Design And Implementation Of A Computerized System For A Manual Artisan Office

by

Jeanie L. Jester Gay

B.S. Kansas State University, 1986

A MASTER'S REPORT

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1987

Approved by:

Dr. Elizabeth A. Unger

Table of Contents

Table of Figures

Chapter 1: Introduction

The quality of placement programs at universities and colleges determines to a great extent the types of job opportunities their graduates are afforded. Such programs are a selling point for potential students. But with budget constraints many universities are working under, a good computer placement program is essential, for it can offer an effective way to deal with cutbacks which are seemingly continually faced. A good computer placement program can offer many advantages as well to students, universities and recruiting companies. Eliminating a great deal of paper shuffling frees up more staff time for tasks that require human judgement. Entering information in a central location can eliminate data duplication and lower the chances of scheduling and omission errors. A quicker response to inquiries can mean more use of the placement center by both students and interviewing companies.

The Career Planning and Placement Center, at Kansas State University, hereafter referred to as the Center, contacted us asking if such a program could be specified functionally and then acquired, or designed and programmed.

A functional evaluation was accomplished, following this several existing placement programs were reviewed and critiqued. The result of this phase of our investigation resulted in the design and implementation of a placement system. The project was too large to be accomplished by one individual and as a result two of us. Cathy Puzzuoli and Jeanie Gay were asked to work on various phases of the project. This report describes my part which includes the design decisions, most of the implementation, and user documentation. Some background given throughout this report is from Puzzuoli's paper [PUZZ87].

To comprehend the scope of the project, the reader must understand what is meant by an

artisan office since that form of office was a significant part of the focus of this work. The concept of an artisan office is defined in section 1.0. As further background we provide in this introduction a discussion of the existing placement programs in section 1.1.

1.1 Artisan Office

A discussion of an artisan office is important at this stage since it gives an indepth understanding of how the office at the Center is organized. An artisan office has little systematic organization. The individuals in this office work independently from each other and each has his own task which is generally taken to completion [CHA80]. There is a wide variety of tasks and therefore a wide variety of different styles of work. Often, at the present, there is little technology in this type of office.

1.2 The Center's Requirements

The functional analysis resulted in the requirements that the placement program must meet. These consisted of: the storage and retrieval of student, company and employment information. Various calculations that are needed include: recruiter statistics, percent of registered students, company statistics, placement by curriculum statistics, and geographical placement statistics. Professionally formated printouts were also a specification; these included listings for: perspective employers, a master visit list, a posting schedule, students by curriculum, students by GPA, students by date available for employment, students by location for employment. Once the requirements were known, a search ensued to find a system to meet the Center's needs, this is discussed in the next section.

1.3 Extant Systems

An extensive search revealed several possible systems. In this report we will review

five of them, Iowa State' System, VitaQuik, Michigan State's system, Colorado State University's system, and the Spartin System.

1.3.1 Iowa State System

Although the Iowa State Universitys resume system used dBase III to develop their program in 1985, and the hardware requirement met that of the Placement Center, it only gathered data about the student [COL85]. The Placement Center required information about the companies interviewing, as well information necessary for scheduling interviews and various statistical purposes.

1.3.2 VitaQuik

VitaQuik, the CIS (Candidate Information System) [CPC84] was developed by a task group of the Midwest College Placement Association. Data for each student is compiled on a one page format in the form of a resume. The menu driven system can store up to 6,000 students. It can sort, select, transmit and print candidate information. If there are enough terminals students can enter there own information. The problem, as was the case with the Iowa State program, is that it didn't gather any employer information or perform any statistical analysis. Also, data entry was clumsy and too slow for the placement center personnel.

1.3.3 SIGI

Extensive statistics are offered by the Michigan State University program. The SIGI (System of Interactive Guidance and Information) developed by the Career Information Center [MIC84], which has one of the largest placement programs in the nation. Because of its vast statistical and reporting

abilities, the program utilizes a mainframe. While this program offered excellent abilities, it was felt that this program was to large for the Placement Center's needs, as well as exceeding the hardware limitations.

1.3.4 CSU, ISS

Colorado State University had a newer placement program called the CSU-ISS (Colorado State University - Interviewing Scheduling System) [COL84], that was developed in 1984. It provides an excellent interview and scheduling system, but lacks the statistical capability our placement center required. Another drawback was the high cost of the software. In addition to purchasing their program an additional purchase of R:base 5000, a relational database management system, was required. The software itself contained too many menus and required extensive input forms for data entry.

1.3.5 Spartin

The Spartin system, developed in 1985, is a placement program called the U-Place (University Placement System). This program was difficult to load, not very user friendly and inputing data was very clumsy. In addition, the program is still under development. The software does accommodate storage of student and employee information. It could also be downloaded from the mainframe [BRU86]. Because it was still in the developmental stage, and the fact that it did not generate the desired statistics, it was also rejected.

1.4 Possible Data Base Management Systems

Since none of the reviewed software programs were exactly what the Center wanted, developing there own became an option. In light of this, two commercial data base management systems were scrutinized, R:base 5000, and the dBase product line.

1.4.1 R:base 5000.

R:base 5000, was found to be very flexible. These included benefits such as individualized application programs, and programming features like its own complier and an application generator. It contained the relational commands, JOIN, APPEND, INTERSECT, UNION, PROJECT, and SUBTRACT for data manipulation [POO85]. Its sort time was found to be very fast and data could be input directly from several popular PC software packages, like Lotus 123. Password protection was provided. R:base 5000 allows up to 400 database fields, a maximum record size of 1530 characters. It can be installed on an IBM compatible computer with a hard disk drive. Though all this made R:base 5000 appealing, the real disadvantages are the $700 cost of the software package along with the need to be knowledgeable about its procedural language for use.

1.4.2 The dBase Line.

dBase provides individual application programs, an internal language is structured and therefore easier to write, and more powerful with single commands for data manipulation. Custom screen formating is also easily accomplished. On-line help, command prompting, and macros are also advantages of dBase. dBase can handle 128 database fields. It can be installed on an IBM compatible computer with a hard disk. The center

already had a copy of dBase III, therefore there was no real cost for the software.

After considerable evaluation and discussion with the end users, it was decided the best path to select was to create a system for the center. The dBase line was chosen for the implementation of the placement program. Its application development features and user-friendliness, as well as its top down design style of programming made it a good choice. It also had adequate storage capabilities, was IBM compatible and as mentioned, required no further expenditure for the software.

This report is organized into four remaining chapters which describe the remaining aspects of the project which cover my responsibilities. Chapter 2 discusses the system design, Chapter 3 discusses the implementation. Chapter 4 includes what is left of the designed to be implemented, and Chapter 5 discusses some future extensions.

Chapter 2: Design of the System

The design of this system commenced with a thorough analysis of the flow of information in the center using a form analysis (one form of input/output analysis). Based upon this analysis, formal data dependencies were created and used in the synthetic algorithm for design of entities, i.e., Bernstein's Algorithm 2. Integrate to the this process, an enterprise view of the data base was created using Chen's E-R diagram as a tool [CHEN]. Finally, a relational schema was created, which was based on the final E-R diagram. This chapter describes these steps in the design process.

2.1 Form Analysis

A form analysis was performed on the input and output documents used by the Center as this is what they relied upon to gather student and company information. Six basic forms are used by the Center: Interview Arrangement Questionnaire, Visit Establishment Sheet, Master Visit List, Student Data Form, Interview Request, and the Employment Report. These forms give the Center the information to set up interview schedules, and compile statistics that are vital to their mission.

An interested company returns a completed Interview Arrangement form (IAQ) see figure 2.1. This form provides the name of the organization, it's address, the number of interviews it would grant, and what type of graduates it is interested in. The company is then asked to fill out the Visit Date Establishment Sheet (DES) which gives a detailed account of the number and dates of the interviews, see figure 2.2. The Master Visit List is a compilation of these forms, see figure 2.3. It is generated at the beginning of each semester. If an employer needed to make changes to the Visit Date Establishment Sheet, a Visit or Schedule Update would be used, see figure 2.4.

A Posting Schedule is posted daily to include any additions or corrections to the MVL, for interested students, see figure 2.5.

All students wanting to use the Center for on campus interviews must fill out the Student Data Sheet, see figure 2.6. This form includes: the student's name, personal data, job interests, educational and some occupational data. When students meet the qualifications of an interviewing company, they sign up on an Interview Request Form and attach a copy of their Student Data Form, see figure 2.7. The company reviews the applicants and selects those it wishes to interview. When a student gets a job or, alternately, at the end of the semester, they are asked to fill out an Employment Report, see figure 2.8. The information from that form gives: the students name, employer (if employed), curriculum, degree, job information, and data about what role the Center played in obtaining interviews/job offers.

The correlation between these forms and the corresponding parties is given in figure 2.9, the form analysis chart.

KANSAS STATE UNIVERSITY

AC 913 532-6506 Career Planning and Placement Center Holtz Hall Manhattan, KS 66506

INTERVIEW ARRANGEMENTS QUESTIONNAIRE (IAQ)

(Please Type)

Interview days and dates for which this IAQ is applicable: _____

CAUTION: Cover more than one date or sequence only if your requirements will be absolutely constant (additional IAQ blanks available upon request for different schedules)

Name of Organization _____

Complete Address _____

No. of daily interview schedules: _____ Name of coordinator: _____

No. of interviewers if more _____ Title: _____

Daily interview start time _____ Phone: AC _____

Interviews to conclude by _____ Length of each interview: _____ minutes

Name(s) of interviewer(s): _____

Inviting graduates of December (19__) ____ May (19__) ____ August (19__) ____ December (19__) ____

Wish to interview for summer employment (classification indicates year completed):

Freshman ____ Sophomore ____ Junior ____ Senior ____ Graduate Students ____ Not at all ____

NOTE: Explain on an attached sheet or by transmittal letter your request for separate summer schedule, mixed regular schedule, or summer group meetings, etc.

