of teaching experience
ADM_TEACHING_EXP	[VARCHAR2(255 BYTE)]: Administrative Services Credentials and demonstration of teaching experience
WORK_EXP	[VARCHAR2(500 BYTE)]: Designated Subjects Credentials Demonstration of work-related experience to DS_SUBJECT_AREA
POST_SECONDARY_UNITS_COURSEWORK	[VARCHAR2(255 BYTE)]: Designated Subject Credentials Demonstration of post-

	secondary courses completed
DS_PRELIM_ISSUE_DATE	[VARCHAR2(255 BYTE)]: Date that the Designated Subjects Preliminary Credential was issued by the CTC
DS_CLEAR_ISSUE_DATE	[VARCHAR2(255 BYTE)]: Date that the Designated Subjects Clear Credential was issued by the CTC
INTERNSHIP	[VARCHAR2(255 BYTE)]: Marks whether the student was issued an INTERN credential by the CTC
PRELIMINARY	[VARCHAR2(255 BYTE)]: Marks whether the student was issued a PRELIMINARY credential by the CTC
PROF_CLEAR	[VARCHAR2(255 BYTE)]: Marks whether the student was issued a CLEAR credential by the CTC
GRADUATE_DEGREE	[VARCHAR2(255 BYTE)]: The content or subject area of graduate degree that the student has previously received or is currently completing
GRAD_DEGREE_INSTITUTION	[VARCHAR2(255 BYTE)]: Institution at which the student has or is completing a graduate degree
CERTIFICATE_OF_ELIGIBILITY	[VARCHAR2(255 BYTE)]: Marks whether the student was issued a CERTIFICATE OF ELIGIBILITY by the CTC
IMPORT_DATE	[DATE]: Date of the import of the FileMaker extraction into

	the Oracle database
IMPORT_STATUS	[VARCHAR2(1 BYTE)]: Marks whether or not the record has been moved to the PRESTAGE tables or not yet.
CRED_SEQ	[NUMBER(10,0)]: Assigns each student record a unique sequence number per extraction date
EXTRACT_DT	[VARCHAR2(11 BYTE)]: Identifies which year the extraction data set belongs to

2. IMP_2_COMP_EXAM.

EMPLID	[VARCHAR2(3000 BYTE)]: Student ID Number
AY_RECORD_CREATED	[VARCHAR2(255 BYTE)]: Used to identify which year the extraction data set belongs to
PROGRAM_OPTION	[VARCHAR2(255 BYTE)]: The subject or discipline area of graduate degree that the student is seeking
OBJECTIVE	[VARCHAR2(255 BYTE)]: Marks whether the student seeking a graduate degree only or a graduate degree and a credential simultaneously
COMP_ELIGIBILITY_DATE	[VARCHAR2(255 BYTE)]: Date that the student has met all program requirements to be eligible to take their comprehensive exam
COMP_INELIGIBILITY_DATE	[VARCHAR2(255 BYTE)]: Date of evaluation which determined if the student has not met all program requirements to take their comprehensive exam
CURRENT_COMP_DATE	[VARCHAR2(255 BYTE)]: Date the student passed all sections of their

	comprehensive exam
COMP_Q1	[VARCHAR2(255 BYTE)]: Comprehensive Exam Question 1 Content Area that this student completed or attempted. Subject and content varies across programs
COMP_Q2	[VARCHAR2(255 BYTE)]: Comprehensive Exam Question 2 Content Area that this student completed or attempted. Subject and content varies across programs
COMP_Q3	[VARCHAR2(300 BYTE)]: Comprehensive Exam Question 3 Content Area that this student completed or attempted. Subject and content varies across programs
COMP_Q4	[VARCHAR2(255 BYTE)]: Comprehensive Exam Question 4 Content Area that this student completed or attempted. Subject and content varies across programs
COMP_Q5	[VARCHAR2(255 BYTE)]: Comprehensive Exam Question 5 Content Area that this student completed or attempted. Subject and content varies across programs
IMPORT_DATE	[DATE]: Date of the import of the

	FileMaker extraction into the Oracle database
IMPORT_STATUS	[VARCHAR2(1 BYTE)]: Marks whether or not the record has been moved to the PRESTAGE tables or not yet
LNAME	[VARCHAR2(255 BYTE)]: Student's Last Name
FNAME	[VARCHAR2(255 BYTE)]: Student's First Name

A.2 Clean-up Tables

All of the following tables have almost the same table structure as the IMPORT tables due to the nature of data migration and clean-up. Fields keys and field size modifications are made with some tables having new fields as well. For the following tables, only size and key status will be noted, unless it is a new field.

3. PRESTAGE_1_CRED and PRESTAGE_1_CRED_NOEMP.

CRED_SEQ	[NUMBER(5), PK]
EMPLID	[VARCHAR2(9 BYTE),PK]
PROGRAM_OPTION	[VARCHAR2(255 BYTE)]
CRED_OBJECTIVE	[VARCHAR2(255 BYTE)]
CRED_PROGRAM_OPTION	[VARCHAR2(255 BYTE)]
CRED_CLASS	[VARCHAR2(255 BYTE)]
DS_SUBJECT_AREA	[VARCHAR2(550 BYTE)]
SUPPLEMENTAL_AUTHORIZATION	[VARCHAR2(255 BYTE)]
EMPHASIS	[VARCHAR2(255 BYTE)]
LANGUAGE	[VARCHAR2(255 BYTE)]

PROGRAM_STANDARD	[VARCHAR2(255 BYTE)]
PAPER_ONLINE_APP	[VARCHAR2(255 BYTE)]
EVALUATION_DATE	[VARCHAR2(255 BYTE)]
CTC_SUB_DATE	[VARCHAR2(255 BYTE)]
PROGRAM_COMPLETION_DATE	[VARCHAR2(255 BYTE)]
DATE_OF_ISSUANCE	[VARCHAR2(255 BYTE)]
EVALUATOR	[VARCHAR2(255 BYTE)]
SMP_PROGRAM	[VARCHAR2(255 BYTE)]
SMP_INSTITUTION	[VARCHAR2(255 BYTE)]
CSET_EXAM_SUBJECT	[VARCHAR2(255 BYTE)]
CBEST_EXAM_DATE	[VARCHAR2(255 BYTE)]
RICA_EXAM_DATE	[VARCHAR2(255 BYTE)]
TPA_DATE	[VARCHAR2(255 BYTE)]
DS_TEACHING_EXP	[VARCHAR2(999 BYTE)]
REG_TEACHING_EXP	[VARCHAR2(255 BYTE)]
ADM_TEACHING_EXP	[VARCHAR2(255 BYTE)]
WORK_EXP	[VARCHAR2(500 BYTE)]
POST_SECONDARY_UNITS_COURSEWORK	[VARCHAR2(255 BYTE)]
DS_PRELIM_ISSUE_DATE	[VARCHAR2(255 BYTE)]
DS_CLEAR_ISSUE_DATE	[VARCHAR2(255 BYTE)]
INTERNSHIP	[VARCHAR2(255 BYTE)]
PRELIMINARY	[VARCHAR2(255 BYTE)]
PROF_CLEAR	[VARCHAR2(255 BYTE)]
GRADUATE_DEGREE	[VARCHAR2(255 BYTE)]
GRAD_DEGREE_INSTITUTION	[VARCHAR2(255 BYTE)]

CERTIFICATE_OF_ELIGIBILITY	[VARCHAR2(255 BYTE)]
CLEAN_STATUS	[VARCHAR2(1 BYTE)]
IMPORT_DATE	[DATE]
LNAME	[VARCHAR2(255 BYTE)]
FNAME	[VARCHAR2(255 BYTE)]
EXTRACT_DT	[VARCHAR2(10 BYTE),PK]
STAGING_IMP	[VARCHAR2(1 BYTE)] Used to indicate whether the cleaned data has been moved to the last stage of the pipeline yet

4. PRESTAGE_2_COMP_EXAM.

EMPLID	VARCHAR2(9 BYTE)
AY_RECORD_CREATED	VARCHAR2(255 BYTE)
PROGRAM_OPTION	VARCHAR2(255 BYTE)
OBJECTIVE	VARCHAR2(255 BYTE)
COMP_ELIGIBILITY_DATE	VARCHAR2(255 BYTE)
COMP_INELIGIBILITY_DATE	VARCHAR2(255 BYTE)
CURRENT_COMP_DATE	VARCHAR2(255 BYTE)
COMP_Q1	VARCHAR2(255 BYTE)
COMP_Q2	VARCHAR2(255 BYTE)
COMP_Q3	VARCHAR2(255 BYTE)
COMP_Q4	VARCHAR2(255 BYTE)
COMP_Q5	VARCHAR2(255 BYTE)
CLEAN_STATUS	VARCHAR2(1 BYTE)
IMPORT_DATE	DATE
LNAME	VARCHAR2(255 BYTE)