International candidates interviewed: ____ without restriction ____ holding permanent resident status

____ U.S. citizens only ____ cannot be interviewed on campus

Comments

Type of industry: _____

Location(s) of work: _____

For curricula and degree combinations invited, see: ☐ reverse ☐ attached

List types of positions available, special qualifications required, and indicate if openings are deferred or speculative:

Literature for student distribution ☐ is being sent ☐ is not available

It is the policy of the Career Planning and Placement Center of Kansas State University to serve only equal opportunity employers. We also are required by Public Law 93-380 to remind employers that the Family Educational Rights and Privacy Act compels the handling of employment records on a confidential basis. By completing and returning this IAQ form the employer publicly affirms full compliance with the law

_____

This IAQ has been prepared on (date) _____ by _____

Correspondence concerning this visit should be directed to: _____

_____

This will be used to publicize your visit and will be attached to your sign-up sheet _____

BLUE COPY—Send to K-State
YELLOW COPY—Employer to retain

IAQ Page 1

Figure 2.1: Interview Arrangement Form

VISIT DATE ESTABLISHMENT SHEET

Organization _____ Date Received _____

_____ Taken by (show initials):

Address _____ Phone ____ Visit ____ LAQ ___

_____ Letter of _____ Other _____

Establisher's Name _____ Tel. (___) _____

Title _____

Confirm to (if other than establisher) _____
Name Title

Address (If Different) _____

_____

scheds. _____ # intvrs. _____ M TU W TH F_____ 86 87 88 89
Month-1st Day

scheds. _____ # intvrs. _____ M TU W TH F_____ 86 87 88 89
Month-2nd Day

scheds. _____ # intvrs. _____ M TU W TH F_____ 86 87 88 89

_____

ACTION REQUIRED	DATE DONE	INITIAL
1. Check Visit Date Card to prevent duplication		
2. Record on Visit Date Card		
3. Record in Visit Date Book (both index and date)		
4. Mail confirming letter standard ___special___		
Their standard form also used ___		
Other dates confirmed on same letter _____		
5. PVM's mailed___with C/L___Later___		
Mail follow-up sent _____(date & initial)		
Telephone follow-up_____(date & initial)		
6. If Visit List has been published enter on Master Visit List		
7. Check Mailing Card existence and/or accuracy		

SHOW # INTERVIEWERS ONLY IF DIFFERENT THAN # OF SCHEDULES THAT DAY
CHECK MARK SUFFICES FOR "DATE DONE" IF SAME DAY AS RECEIVED.
_____ Additional information on reverse (head to tail) or attached.

Figure 2.2: Visit Date Establishment Sheet

SCHEDULED EMPLOYMENT INTERVIEWS - SPRING SEMESTER 1987

INTERVIEW VISIT DATE ACCEPTABLE GRAD. DATES SEMESTER(S) VISITING KSU	COMPANY NAME (LOCATION OF COMPANY) LOCATION OF JOB OPENINGS	DEGREE REQUIRED LEVEL & CURRICULUM OR MAJOR
February 2 5/87 Spring Only	Cigna Corporation (Philadelphia, PA) Nationwide	B: ECON, ACCTG, FINAN, GBA, MKTG GPA 3.0
February 2 12/86, 5-8/87 Summer: FR, SOPH, JR Spring & Fall	US Army (Manhattan, KS) Worldwide	B: ANY & ALL MAJORS
February 2 5/87, 8/87 First Visit	Westvaco (Covington, VA) Covington, VA; Laurel, MD; Charleston, SC	M OR D: ME, CHE
February 3 12/86, 5-8/87 Spring Only	ADM Milling Company (Shawnee Mission, KS) Various	B: MSM
February 3 Summer: All Classes Spring Only	Cheley Colorado Camps (Denver, CO) Estes Park, CO	Summer: ANY & ALL MAJORS - OPEN SIGNUP
February 3 12/86, 5/87 Spring Only	Civilian Personnel Office (Ft. Riley, KS) Ft. Riley, KS	B: INSYS, JMC, ALL BUS EXCEPT ACCTG B OR M: CMPSC, ECON, ENGL, POLSC, PSYCH HIST
February 3 5/87, 8/87 Spring & Fall	FDIC (Overland Park, KS) KS, MO, IA, NE, MN, ND, SD	B: FINAN, GBA B OR M: AGEC, ACCTG (Minimum of 6 hrs ACCTG)
February 3 5/87 Spring Only	Parker Hannifin (Cleveland, OH) Nationwide	B: EE, ME
February 3 Summer: JR, SR Spring & Fall	The Procter & Gamble Co. (Cincinnati, OH) Kansas City Coffee Plant	B OR M: CHE, EE, ME
February 3 5/87, 8/87 Spring & Fall	Red Lobster Inns of America (Rolling Meadows, IL) Midwest	B: MANGT

Figure 2.3: Master Visit List

VISIT OR SCHEDULE CANCELLATION SHEET

Organization _____ Date Canc. Rec'd. _____

_____ Phone Call Taken By _____

Address _____ Letter Dated____Rec'd By___

_____ Other _____

Cancellor's Name _____ Tel. (___) _____ ____
AC

Title _____

_____ Complete cancellation of visit, no transfer or replacement dates.

_____ Complete cancellation of visit, but involving replacement date(s) _____

No. Schedules ____ _____ _____ M TU W TH F 86 87 88 89 90
Month 1st Day

No. Schedules ____ _____ _____ M TU W TH F 86 87 88 89 90
Month 2nd Day

_____ Partial Cancellation

_____ _____ M TU W TH F 86 87 88 89 90
Month 1st Day

_____ _____ M TU W TH F 86 87 88 89 90
Month 2nd Day

_____ Schedules reduced in number from _____ to _____

_____ Schedules reduced in length as described:

_____

_____ Other (Explain) _____

Remarks _____

* *

Reason for cancellation: _____ Light Schedule _____; _____ Reduced Manpower
No.

Need; _____ Recruiter Convenience; Other _____

SEE REVERSE SIDE FOR ACTION REQUIRED

One of these sheets is to be prepared for each company cancellation. Cancellation sheets are to be attached to the Visit Date Establishment sheets which they cancel; on top for complete cancellation, and underneath for partial cancellations.

Figure 2.4: Visit or Schedule Update

CAREER PLANNING AND PLACEMENT CENTER
INTERVIEW REQUEST FORMS POSTED ON MONDAY, MARCH 23, 1987
FOR MONDAY, APRIL 13, 1987
POSTED AT 8 A.M.

EMPLOYERS REINSURANCE CORPORATION — B: ANY & ALL MAJORS
1 SCHEDULE — 6 HOURS COBOL
12/86, 5/87 — GPA 3.0

OPEN SIGNUP FOR MONDAY AND TUESDAY, MARCH 30-31, 1987

PRODUCTION ADVISORY SERVICE, INC. — ALL ARICULTURE
1 SCHEDULE EACH DAY
SUMMER - ALL CLASSES

Figure 2.5: Daily Update To Master Visit List

DATA SHEET

KANSAS STATE UNIVERSITY
Career Planning and Placement Center
Manhattan, Kansas 66506

Please Type

It is our policy to deal only with equal opportunity employers and those complying with PL 93-380 re confidentiality of records

Name _____ Date _____
Last First Middle

Other name(s) used now or previously _____
Present Address _____ Phone (AC) _____ _____
Permenent Address _____ Phone (AC) _____ _____

PERSONAL DATA: (optional)

Birthdate _____ Height _____ Weight _____ Sex _____ Citizenship _____ (Country)

Relevant health considerations _____ Type of Visa _____

JOB INTERESTS: _____
Available for employment? _____ (Date) Location Preferences: _____

EDUCATIONAL DATA:

Colleges Attended and Location	Dates of Attendance	Degree	Major Field*	G.P.A. in Major	Overall G.P.A.	Graduation Date

Grade point system at Kansas State. A = 4, B = 3, C = 2

*Other areas of concentration and hours _____

Percent of college expenses: Working ____ Scholarships ____ Grants ____ Other _____ (Specify)

Honors, Scholarships, Professionel, and Honor Societies _____

Community/Extra-curricular activities _____

Special skills, interests, and hobbies _____

OCCUPATIONAL EXPERIENCE: (Include full and pertinent part-time work)

Employer	Address	Duties	Dates Employed

☐ "X" in block indicates continuation under related headings on reverse or attachment.

REFERENCES: (Names, titles, addresses, phone numbers of faculty members, former employers, etc.)

1. _____
2. _____
3. _____

ADDITIONAL INFORMATION _____

I affirm that the above information is correct to the best of my knowledge and, subject to the provisions of PL 93-380, hereby authorize the Career Planning and Placement Center to release this data sheet and related information, including references, to prospective employers and/or to institutions of higher learning. I have read and understand the statement of Principles and Practices of College Placement promulgated by the College Placement Council, Inc. and promise to comply therewith.

_____ _____
Legal Signature Social Security No. (typed)

Figure 2.6: Student Data Sheet

CAREER PLANNING AND PLACEMENT CENTER
Commercial & Service Placement
KANSAS STATE UNIVERSITY
Manhattan, KS 66506
Page No. ____

INTERVIEW REQUEST FORM

INTERESTED CANDIDATES MUST LEAVE DATA SHEET FOR FORWARDING TO THE ORGANIZATION

ORGANIZATION _____

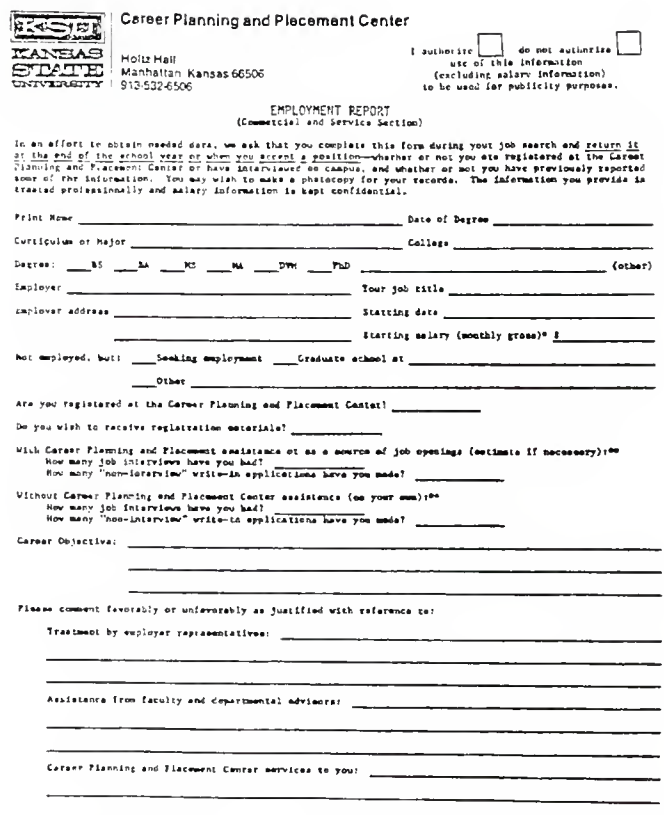
DATE OF INTERVIEW _____

MAJOR(S) & DEGREE(S) REQUESTED (Persons with closely RELATED degree, coursework, or work experience may also request an interview) _____

GRADUATION DATE(S) _____

NAME	PHONE	NAME	PHONE
1.		19.	
2.		20.	
3.		21.	
4.		22.	
5.		23.	
6.		24.	
7.		25.	
8.		26.	
9.		27.	
10.		28.	
11.		29.	
12.		30.	
13.		31.	
14.		32.	
15.		33.	
16.		34.	
17.		35.	
18.		36.	

Figure 2.7: Interview Request Form



KANSAS STATE UNIVERSITY

Career Planning and Placement Center

Holtz Hall
Manhattan, Kansas 66506
913-532-6506

I authorize ☐ do not authorize ☐ use of this information (excluding salary information) to be used for publicity purposes.

EMPLOYMENT REPORT
(Commercial and Service Section)

In an effort to obtain needed data, we ask that you complete this form during your job search and return it at the end of the school year or when you accept a position--whether or not you are registered at the Career Planning and Placement Center or have interviewed on campus, and whether or not you have previously reported some of the information. You may wish to make a photocopy for your records. The information you provide is treated professionally and salary information is kept confidential.

Print Name _____ Date of Degree _____

Curriculum or Major _____ College _____

Degree: ___BS ___AA ___MS ___MA ___DVM ___PhD _____ (other)

Employer _____ Your job title _____

Employer address _____ Starting date _____

_____ Starting salary (monthly gross)* $ _____

Not employed, but: ___Seeking employment ___Graduate school at _____

___Other _____

Are you registered at the Career Planning and Placement Center? _____

Do you wish to receive registration materials? _____

With Career Planning and Placement assistance or as a source of job openings (estimate if necessary):**
How many job interviews have you had? _____
How many "non-interview" write-in applications have you made? _____

Without Career Planning and Placement Center assistance (on your own):**
How many job interviews have you had?
How many "non-interview" write-in applications have you made? _____

Career Objective: _____

Please comment favorably or unfavorably as justified with reference to:

Treatment by employer representatives: _____

Assistance from faculty and departmental advisors: _____

Career Planning and Placement Center services to you: _____

(over)

* Include amounts estimated for services such as lodging and food related to Ag positions, etc.
**Please complete the chart on the back of this page.

Figure 2.8: Employment Report

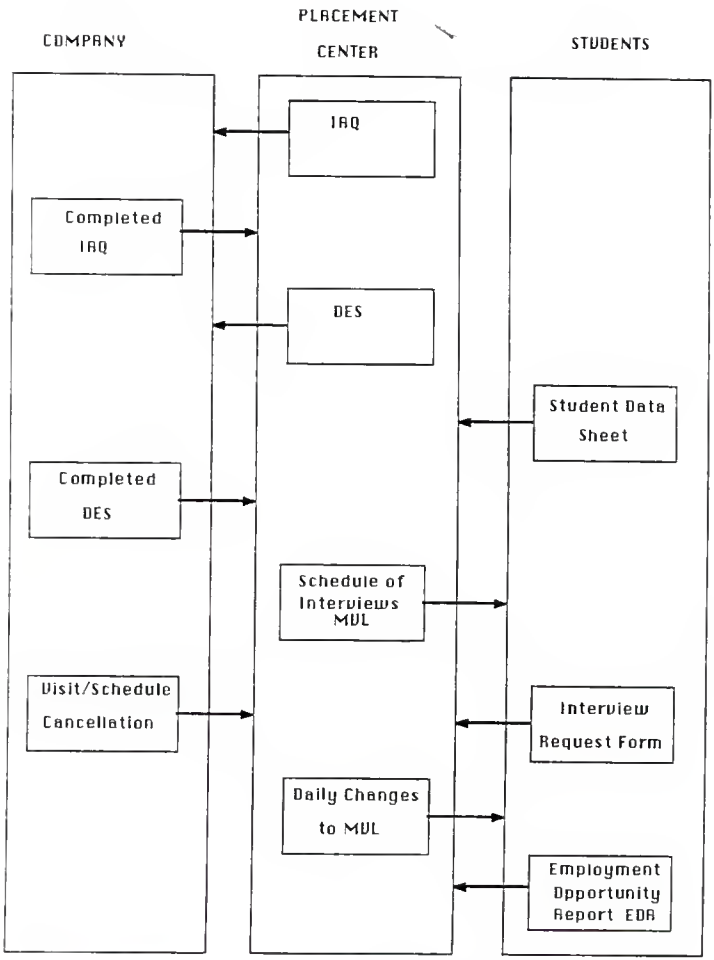


Figure 2.9: Form Analysis

2.2 E-R Diagrams

The interested reader can refer to Cathy Puzzuoli's work [PUZZ87] for a discussion of the data design, the E-R, and the schema. In this section, we provide the E-R and the relational schema for clarity. The Entity-Relationship Model, (E-R Model), is a tool used to determine data structures [TEO82]. Figure 2.10 gives the attribute list and student information E-R model which represents the entity student. Figure 2.11 gives the attribute list and the employment report model. Figure 2.12 is the attribute list and the company information model which represents the company entity. Figure 2.13 is the attribute list and the perspective employer model. Figure 2.14 is the attribute list and employment opportunity report model which represents the company entity. Figure 2.15 is the attribute list and the student address model.

Student Name	Curriculum
Student Number	Degree
Student Present Address and Phone	College
Student Permanent Address and Phone	Curriculum GPA
When Available for Employment	Overall GPA
Where Available for Employment	Graduation Date
Sex	

Attribute List for Student Information

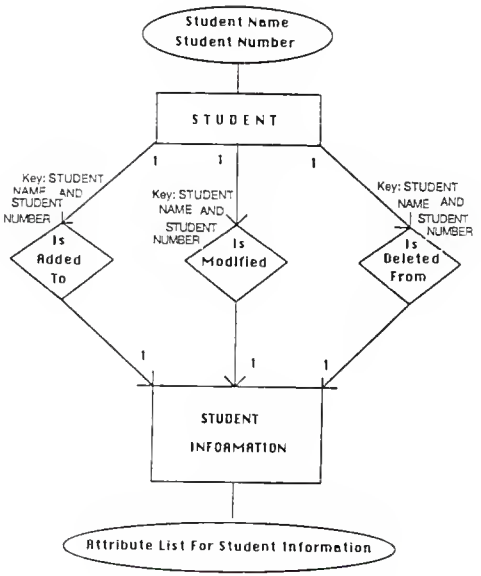


Figure 2.10: Student Information E-R

Date Received	Company Address
Student Name	Starting Salary
New Address of Student	Job Description
Curriculum	Hired
Degree	Seeking Employment
Graduation Date	Continuing Education
Company Name	

Attribute List Employment Information

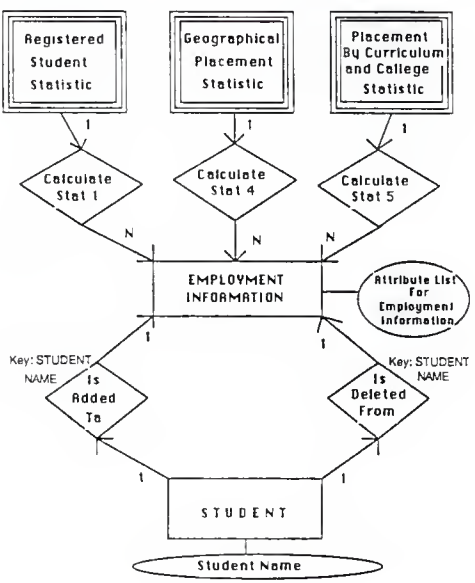


Figure 2.11: Employment Information E-R

Company Name
Company Address and Phone
Contact Name
Contact Address and Phone
Date Received DES
Number of Schedules
Number of Recruiters per Schedule
Interview by GPA
Interview by Curriculum
Which Curriculums Are Accepted
International Candidates Cannot be Interviewed
International Candidates Interviewed without Restriction
Interview Students Holding Permanent Resident Status
Interview U.S. Citizens Only

Attribute List for Company Information

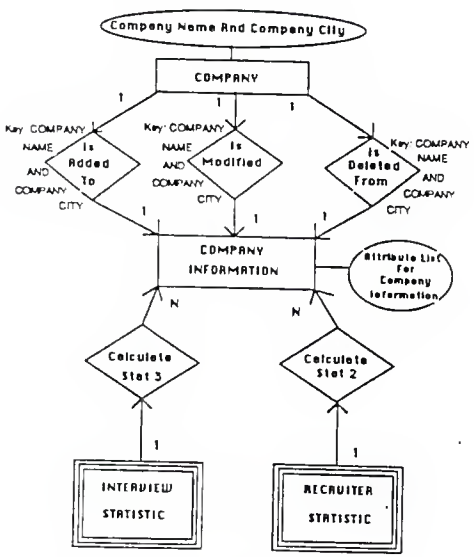


Figure 2.12: Company Information E-R

Company Name Company Address and Phone Contact Name Title Contact Address and Phone

Attribute List for Perspective Employer

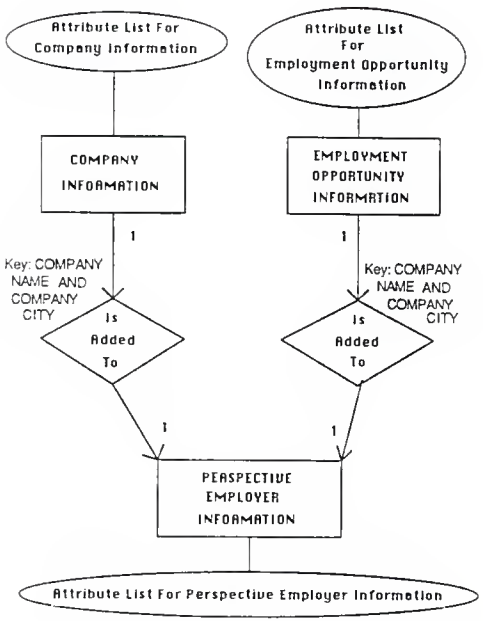


Figure 2.13: Perspective Employer E-R

Date Received	Job Location
Application Deadline	Salary
Job Title	Company Name
Degree	Company Address and Phone
Curriculum	Contact Name
Special Requirements	Title
Job Description	Contact Address and Phone

Attribute List for Employment Opportunity Report

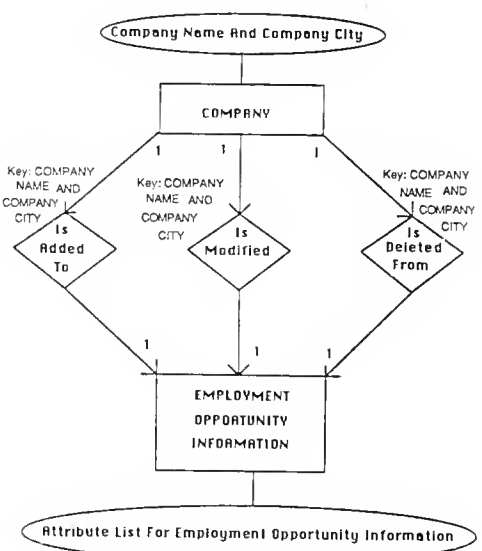


Figure 2.14: Employment Opportunity Report E-R

Student Name
Student Number
Present Address and Phone
Permanent Address and Phone

Attribute List for Student Address

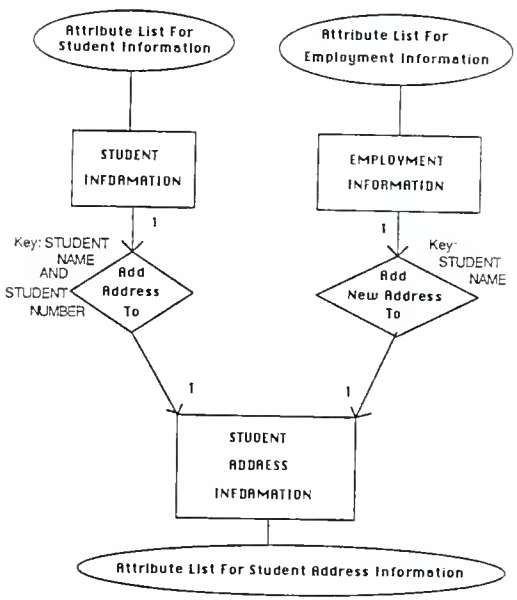


Figure 2.15: Student Address E-R

2.3 Relational Schema

The formal relational schema is shown below. A relational database is a set of all the relations in the schemata [UNG87]. Each relation represents an entity. The relational data base for this project is shown below. Each of the six relations shown below, represents an entity.