FNAME	VARCHAR2(255 BYTE)
STAGING_IMP	[VARCHAR2(1 BYTE)] Used to indicate whether the cleaned data has been moved to the last stage of the pipeline yet

A.3 Final Tables

5. STAGING_1_CRED and STAGING_1_CRED_NOEMP.

CRED_SEQ	[NUMBER(5), PK]
EMPLID	[VARCHAR2(9 BYTE),PK]
PROGRAM_OPTION	[VARCHAR2(255 BYTE)]
CRED_OBJECTIVE	[VARCHAR2(255 BYTE)]
CRED_PROGRAM_OPTION	[VARCHAR2(255 BYTE)]
CRED_CLASS	[VARCHAR2(255 BYTE)]
DS_SUBJECT_AREA	[VARCHAR2(550 BYTE)]
SUPPLEMENTAL_AUTHORIZATION	[VARCHAR2(255 BYTE)]
EMPHASIS	[VARCHAR2(255 BYTE)]
LANGUAGE	[VARCHAR2(255 BYTE)]
PROGRAM_STANDARD	[VARCHAR2(255 BYTE)]
PAPER_ONLINE_APP	[VARCHAR2(255 BYTE)]
EVALUATION_DATE	[VARCHAR2(255 BYTE)]
CTC_SUB_DATE	[VARCHAR2(255 BYTE)]
PROGRAM_COMPLETION_DATE	[VARCHAR2(255 BYTE)]
DATE_OF_ISSUANCE	[VARCHAR2(255 BYTE)]
EVALUATOR	[VARCHAR2(255 BYTE)]
SMP_PROGRAM	[VARCHAR2(255 BYTE)]
SMP_INSTITUTION	[VARCHAR2(255 BYTE)]

CSET_EXAM_SUBJECT	[VARCHAR2(255 BYTE)]
CBEST_EXAM_DATE	[VARCHAR2(255 BYTE)]
RICA_EXAM_DATE	[VARCHAR2(255 BYTE)]
TPA_DATE	[VARCHAR2(255 BYTE)]
DS_TEACHING_EXP	[VARCHAR2(999 BYTE)]
REG_TEACHING_EXP	[VARCHAR2(255 BYTE)]
ADM_TEACHING_EXP	[VARCHAR2(255 BYTE)]
WORK_EXP	[VARCHAR2(500 BYTE)]
POST_SECONDARY_UNITS_COURSEWORK	[VARCHAR2(255 BYTE)]
DS_PRELIM_ISSUE_DATE	[VARCHAR2(255 BYTE)]
DS_CLEAR_ISSUE_DATE	[VARCHAR2(255 BYTE)]
INTERNSHIP	[VARCHAR2(255 BYTE)]
PRELIMINARY	[VARCHAR2(255 BYTE)]
PROF_CLEAR	[VARCHAR2(255 BYTE)]
GRADUATE_DEGREE	[VARCHAR2(255 BYTE)]
GRAD_DEGREE_INSTITUTION	[VARCHAR2(255 BYTE)]
CERTIFICATE_OF_ELIGIBILITY	[VARCHAR2(255 BYTE)]
CLEAN_STATUS	[VARCHAR2(1 BYTE)]
IMPORT_DATE	[DATE]
LNAME	[VARCHAR2(255 BYTE)]
FNAME	[VARCHAR2(255 BYTE)]
EXTRACT_DT	[VARCHAR2(10 BYTE),PK]
PS_IMPORT	[VARCHAR2(1 BYTE)] Used to indicate whether the cleaned data has been moved to PeopleSoft

6. STAGING_2_COMP_EXAM.

EMPLID	VARCHAR2(9 BYTE)
AY_RECORD_CREATED	VARCHAR2(255 BYTE)
PROGRAM_OPTION	VARCHAR2(255 BYTE)
OBJECTIVE	VARCHAR2(255 BYTE)
COMP_ELIGIBILITY_DATE	VARCHAR2(255 BYTE)
COMP_INELIGIBILITY_DATE	VARCHAR2(255 BYTE)
CURRENT_COMP_DATE	VARCHAR2(255 BYTE)
COMP_Q1	VARCHAR2(255 BYTE)
COMP_Q2	VARCHAR2(255 BYTE)
COMP_Q3	VARCHAR2(255 BYTE)
COMP_Q4	VARCHAR2(255 BYTE)
COMP_Q5	VARCHAR2(255 BYTE)
CLEAN_STATUS	VARCHAR2(1 BYTE)
IMPORT_DATE	DATE
LNAME	VARCHAR2(255 BYTE)
FNAME	VARCHAR2(255 BYTE)
STAGING_IMP	[VARCHAR2(1 BYTE)] Used to indicate whether the cleaned data has been moved to PeopleSoft