Below is a list of relations in the database design.

Relation R1 represents the company entity. This relation was developed from the Company Information E-R model, figure 2.12.

R1 (Company Name, Company City, Company Address, Company Phone, Contact Name, Contact Address, Contact Phone, Interview Schedules, Number of Recruiters, Interviewing by Dates of Graduation, Special Restrictions)
KEY: (Company Name and Company City)

Relation 2 is represents the Student Information entity in the E-R Model in figure 2.10.

R2 (Student Name, Student Number, Present Address, Present Phone, Permanent Address, Permanent Phone, Available for Employment, Overall GPA, Sex, Curriculum, Degree, Graduation Date, Number of Times Not Shown Up for Interviews)
KEY: (Student Name and Student Number)

Relation 3 represent the student's employment entity and was generated from

the Employment Information E-R Model in figure 2.11 using Bernstien's synthesis algorithm.

R3 (Student Name, Company Name, Company Address, Starting Salary, New Student Address, Job Description, Location of Job, Curriculum, Degree, Graduation Date, Seeking Employment, Hired, Continuing Education)
KEY: (Student Name)

The Perspective Employer E-R model, figure 2.13, is used to derive Relation 4 which represents the employer entity.

R4 (Company Name, Company City, Company Address, Company Phone, Contact Name, Contact Title, Contact Address, Contact Phone)
KEY: (Company Name and Company City)

Relation 5 represent the employment opportunity entity and matches the Employment Opportunity Report E-R model in figure 2.14.

R5 (Company Name, Company City, Date Received EOR, Application Deadline, Job Title, Degree, Curriculum, Special Requirements, Job Description, Job Location, Salary)
KEY: (Company Name, Company City)

Relation 6 represents from the Student Address E-R model in figure 2.15 and was derived synthetically from Bernstien's algorithm.

R6 (Student Name, Student Number, Present Address, Present Phone, Permanent

Address, Permanent Phone)

KEY: (Student Name and Student Number)

Chapter 3 describes the use of these six relations in the implementation of the Center's system.

Chapter 3: Implementation

Since the dBase product line was the chosen software package for the implementation of the system, a comparison of the versions is discussed in section 3.1. and the application programs are described in 3.2. Section 3.3 consist of a users guide to the program.

3.1 Comparison of dBase Line

dBase II has several advantages over dBase III. One advantage is that dBase II takes less space than dBase III, thereby making it more adaptable for programs in floppy disk systems. The one character commands in dBase II have been expanded by dBase III to multiple character functions. While this improves legibility for a novice user, it makes the program bulky and harder to write. It should be noted that dBase III has no significant execution speed advantage over dBase II for most applications.

dBase III has many advantages over dBase II. dBase III allows the user to keep up to 10 files open at a time. This relives the programmer from having to open and close files constantly. dBase III lets the user up to 256 memory variables, thereby alleviating the user from setting up a memory variable file and permitting more activity in memory and less in database files. Also, dBase III permits up to 128 fields in a record. Files can be linked together for reporting and other purposes with a new command called, SET RELATION TO. Numerical accuracy has been extended from 10 digits to 15.9 and three additional math functions, EXP, LOG, and SQRT have been added. Two new variable types, MEMO and DATE permit more flexibility in defining fields. dBase III is written in the C language, thereby making it more portable and reliable.

3.2 Application Programs

There are 17 programs which were fully implemented for the Center. These programs are grouped into the following classes: student application programs, company application programs, statistical programs, and menus.

3.2.1 Student Application Programs

There are four programs written to allow the user to modify the student data in the student database, deletest.prg which deletes an entire student's record from the database. Addstud.prg adds a student to the database. Chgstud.prg lets the user update fields in the student's record. Emreport.prg allows the user to view the names of all students with an existing employment report on file at the Center or to enter an employment report for a student.

3.2.2 Company Application Programs

Three programs modify the company data. They are addcomp.prg which adds a company to the database, delcomp.prg which deletes a company from the database, and chgcomp.prg which lets the user update fields in a company record.

3.2.3 Statistical Programs

Five statistical programs were created to meet the specific needs of the Center. Stat1.prg computes the number of registered students, and then gives the percent of students who are registered and graduating as opposed to those who are graduating and not registered at the Center. It also informs the Center on how many registered students have found employment, how many are still seeking employment, and the number those furthering their education.

Stat2.prg calculates the total number of schedules made by companies, and

the number of recruiters sent per semester. It lists the number of schedules made, the number of schedules cancelled, and the percent of schedules kept by all companies visiting the University.

Stat4.prg is responsible for calculating the number of companies recruiting by curriculum, grade point average, or both.

Stat5.prg calculates the yearly placement of students by both curriculum and college.

Stat7.prg computes the statistics for geographic placement of students by states in the U.S., for each college.

A discussion of each is given in section 3.4.3.

3.2.4 The Menu Structure

Five menu programs were written to allow the user to interact with the system. They are the Main Menu, the Student Menu, the Company Menu and the Print Option Menu, and are discussed below. Submenus were included in for clarity in connection with the Student and Company Menus (Figure 3.1).

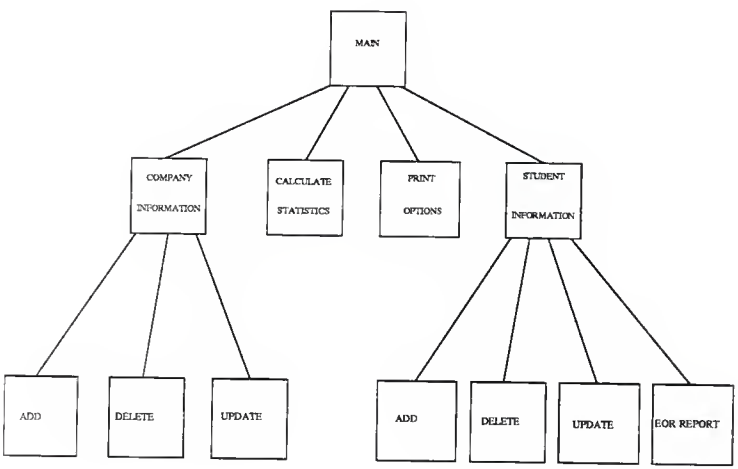


Figure 3.1: Menu Hierarchical Chart

3.2.4.1 Main Menu

The mainm.prg which gives the user the ability to chose modification of students, or company records, or perform desired statistics, or the print menu which allows for various documents to be printed (see figure 3.2).

3.2.4.1 Company Menu

This menu allows for the addition, deletion or updating of a company record (see figure 3.3).

3.2.4.2 Student Menu

This menu lets the user chose whether to add, delete, update or enter an employment report for a student (see figure 3.7).

3.2.4.3 Statistic Menu

This menu allows the user to chose one of six specified statistical calculations (see figure 3.16). These are the calculation of the total number and percentage of: graduating students, registered students, the semester recruiting statistics and cancellation rate for all companies, the calculation of companies recruiting by curriculum and/or grade point average, the yearly placement of students by curriculum, the yearly placement of students by curriculum and college, and the statistics for geographic placement by college. All these programs will be discussed below in the Users Manual.

3.3 Users Manual

The Center has dBase III on its Zenith 150 and therefore after entering the program needs to enter <do mainm> when encountering the dBase prompt. The Main Menu will come to the screen as seen in figure 3.2. The four choices given by this menu are, Select Company, Select Student, Calculate Statistics, or the Print or View Option. These options are discussed below. There are some uniform responses to incorrect menu selections. For any invalid menu selection, the message "Invalid Number. Hit Any Key To Continue" is given and returns the user to the menu to make another selection.

****

MAIN MENU

1. Select Company
2. Select Student
3. Calculate Statistics
4. Exit Program

****

Enter The Number Of Your Selection: ____

Figure 3.2: Main Menu for Placement Program

3.3.1 Select Company

If entering company information, this is the selection that should be made. There are three options that can be taken, adding a company, delete a

company, or update a company as seen in figure 3.3. Each of these options are discussed below.

****

COMPANY INFORMATION MENU

1. Add A New Company
2. Update Company Information
3. Delete A Company
4. Return To Main Menu

****

Enter The Number Of Your Selection: ____

Figure 3.3 Company Information Menu

3.3.1.1 Add A Company Record

In chosing this menu option the user has the choice of adding a company, viewing all existing companies in the data base, or quitting (returning to select company menu. If the option to add is chosen, the user is prompted to enter the company name and the city where the company resides; these two fields are the keys to this relation.

If the company is already in the data base, the user is so informed and upon striking any key, is returned to the submenu.

If the company is not in the data base, the company name and city is displayed and the user is asked to confirm that the company is to be added (figure 3.4). This confirmation is to check for any errors in the name and city that may have occurred. A Company input template is then displayed (figure 3.5) and the user can then enter the desired data into the data fields. When finished the user is asked whether the record is to be saved to the database. If so, the company is added to the database and the user is asked whether or not they wish to save the modified database. When the user confirms that they wish to save the modified database, it is displayed on the screen that is has been saved.

No Record On File For:

COMPANY: **XEROX CORPORATION**
CITY: **KANSAS CITY**

Would You Like To ADD This Company?

1. To ADD
2. To ENTER Another Company
3. To QUIT

Enter Choice: ____

Figure 3.4: Adding A New Company

COMPANY INFORMATION

NAME: XEROX CORPORATION SCHEDULES & NO. OF RECRUITERS
ADDRESS: AAAAAAAAAAAAAA
AAAAAAAAAAAAAA 1. NN NN 5. NN NN
KANSAS CITY 2. NN NN 6. NN NN
3. NN NN 7. NN NN
PHONE: (NNN)NNN-NNNNN 4. NN NN TOTAL SCH: NN
CANCELLED: NN
DES NAME: CCCCCCCCCCCC RECEIVED: MM/DD/YY
TAKEN BY: CC
PHONE: (NNN)NNN-NNNNN
INDUSTRY: AAAAAAAAAAAA WK LOCATION:

INTERVIEWING INFORMATION

CURRICULUM(S): NNNN NNNN SUMMER EMPLOYMENT: (Y/N)
NNNN NNNN FR B SO B JU B
SR B GR B NON B
GRADUATES: DEC NN MAY NN
AUG NN DEC NN

INTERNL: No Restr. B US Only B Perm Resident B Cant Interview B

KEY	
A	Alphanumeric (0..9,a..Z)
B	Boolean (0,1,Y,N)
C	Characters (a..Z)
D	Days (01..31)
M	Months (01..12)
N	Numeric (0..9)
Y	Years (00..99)

Figure 3.5: Company Input Template

3.3.1.2 Delete a Company Record

This option works in much the same manner as adding a company. Upon entry, the user may chose one of three options, to delete, to view all company names in the database, or to quit and return to the

Select Company Menu.

If deleting, the user is asked for the company name and the city in which the the company resides. If the company is not found in the database, the user is given two options: to enter another company name and city or to quit and return to the Select Company Menu.

If the company is found, it is displayed and the user is given the option of deleting or returning to enter another company name and city, or quitting this option (Figure 3.6). If the user decides to delete, it must be confirmed, and is notified on the screen that the company has been removed from the database. If the decision to delete is not confirmed, notification that the company has not be removed is displayed.

This Company Is In The Database As:

COMPANY NAME: **XEROX CORPORATION**
ADDRESS: 1212 HANEY STREET
P.O. BOX 2323
KANSAS CITY, MO

1. To DELETE
2. To CONTINUE
3. To QUIT

Enter Choice: ____

Figure 3.6: Display of Company to be Deleted

3.3.1.3 Update A Company Record

Upon entering the user is given three options: to change one or more fields in a company record, to view all company names and corresponding cities in the database or to return to the Select Company Menu.

If chosing to update a company, the user is prompted to enter the company name, and then the city where the company resides. If the company is not found, the user has the following options: to enter another company name and city, or to return to the Select Company Menu.

If the company is found the two options are to change the company information to enter another company name and city or the return to the Select Company Menu. If the option to change is taken, the company's record is displayed with the cursor at the first changeable field. The user may change the field or hit return to go to the next field until all fields have been reviewed or changed. Upon hitting return after the last field the user is taken back to the submenu.

3.3.2 Select Student

This option allows the user to add new students to the student data base. Figure 3.7 shows the Student Information Menu which gives the four options: adding a student, updating a student, deleting a student or creating/viewing an employment report for a student. All options are discussed in detail below.

****

STUDENT INFORMATION MENU

1. Add A Student
2. Change Student Information
3. Delete A Student
4. Employment Report
5. Return To Main Menu

****

Enter The Number Of Your Selection: ____

Figure 3.7 Student Information Menu

3.3.2.1 Add A Student Record

To add a student to this data base, the student must have a unique student number. Upon entering this selection the user is given three options: to add a student, to view all student names in the data base, or to quit and return to the Main Menu (figure 3.8).

ADD A STUDENT

1. To ADD A Student
2. To VIEW ALL Students In Database
3. To QUIT

Enter Choice: ____

Figure 3.8: Add A Student Menu

If chosing to add a student, the user is prompted for a student number. If this number is already in the data base, the number and the student name associated with it is displayed, and the user is prompted to hit any key to return to the submenu (figure 3.9).

This Student Is Already In The DataBase As:

Bridget T. Vanek

Hit Any Key To Continue

Figure 3.9: Message If Student Already In Database

If the number is not found in the data base, the user has three options. to add the student that corresponds with the number, to

enter another number (this number was not entered correctly), or the quit and return to the Main Menu (figure 3.10).

No Record On File For This Student.

Would You Like To ADD This Student?

1. To ADD This Student.
2. To ENTER Another Student Number.
3. To QUIT.

Enter Choice: ____

Figure 3.10: Message If Student Not In Database

If the decision to add is made, the user is prompted to enter the student's first name, and last name (figure 3.11). A input template is then placed on the screen for the users convenience (figure 3.12). The student's name, and number is displayed in the heading. Pertinent information can now be entered. The cursor is moved field to field by striking the <return> key. The user can use the <backspace> key to return to previously entered fields. When finished entering, the user is prompted for confirmation in order for the student to be saved in the data base.

Enter Student First Name: XXXXXXXXXXXX

Enter Student Middle Initial: X

Enter Student Last Name: XXXXXXXXXXXX

Figure 3.11: Entering New Students Name To Database

STUDENT INFORMATION

FOR STUDENT: Bridget T. Vanek
NUMBER: 333333333

Local Street Address	AAAAAAAAAAAAAAAA	State CC	Zip Code NNNNN
City	CCCCCCCCCCC		
Local Area Code/Phone Number	(NNN)-NNN-NNNN		
Permanent Street Address	AAAAAAAAAAAAAAAA	State CC	Zip Code NNNNN
City	CCCCCCCCCCC		
Area Code/Phone Number	(NNN)-NNN-NNNN		
Sex C			
Date Available For Employment	MM/DD/YY		
Overall GPA	N.NNN	Curriculum GPA	N.NNN
College:	NNNN	Curriculum Code: NNNN	Degree: NNN
Graduation Date (mm/yy)	MM/YY		
Is This Student Registered? (Y/N)	B	No. of No Shows: NN	

KEY	
A	Alphanumeric (0..9,a..Z)
B	Boolean (0,1,Y,N)
C	Characters (a..Z)
D	Days (01..31)
M	Months (01..12)
N	Numeric (0..9)
Y	Years (00..99)

Figure 3.12: Student Input Template

3.3.2.2 Delete A Student Record

Upon entering this selection, the user has four options: to delete by the student number, to delete by the student's name, to view all the students in the data base, or to quit and return to the Main Menu.

If the user choses to delete by student name or number and the

student is not in the data base; the user is informed there is no record for this student before control is returned to the submenu.

If the student is found, the student name and number are displayed and the user has three options: to continue with the deletion process, the enter another student name or number, or to quit and return to the submenu.

If continuation of the deletion process is selected, a confirmation is requested and the deletion takes place. This is so noted to the user.

3.3.2.3 Update A Student Record

As in the deletion of a student, the user is given the option of looking up the student's record by the student's name or the student's number. In either case if a corresponding record is not found, the user is notified and returned to the submenu.

If a record is found, the student's name and number are displayed and the user has three options available: to go into the record and make changes, to enter another student name or number or to return to the Select Student Menu.

If the option to change a student's record is taken, the record is displayed and the fields can be changed by moving the cursor around to the desired fields and typing in the changes. Once the changes are made a message is displayed that the changes have been saved to the

database (figure 3.13).

These Changes Have Been Saved. Hit Any Key To Continue.

Figure 3.13: Message After Making Changes To A Student Record.

3.3.2.4 Employment Report

An employment report is out when a student gets a job or alternatively at the end of the semester. To file an employment report the student must be registered at the Center. The user upon making this selection, is has three options: to add an employment report for a student, to view an employment report for a student who has already filed one, or to return to the student menu. see figure 3.14. If the user chooses to view an employment report, the student's first and last name is asked for, see figure 3.15.

Employment Report

1. To ADD An Employment Report
2. To VIEW An Employment Report
3. To QUIT

Enter Choice: ____

Figure 3.14: Menu For Employment Report

View An Employment Report

Enter Students First Name: XXXXXXXXXXXX
Last Name: XXXXXXXXXXXX

Figure 3.15: Request To View Employment Report Input

If the student does not have an employment report it is so stated and the user can return to the employment menu by striking any key. If the student has an employment report it is displayed on the screen. The user is allowed to view but cannot change any fields. The user can continue by striking any key.

If the user choses to add an employment report, the prompt for the students first and last name is encountered. If the student is not registered a message is stating that the student first needs to have a data sheet on file at the center before filing an employment report, see figure 3.16. If the student is registered an input template is placed on the screen, see figure 3.17. The user may enter the appropriate data. When finished, the user is asked whether or not they wish to save the report.

Enter Student First Name: XXXXXXXXXXXXX
Last Name: XXXXXXXXXXXX

Need to Input A Student Data Sheet First.

Hit Any Key To Continue.

Figure 3.16: Employment Report For Student Not Registered

STUDENT EMPLOYMENT REPORT

STUDENT NAME:	Bridget Vanek		DATE: MM/DD/YY
CURRICULUM:	NNNNN	DEGREE CODE: NNN	DATE DEGREE: MM/YY
NEW ADDRESS:		EMPLOYED BY:	AAAAAAAAAA
		STREET:	AAAAAAAAAA
STREET:	AAAAAAAAA	STREET:	AAAAAAAAAA
CITY:	CCCCCCCCC	CITY:	CCCCCCCCCC
STATE/ZIP:	CC NNNNN	STATE/ZIP:	CC NNNNN

FOUND A JOB? (Y/N) B

If You Found A Job: SALARY/Month: $NNNNN.NN
JOB DESCRIPTION:
LOCALE (State):

IF YOU DO NOT HAVE A JOB:
ARE YOU SEEKING EMPLOYMENT? B
ARE YOU CONTINUING YOUR EDUCATION? B

Hit Any Key To Continue.

KEY	
A	Alphanumeric (0..9,a..Z)
B	Boolean (0,1,Y,N)
C	Characters (a..Z)
D	Days (01..31)
M	Months (01..12)
N	Numeric (0..9)
Y	Years (00..99)

Figure 3.17: Student Employment Report.

If the student already has an employment report on file, the user is told that an employment report cannot be entered as there is already one on file for that student.

3.3.3 Calculate Statistics

The calculation of specific statistics hy the system is discussed in this section. The user has five statistical functions to chose from: the yearly total and percentage of registered students hy curriculum, semester recruiting statistics, the number of companies that recruit hy curriculum, gpa or hoth, the yearly employment statistics by college and curriculum, and the geographic placement by curriculum. Figure 3.18 shows the statistics menu and options.

****

CALCULATE STATISTICS MENU

1. Yearly Total & Percentage of Students By Curriculum
2. Semester Recruiting Statistics & Cancellation Rate
3. Companies Recruiting By Curriculum And/Or Degree
4. Yearly Employment Statistics By College & Curriculum
5. Yearly Geographic Placement by Curriculum
6. Return To Main Menu

****

Enter The Number Of Your Selection: ____

Figure 3.18: Calculate Statistics Menu.

3.3.3.1 Yearly Total and Percentage of Students by Curriculum

When entering this program the user is prompted to enter a date of graduation. All students graduating by this date will ge used in the calculations. Next the user is prompted for the total number of graduates. This number is calculated by the University Registrar office. It includes all students, not those just registered by at the Center. The output includes: the number of registered students at the Center, the number of these students that found employment, the number still seeking

employment and those students who are furthering there education, see figure 3.19.

Calculation of Total and Percentage of Graduating Students

Enter Date (mm/dd/yy): 05/25/87
Enter Total Number of Graduates: 1000

Total Number of Graduating Students: 100
Graduating Student Registered: 20%
Hired:
Seeking Employment: 10%
Graduate School: 2%

Calculations Have Been Completed.
Hit Any Key To Continue.

Figure 3.19: Calculation of Graduating Students

3.3.3.2 Semester Recruiting Statistics and Cancellation Rate

This program calculates the total number of schedules made by all companies. It also computes the number of recruiters sent and the number of canceled schedules. With this information it calculates the percent of companies that did show up as planned, see figures 3.20 and 3.21.

Calculation of Recruiting Statistics and Cancellation Rate For All Companies

Calculations Have Been Completed.

Hit Any Key To Continue.