APPENDIX B

APPLICATION PROGRAM FILES

B.1 Application Program Files

IMPORT_EXTRCTDT_UPDT	Fills in the EXTRACT_DT field with the appropriate year extraction based on student ID and CRED_SEQ.
STAGING_INSERT_CRED	Modifies field using regular expressions for pattern matching. Modifies field data contents according to requests from the College of Education. Inserts fields from IMPORT CRED table to PRESTAGE CRED table. Updates flag from the IMPORT table from “N” to “Y”.
STAGING_INSERT_COMP_EXAM	Modifies field using regular expressions for pattern matching. Modifies field data contents according to requests from the College of Education. Inserts fields from IMPORT COMP_EXAM table to PRESTAGE COMP_EXAM table. Updates flag from the IMPORT table from “N” to “Y”.
CLEANUP_CRED_STAGE	Calls STAGING_INSERT_CRED, CRED_FILL_ID, MET_TEACHING_EXP, CRED_CLEAN_Y
CLEANUP_COMP_EXAM_STAGE	Calls STAGING_INSERT_COMP_EXAM, COMP_EXAM_CLEAN_Y
FINAL_INSERT_CRED	Inserts records from PRESTAGE CRED to STAGING CRED where the flag STAGING_IMP is “N” Updates PRESTAGE CRED’s

	STAGING_IMP field from “N” to “Y”
FINAL_INSERT_COMP_EXAM	Inserts records from PRESTAGE COMP_EXAM to STAGING COMP_EXAM where the flag STAGING_IMP is “N”
	Updates PRESTAGE COMP_EXAM’s STAGING_IMP field from “N” to “Y”
COMP_EXAM_STAGE.COMP_EXAM_CLEAN_Y	Marks flag for data clean-up as “Y” after process finishes.
CRED_STAGE.CRED_FILL_ID	Fills in any blank student IDs with the most recent student ID.
CRED_STAGE.MET_TEACHING_EXP	Corrects the fields DS_TEACHING_EXP, REG_TEACHING_EXP, ADM_TEACHING_EXP based on specific criteria supplied by Dr. Mahoney
CRED_STAGE.CRED_CLEAN_Y	Marks flag for data clean-up as “Y” after process finishes.
PS_CRED_DML	Inserts records from STAGING CRED the PeopleSoft CRED table where PS_IMPORT = “N”
	Updates PS_IMPORT to “Y” for these new inserts
PS_EXAM_DML	Inserts records from STAGING COMP_EXAM the PeopleSoft EXAM table where PS_IMPORT = “N”
	Updates PS_IMPORT to “Y” for these new inserts

REFERENCES

[1] J. Morris, Practical data migration. London: BCS Learning and Development, 2012.

[2] Oracle Corporation, “Successful Data Migration,” Nov. 2011

[3] P. Koletzke and P. Dorsey, Oracle Designer Handbook. Berkeley, CA: Osborne/McGraw-Hill, 1999.

[4] M. Mahoney, "FM_Legacy_Field_Descriptions", 2021.

[5] M. Mahoney, “FileMaker_Extraction_Description”, 2021.

[6] L. Morin, “Oracle Database PL/SQL Language Reference, 12c Release 1 (12.1),” May 2017.

[7] Oracle Corporation, “PeopleTools 8.52: PeopleSoft Application Designer Developer’s Guide,” Oct. 2011.

[8] CSUSB ACBI, *CASBPRJ PeopleSoft Database*

[9] P. Jashnani, “Oracle Database Installation Guide,” Jul. 2017.

[10] L. Tran, “Instructions to Install PS85711,” Nov. 2021.

[11] Y. Chiang, “GRADS: Advising Module for Computer Science Graduate Program,” California State University, San Bernardino, 2005.

[12] Y. Chiang, “Data Migration and Clean-Up Pipeline,” Jul-2021.

[13] J. Murphy, “Library Guides: IEEE Referencing: Getting started with IEEE referencing,” 2015.

[14] A. Tran, “Divining Rod Software Requirements Specification,” Mar. 2020.