Figure 3.20: Semester Recruiting Screen

Schedules Made	Schedules Cancelled	Number of Recruiters	% Schedules Kept
200	20	40	90%

Hit Any Key To Continue.

Figure 3.21: Semester Recruiting Output

3.3.3.3 Companies Recruiting by Curriculum and/or GPA

This calculates the total number of companies who recruit by curriculum, grade point average or both. After chosing this option, a message lets the user know that calculations are underway, see figure 3.22.

Calculation of Companies Recruiting By
Curriculum and/or GPA

Calculations Have Been Completed.

Hit Any Key To Continue.

Figure 3.22: Companies Recruiting By Curriculum and/or GPA Screen.

After striking any key, figure 3.23 appears on the screen giving the number of companies recruiting by the aforementioned.

Company Recruiting Information

150 Companies Recruiting by Curriculum
100 Companies Recruiting by GPA
200 Companies Recruiting by Both

Hit Any Key To Continue.

Figure 3.23: Company Recruiting Information Output

3.3.3.4 Yearly Employment Statistics By College & Curriculum

This selection calculates the number of students employed by college and curriculum, see figure 3.24. The user can view the statistics by either college or curriculum, see figure 3.25.

Calculation of Placement Information
By Curriculum

And Now Calculating Totals By College

Calculations Have Been Completed.

Hit Any Key To Continue

Figure 3.24: Company Recruiting Information Screen

View The Statistics?

1. To View Totals By Colleges
2. To View Total By Curriculum
3. To Quit

Enter Choice: ____

Figure 3.25: Submenu for Company Recruiting Information

If the user selects to view the totals by colleges, figure 3.26 appears.

******** PLACEMENT TOTALS BY COLLEGE ********

Colleges	College Totals
All Agriculture	20
All Arts & Sciences	125
All Business	75
All Business Except Accounting	70
All College Architecture & Design	12
All Education Majors	35
All Engineering	50
All Home Economics	15

Hit Any Key To Continue

Figure 3.26: Output of Placement Statistics by College.

Striking any key returns the user to the submenu in figure 3.25. The user may choose the option to view the totals by curriculum in which will result in each college being placed on the screen one at a time. Beneath each college will be a listing of those curriculums in the college in which students found employment. The number of students is also displayed. see figure 3.27.

********** PLACEMENT TOTALS BY CURRICULUM **********

College of Arts and Sciences

Curriculum Code	Name	Number Employed
26	Anthropology	2
30	Biology	5
33	Computer Science	25
39	Geology	9
43	Information Systems	12
56	Physics	4
63	Social Work	10

Hit Any Key To Continue To Next College

Figure 3.27: Output of Placement Statistics By Curriculum

3.3.3.5 Yearly Geographic Placement by Curriculum

This program computes the geographic placement of students within the United States. The user has the option of choosing which college it would like to calculate, see figure 3.28. This program was designed and implemented. Displaying the results to the screen was extremely messy. Therefore it was recommended that the program would need to print the results to a printer. For further discussion on this feature, see Chapter 4.

Computing Statistics For Geographic Placement By College

1. College of Agriculture
2. College of Architecture & Design
3. College of Arts & Sciences
4. College of Business Administration
5. College of Education
6. College of Engineering
7. College of Human Ecology

Enter Choice: ____

Figure 3.28: Menu for Geographic Placement by College

3.3.3.6 Statistic Specification

These specific statistics were implemented for the following reasons, they were specified by the Center, they are commonly used, and they require a lengthy command sequence. The Center specified what statistics they wanted produced and because of they were frequently used and lengthy we concurred. These statistics are calculated at the end of each semester and/or school year. Since the Center requires a historical record of these statistics, storing the results in a database makes this easy. The shortest statistic program written is thirty-four lines long. Implementing these statistics programs therefore eliminates the need for good background knowledge of the dBase III command language.

Chapter 4: Future Work

The size of the project was such that it would take at least three graduate students to bring the project to completion. Only two students were available, so the project was designed but not completely implemented. The various subprojects involved were prototyped and were accomplished in order of priority. Even though incomplete, the system is very functional. There are two phases that remain unfinished. These are an application program to produce an EOR Report, and the capability to get a professionally formatted listing of the results of all the various application programs. These deficiencies are discussed below.

4.1 EOR Report

An Employment Opportunity Report is generated when a prospective employer informs the Center of a job opportunity. This company is not going to come to the University to interview applicants, instead applicants are notified of the opening when it is posted in the Job Opportunity Bulletin. The applicants are given the name of the company, information about the job, and who to contact. This bulletin is published on the first and fifteenth of every month.

A entity was created for the EOR Report. The application programs to insert, delete and modify data for this entity were not implemented. These will be simple to do as there are only three fields to be manipulated. Data is inserted, saved, and then a formated printout is produced. There are no calculations involved.

4.2 Print Outs

The Center needs hard copy reports for some of its functions and would like the capability to produce such for all of its applications. Two critical reports are the Master Visit List, produced at the beginning of each semester, and the Job Opportunity Bulletin, printed every two weeks. The third document would include a list of companies that need information about the Center. These might be new companies, or companies needing current information. Figure 4.1 shows the print menu.

****

PRINT OPTION MENU

1. Scheduled Employment Interview (MVL)
2. Posting Schedule (Daily)
3. Companies That Need Literature
4. Return To Main Menu

****

Enter The Number Of Your Selection: ____

Figure 4.1: Print Option Menu

Chapter 5: Results & Extensions

The results of the implementation project and future extensions are given in the following sections.

5.1 Results

The results of this project include:

A. Design of a complete system for the business and industry placement of a major university's placement center.

B. Implement a major portion of the system:
   1. dBase files
   2. dBase programs
      a. Add, delete, and update student files
      b. Add, delete, and update company files
      c. Calculate statistics
   3. Menu Driven System

C. Installation of that portion of the system

D. Collection of information on the process of moving an artisan office to an electronic office

5.2 Complications Involving Artisan Office Conversion

Some complications arise when dealing with an artisan office. As mentioned in Section 1.1, individuals work independently of one another at their tasks in an artisan style of office. If two or more people do the same task they often have different ways of achieving the goal. This is the case at the Center. Therefore, during the design process more people had to be considered. There was, as a result of the artisan environment, no authority to make decisions when conflicting design criteria arose. There was no one person who knew exactly how all work was carried to completion.

Many times compromises had to be made between two or more parties and our job became one of facilitating that compromise through meetings, design suggestions, and prototype systems. The design process was also extended, as some issues were approached more than once after a group decision had been made. The maintenance of an artisan office was desired because of the advantages such an environment has, namely, employee motivation.

5.3 Extensions

The system itself can be extended to increase its capability for the business and industry placement. Some of these extensions are given below.

5.3.1 Mini Data Sheet

Companies would like to get a mini-data sheet on registered students who meet some basic criteria. This sheet might include, curriculum, graduation date and desired work location. Such a sheet could be easily be produced by implementing the appropriate application program.

5.3.2 Letter Generator

The Center would like to generate letters and labels to students and prospective employers. The letters might be requesting employment information from students or sending out information about the Center to companies.

5.3.3 Command Front End

Menus were designed as this artisan office had little experience with computers. The menus gave the Center both ease in use and a quick

start. After extensive use though, menus can become burdensome and slow. A command front end might be useful to speed up the program.

5.4 Educational Placement

The placement of preschool, primary, and secondary, and junior college teachers has not been developed. Because that process is so different we are recommend that business and industry be accomplished mutually. A large second project would be the design and implementation of this system.

Bibliography

[BEI86] Beisner, Karl, "Database Management Software: A Selection Guide," Wilson Library, Bulletin, Vol. 60, June 1986, pp. 17-20.

[BEI86] Beisner, Karl, "dBase III," Library Software Review, Vol. 4, No. 2, March-April 1985, pp. 82-89.

[BRU86] Bruce, Robert C. and Williams, Ric., "U-Place: One Placement Director's Experience With This New Computerized System," Journal of Career Planning and Employment, Spring, 1986, pp. 55-56.

[CHA80] Champine, George A., "Perspectives on Business Data Processing," Computer, Vol. 13, No. 2., 1980, pp. 84-99.

[COL85] College Placement Council, VitaQuik Candidate Information System User Manual, College Placement Council, Inc., Bethlehem, PA, 1985.

[COL84] Colorado State University, CIS-ISS Placement Center Package, Colorado State University, Fort Collins, CO, 1984.

[CPC84] CPC Foundation, Use of Computers in Career Planning and Placement, CPC Foundation, Bethlehem, PA, 1984.

[GEH82] Gehani, Narain H., "The Potential of Forms in Office Automation," IEEE Transactions on Communications, Vol. 30, No. 1, January, 1982, pp. 120-125.

[KRU83] Kruglinski, David, Data Base Mangement systems, Osborne/McGraw-Hill, Berkeley, CA., 1983.

[MAS85] Mason, Robert M., "Database Mangement Software," Library Journal, Vol. 110, Nov. 15, 1985, pp. 64-66.

[POO85] Poor, Alfred, "Microrim's R:base 5000: A Database That Delivers," PC Magazine, Vol. 4, Sept. 3, 1985, pp. 185-192.

[PUZZ87] Puzzuoli, Cathy D., "Placement Center: A Database Design For An Artisan Office," MS Report, Kansas State University, 1987.

[TEO82] Teorey, Toby J. and Fry, James P., Design of Database Structures, Prentice Hall, Inc., 1982.

[TOW85] Townsend, Carl, Mastering dBase III, Sybex, Inc., Berkely, CA., 1985.

[ULL82] Ullman, Jeffrey D., Principles of Database Systems, Computer Science Press, 1982.

Software Referenced

Ashton-Tate, dBase III, Culver City, CA, 1984.

College Placement Council, VitaQuik Candidate Information System, College Placement Council, Inc., Bethlehem, PA., 1985.

Colorado State University, CIS-ISS, Colorado State University, Fort Collins, CO, 1984.

Iowa State University, ISU Resume System, Placement Center, Iowa State University, Ames, Iowa, 1985.

Microrim, R:base 5000, Bellevue, Washington, 1985.

Spartin Systems, U-Place, 1984.

```
*********************************************************************
* This program, mainm.prg, displays the main menu and allows
* the selection to the company, student, statistics or print
* options.
*********************************************************************
* Database Used:  None
* Memory File:    msave
* Index File:     None
*********************************************************************

set confirm on
set safety off
clear all
set talk off
restore from msave

do while .T.
clear
?
?
? '      ******************************************************'
?
?
? '                    MAIN MENU'
?
?
? '                    1. Select Company'
?
? '                    2. Select Student'
?
? '                    3. Calculate Statistics'
?
? '                    4. Print or View Options'
?
? '                    5. Exit Program'
?
?
? '      ******************************************************'
?
@20,9 say 'Enter The Number Of Your Selection: '
mselect = ' '
@20,46 get mselect picture '9'
read
 do case
   case mselect = '1'
     do comenu
   case mselect = '2'
     do stumenu
```

```
    case mselect = '3'
      do calcmenu
    case mselect = '4'
      do prtmenu
    case mselect = '5'
      save to msave
      release all
      exit
    otherwise
      clear
      @9,20 say 'Not A Valid Selection.'
      @16,20 say 'Hit Any Key To Continue.'
      wait ' '
      loop
  endcase mselect
enddo
```

```
*********************************************************************
* This program, comenu.prg, puts up a menu for selections to
* manipulate the company database.
*********************************************************************
* Database Used:  None
* Memory File:    msave
* Index File:     None
*********************************************************************

set safety off
clear all
restory from msave
set talk off
clear

do while .T.
?
?
? '      ************************************************'
?
?
? '          COMPANY INFORMATION MENU'
?
?
? '                    1. Add A New Company'
?
? '                    2. Update Company Information'
?
? '                    3. Delete A Company'
?
*? '                    4. Add An EOR'
*?
? '                    4. Return To Main Menu'
?
?
? '      ************************************************'
?
@22,9 say 'Enter The Number Of Your Selection: '
ms = ' '
@22,46 get ms picture '9'
read

do case
   case ms = '1'
     do addcomp
   case ms = '2'
     do chgcomp
```

```
   case ms = '3'
     do delcomp
* not available
*    case ms = '4'
*      do addeor
   case ms = '4'
     return
   otherwise
     clear
     @9,20 say 'Not A Valid Selection. '
     @16,20 say 'Hit Any Key To Continue.'
     wait ' '
     loop
  endcase ms
  clear all
 enddo
return
set confirm on
set safety off
clear all
do while .T.
clear
set talk off
restore from msave
clear
?
?
? '      ************************************************'
?
?
? '        STUDENT INFORMATION MENU'
?
?
? '              1. Add A Student'
?
? '              2. Change Student Information'
?
? '              3. Delete A Student'
?
? '              4. Employment Report'
?
? '              5. Return To Main Menu'
?
?
? '      ************************************************'
?
@20,9 say 'Enter The Number Of Your Selection: '
msel = ' '
@20,46 get msel picture '9'
read
```

```
@20,46 get msel picture '9'
read
  do case
    case msel = '1'
      do addstud
    case msel = '2'
      do chgstud
    case msel = '3'
      do deletest
    case msel = '4'
      do emreport
    case msel = '5'
      return
    otherwise
      clear
      @9,20 say 'Not A Valid Selection.'
      @16,20 say 'Hit Any Key To Continue.'
      wait ' '
      loop
  endcase msel
  clear all
enddo
return
```

```
*********************************************************************
* This program, calcmenu.prg, puts up a menu for calculating
* statistics.
*********************************************************************
* Database Used:  None
* Memory File  :  msave
* Index File:     None
*********************************************************************

* set up
set safety off
clear
set talk off
restore from msave


do while .t.
clear
?
? '     ******************************************************'
?
? '       CALCULATE STATISTICS MENU'
?
?
? '       1. Yearly Total & Percentage of Students by Curriculum'
?
? '       2. Semester Recruiting Statistics & Cancellation Rate'
?
? '       3. Companies Recruiting by Curriculum and/or GPA'
?
? '       4. Yearly Employment Statistics by College and Curriculum'
?
? '       5. Yearly Geographic Placement by Curriculum'
?
? '       6. Return To Main Menu'
?
? '     ******************************************************'
?
@23,7 say 'Enter The Number Of Your Selection: '
mselect = ' '
@23,44 get mselect picture '9'
read
 do case
   case mselect = '1'
     do stat1
   case mselect = '2'
     do stat2
```

```
    case mselect = '3'
      do stat4
    case mselect = '4'
      do stat5
    case mselect = '5'
      do stat7
    case mselect = '6'
      return
    otherwise
      clear
      @9,20 say 'Not A Valid Selection.'
      @16,20 say 'Hit Any Key To Continue.'
      wait ' '
      loop
  endcase mselect
enddo
return
```

```
*********************************************************************
* This program, prtmenu.prg, displays the print options and
* allows the selection of one.
*********************************************************************
* Database Used:  None
* Memory File:    msave
* Index File:     None
*********************************************************************

clear
set talk off
restore from msave
clear
?
?
? '      ******************************************************'
?
?
? '              P R I N T  O P T I O N  M E N U'
?
?
? '            1. Sceduled Employment Interview (MVL)'
?
? '            2. Posting Schedule (Daily)'
?
? '            3. Companies That Need to Send Literature'
?
? '            4. Return To Main Menu'
?
?
?
? '      ******************************************************'
?
accept '        Enter The Number Of Your Selection: ' to mselect

do while .T.
 do case
   case mselect = '1'
    do inview
   case mselect = '2'
    do posting
   case mselect = '3'
    do liter
   case mselect = '4'
    return
```

```
   otherwise
     accept 'Invalid Selection, Please Try Again ' to mselect
     loop
  endcase mselect
enddo
return
```

```
*****************************************************************
* This program addcomp.prg. adds a company to the company
* database compinfo.
*****************************************************************
* Database Used:  compinfo
* Memory File :  maddcomp
* Index File  :   ndxcompinfo
*****************************************************************

* setup
set safety off
set confirm on
set talk off
set exact off
restore from maddcomp

* maintenance for index file
use compinfo
index on trim(coname) - trim(cocity) to ndxcompinfo
set index to ndxcompinfo
reindex

use compinfo index ndxcompinfo
count to mrec2
* if dbase empty
if recno() = 1
  insert blank
  insert blank before
endif
manswer = ' '
set device to screen
do while upper(manswer) = ' '
  use compinfo index ndxcompinfo
  reindex
  clear
  choice = ' '
  @3,25 say 'A D D  A  N E W  C O M P A N Y'
  @10,25 say '1. ADD A Company'
  @11,25 say '2. VIEW All Companies In File'
  @12,25 say '3. To QUIT'
  @14,25 say 'Enter Choice:'
  @14,42 get choice picture '9'
  read
```

```
do case
 case choice = '3'
   save to maddcomp
   return
 case choice = '2'
   clear
   set heading off
   use
   use compinfo
   ?' COMPANY NAME                CITY            STATE'
   ?
   display all coname, cocity, costate off
   ?
   ?'                Hit Any Key To Continue'
   wait ' '
   loop

 case choice = '1'
   clear
   skey = '                    '
   @9,17 say 'Enter The Company Name: '
   @10,17 get skey picture 'xxxxxxxxxxxxxxxxxxx'
   read
   skey = upper(skey)
   ckey = '                    '
   @12,17 say 'Enter The City: '
   @13,17 get ckey picture 'xxxxxxxxxxxxxxxxxxx'
   read
   ckey = upper(ckey)
   cokey = trim(skey) - trim(ckey)
   find &cokey
   * company not in dbase
   if eof()
    clear
    madd = ' '
    @5,20 say 'No Record On File For:'
    @7,20 say 'COMPANY:'
    @7,35 say skey
    @8,20 say 'CITY:   '
    @8,35 say ckey
```

```
@11,20 say 'Would You Like To ADD This Company?'
@13,20 say '1. To ADD '
@14,20 say '2. To ENTER Another Company'
@15,20 say '3. To QUIT'
@17,20 say 'Enter Choice: '
@17,36 get madd picture '9'
read
do case
 case madd = '1'
   use
   use compinfo
   numsched = 0
   numcancl = 0
   * open new format file
   set format to comp
   count to mreccount
   goto top
   insert
   close format
   count to mrec2
   clear
   replace no_sched with 0
   replace cancelcnt with 0
   if mrec2 <> mreccount
    myn = ' '
    @9,20 say 'Do you want to SAVE this (Y/N)? '
    @9,55 get myn picture 'a'
    read
    if myn = 'Y' .or. myn = 'y'
    *  adding record to company database
      replace record 2 coname with upper(skey)
      replace record 2 cocity with upper(ckey)
      number = 0
      save to maddcomp
      do count_sched
      restore from maddcomp
      replace no_sched with number
      reindex
      @11,20 say 'This Information Has Been SAVED.'
      @17,20 say 'Hit Any Key To Continue.'
      wait ' '
      loop
    else
      delete record 2
      * deletes from dbase
      pack
    endif
```

```
            else
             * input error
             @10,20 say 'Please Enter Information BEFORE You Hit <RETURN>.'
             @16,20 say 'Hit Any Key To Continue.'
             wait ' '
             loop
            endif


          case madd = '2'
            * return to menu
            loop
          case madd = '3'
            save to maddcomp
            return
            * return to company information menu
          otherwise
            clear
            @10,20 say 'Invalid Value Entered. Not a <1,2,3>. '
            @16,20 say 'Hit Any Key To Continue.'
            wait ' '
            loop
        endcase madd
      else
      * confirm correct company
      clear
      @8,20  say 'This Company Is Already In The Database As: '
      @10,20 say 'COMPANY:'
      @10,35 say  coname
      @11,20 say  'ADDRESS:'
      @11,35 say costr1
      @12,35 say  costr2
      @13,35 say  trim(cocity) + "," + " " + costate + " " + cozip
      @18,20 say 'Hit Any Key To Continue.'
      wait ' '
      loop
      endif
    otherwise
      clear
      @10,20 say 'Invalid Number. Hit Any Key To Continue.'
      wait ' '
      loop
    endcase choice
enddo
save to maddcomp
return
* return control back to company menu
```

```
*************************************************************************
*  This program, delcomp.prg, deletes a company from the company
*  data base by company number or company name.
*************************************************************************
*  Database Used:  compinfo
*  Memory File:    mdelcomp
*  Index File:     ndxcompinfo
*************************************************************************


set safety off
set talk off
set exact on
set heading off
restore from mdelcomp

*  maintenance on index file
use compinfo
index on trim(coname) - trim(cocity) to ndxcompinfo
set index to ndxcompinfo
reindex


manswer = ' '
set device to screen
* loop to continue deleting companies
do while upper(manswer) = ' '
 clear
 @3,20 say 'D E L E T E   A   C O M P A N Y'
 @10,20 say ' 1. DELETE by Company Name'
 @11,20 say ' 2. VIEW All Companies In Database'
 @12,20 say ' 3. To QUIT'
 @14,20 say ' Enter Choice: '
 choice = ' '
 @14,37 get choice picture '9'
 read
 do case

   case choice = '1'
   * deleting by company name
    clear
   @7,20 say 'Enter Company Name: '
   skey = '                    '
   @8,20 get skey picture 'xxxxxxxxxxxxxxxxxxx'
   read
   skey = upper(skey)
```

```
    @10,20 say 'Enter The City The Company Resides In: '
    ckey = '                    '
    @11,20 get ckey picture 'xxxxxxxxxxxxxxxxxxx'
    read
    ckey = upper(ckey)
    compkey = trim(skey) - trim(ckey)

    case choice = '2'
     clear
     use
     use compinfo
     ?' COMPANY NAME                STREET              CITY'
     ?
     display all coname, costr1, cocity off
     ?
     ? '                    Hit Any Key To Continue'
     wait ' '
     loop

    case choice = '3'
    * option to quit
     save to mdelcomp
     return

    otherwise
    * Didn't select a valid (1/2)
    clear
    @9,20 say ' Invalid Number. Hit Any Key To Continue. '
    wait ' '
    loop
  endcase choice

 if choice  = '1'
  * key to index file is company name and city
  use compinfo index ndxcompinfo
  clear
  reindex
  find &compkey
 endif

* if company name has not been found,ask to either continue or quit
 if eof()
    @7,20 say 'No Record On File For This Company To Delete.'
    @9,20 say 'COMPANY:'
    @9,35 say skey
    @10,20 say 'CITY:   '
    @10,35 say ckey
    @16,20 say 'Hit Any Key To Continue.'
```

```
    wait ' '
    loop
  else

* confirm correct company to delete
  clear
  @5,20  say 'This Company Is In The Database As: '
  @7,20 say 'COMPANY NAME:'
  @7,35 say coname
  @8,20 say 'ADDRESS:'
  @8,35 say costr1
  @9,35 say costr2
  @10,35 say trim(cocity) +  ',' + ' ' + costate + ' ' + cozip
  @13,20 say '1. To DELETE'
  @14,20 say '2. To CONTINUE'
  @15,20 say '3. To QUIT'
  @17,20 say 'Enter Choice: '
  md = ' '
  @17,36 get md picture '9'
  read

  do case
    case md = '1'
      clear
      @9,20 say 'DELETE This Company (Y/N)? '
      mdele = ' '
      @9,50 get mdele picture 'a'
      read
      if mdele = 'Y' .or. mdele = 'y'
         delete
         pack
         @11,20 say 'This Company Has Been DELETED.'
         @17,20 say 'Hit Any Key To Continue.'
         wait ' '
         loop
      else
         @11,20 say 'This Company Has NOT Been DELETED.'
         @17,20 say 'Hit Any Key To Continue.'
         wait ' '
         loop
      endif

    case md = '2'
      loop
    case md = '3'
      save to mdelcomp
      return
```

```
    otherwise
      clear
      @9,20 say 'Invalid Number. Hit Any Key To Continue. '
      wait ' '
      loop
   endcase md
enddo
save to mdelcomp
return
* return control back to company menu
```

```
*****************************************************************
* This program, chgcomp.prg, updates a company file.
*****************************************************************
* Database Used:  Compinfo
* Memory File:    mchgcomp
* Index File:     ndxcompinfo
*****************************************************************

* set up
set safety off
set talk off
set exact off
set heading off

*  maintenance for index file
use compinfo
index on trim(coname) - trim(cocity) to ndxcompinfo
set index to ndxcompinfo
reindex
restore from mchgcomp
use compinfo index ndxcompinfo


manswer = ' '
set device to screen
do while upper(manswer) = ' '
 clear
 @3,13 say 'U P D A T E   C O M P A N Y  I N F O R M A T I O N'
 @11,20 say '1. CHANGE By Company Name'
 @12,20 say '2. VIEW All Companies In Database'
 @13,20 say '3. To QUIT'
 @15,20 say 'Enter Choice: '
 choice = ' '
 @15,36 get choice picture '9'
 read
 do case
    case choice = '1'
    * changing by company name
    clear
    @9,20 say 'ENTER Company Name: '
    skey = '                    '
    @10,20 get skey picture 'xxxxxxxxxxxxxxxxxxx'
    read
    skey = upper(skey)
    @12,20 say 'ENTER The City The Company Resides In: '
    ckey = '                    '
```

```
    @13,20 get ckey picture 'xxxxxxxxxxxxxxxxxxx'
    read
    ckey = upper(ckey)
    cokey = trim(skey) - trim(ckey)
    case choice = '2'
     clear
     use compinfo
     ?'COMPANY NAME                STREET                CITY'
     display all coname, costr1, cocity off
     use
     ?
     ? '                    Hit Any Key To Continue.'
     wait ' '
     loop
    case choice = '3'
     save to mchgcomp
     return
    otherwise
     clear
     @9,20 say 'Invalid Number. Hit Any Key To Continue.'
     wait ' '
     loop
    endcase choice
    if choice = '1'
     * key for this index is coname - cocity
      use compinfo index ndxcompinfo
      clear
      reindex
      find &cokey
    endif
if eof()
  @7,20 say 'No Record On File For This Company. '
  @9,20 say 'COMPANY:'
  @9,35 say skey
  @10,20 say 'CITY:   '
  @10,35 say ckey
  @17,20 say 'Hit Any Key To Continue.'
  wait ' '
  loop
else
* confirm correct company for changing info
clear
@5,20  say 'This Company Is In The Database As: '
@7,20 say 'COMPANY NAME:'
@7,35 say coname
@9,20 say 'CITY       :'
@9,35 say  trim(cocity)
```

```
@12,20 say '1. To CHANGE INFORMATION '
@13,20 say '2. To ENTER Another '
@14,20 say '3. To QUIT'
@16,20 say 'Enter Choice: '
mchange = ' '
@16,36 get mchange picture '9'
read
mrecno = recno()
do case
 case mchange = '1'
  use
  use compinfo
  goto mrecno
  temp1 = sched1
  temp2 = sched2
  temp3 = sched3
  temp4 = sched4
  temp5 = sched5
  temp6 = sched6
  temp7 = sched7
  numsched = no_sched
  numcancl = cancelcnt
  set format to comp
  change record mrecno
  close format
  save to mchgcomp
  goto mrecno
  do addsched
  restore from mchgcomp
  replace no_sched with nsched
  goto mrecno
  save to mchgcomp
  do sched
  restore from mchgcomp
  use
case mchange = '2'
  loop
case mchange = '3'
  save to mchgcomp
  return
```

```
   otherwise
     clear
     @9,20 say 'Invalid Value Entered.  Not a <1,2,3>.'
     @16,20 say 'Hit Any Key To Continue.'
     wait ' '
     loop
   endcase mchange
enddo
save to mchgcomp
return
```

```
do case
 case choice = '3'
   save to maddstud
   return
 case choice = '2'
   clear
   use sinfo
   ?'LAST NAME        FIRST NAME     STUDENT NUMBER'
   ?
   display all slastnam, sfirstnm, ssn  off
   ?
   ? '         Hit Any Key To Continue.'
   wait ' '
   loop

 case choice = '1'
  clear
  set confirm off
  skey = '          '
  @10,18 say 'Enter Student Number: '
  @11,18 get skey picture '@r 999-99-9999'
  read
  find &skey
  if eof()
   set confirm on
   madd = ' '
   clear
   @7,18 say 'No Record On File For This Student.'
   @10,18 say 'Would You Like To ADD This Student?'
   @12,18 say '1.  To ADD This Student'
   @13,18 say '2.  To ENTER Another Student Number'
   @14,18 say '3.  To QUIT'
   @16,18 say 'Enter Choice: '
   @16,35 get madd picture '9'
   read
   do case
    clear
    case madd = '1'
     clear
     first = '                    '
     initial = ' '
     last = '                    '
     @9,15 say 'Enter Student First Name'
     @9,41 get first picture 'aaaaaaaaaaaaaaaaaa'
     read
     @11,15 say 'Middle Initial'
     @11,32 get initial picture 'a'
     read
```

```
@13,15 say 'Last Name'
@13,41 get last picture 'aaaaaaaaaaaaaaaaaaa'
read
last = upper(last)
first = upper(first)
initial = upper(initial)
use
use sinfo
set format to sinfo
count to mreccount
goto top
insert
close format
count to mrec2
clear
* if mrec2 <> mrecount then user inserted data and we
* want to know if they want to save it
if mrec2 <> mreccount
  myn = ' '
  @10,18  say 'Do you want to SAVE this (Y/N)? '
  @10,53 get myn picture 'a'
  read
  if myn = 'Y' .or. myn = 'y'
   replace record 2 sfirstnm with upper(first)
   replace record 2 smidinit with upper(initial)
   replace record 2 slastnam with upper(last)
   replace record 2 ssn with skey
   temp = substr(curriculum,1,1)
   temp2 = substr(curriculum,2,1)
   if temp = '0'
    replace curriculum with ' ' + substr(curriculum,2,1)
   endif
   if temp2 = ' '
    replace curriculum with ' ' + substr(curriculum,1,1)
   endif
  else
   if myn = 'n' .or. myn = 'N'
    delete record 2
    pack
   endif
  endif
  clear
else
  clear
  @10,18 say 'Please Enter Information Before You Hit <RETURN>'
  @16,18 say 'Hit Any Key To Continue.'
  wait ' '
  loop
endif
```

```
      case madd = '2'
        loop
      case madd = '3'
        save to maddstud
        return
      otherwise
        clear
        @10,18 say 'Invalid Value Entered. Not a <1,2,3>. '
        @16,18 say 'Hit Any Key To Continue.'
        wait ' '
        loop
      endcase madd
    else
  * confirm correct student
    set confirm on
    clear
    @9,18  say 'This Student Is Already In The Database As: '
    @11,18 say  trim(sfirstnm) + " " + smidinit + " " + slastnam
    @17,18 say 'Hit Any Key To Continue.'
    wait ' '
    loop
    endif
  otherwise
    clear
    @10,18 say 'Invalid Number. Hit Any Key to Continue.'
    wait ' '
    loop
  endcase choice
  enddo
  save to maddstud
  return
  * return control back to student menu
```

```
*********************************************************************
* This program, delstud.prg, deletes a student from the
* student database.
*********************************************************************
* Database Used:  sinfo
* Memory File:    mdeletest
* Index File:     ndxsinfo
*********************************************************************

set safety off
set exact on
set talk off
set heading off
set confirm on
restore from mdelest

* maintenance on index files
use sinfo
index on ssn to ndxsinfo
set index to ndxsinfo
index on trim(slastnam) - trim(sfirstnm) to ndxsname
set index to ndxsname
reindex

use sinfo
found = 'f'
manswer = ' '

set device to screen
do while upper(manswer) = ' '
 reindex
 clear
 @3,20 say 'D E L E T E   A   S T U D E N T'
 @10,20 say '1.  DELETE by Student Number'
 @11,20 say '2.  DELETE by Name'
 @12,20 say '3.  VIEW All Students In Database'
 @13,20 say '4.  QUIT'
 @15,20 say 'Enter Choice: '
 choice = ' '
 @15,36 get choice picture '9'
 read
 do case
    case choice = '1'
    * deleting by social security number
    set confirm off
    clear
    @7,20 say 'Enter Student Number: '
    skey = '          '
```

```
      @8,25 get skey picture '@r 999-99-9999'
      read
      set confirm on
     case choice = '2'
      * deleting by name
      clear
      @9,20 say 'Enter Student First Name: '
      fkey = '                    '
      @10,20 get fkey picture 'xxxxxxxxxxxxxxxxxxx'
      read
      fkey = upper(fkey)
      @12,20 say 'Enter Last Name'
      nkey = '                    '
      @13,20 get nkey picture 'xxxxxxxxxxxxxxxxxxx'
      read
      nkey = upper(nkey)
      name = trim(nkey) - trim(fkey)
    case choice = '3'
      clear
      use sinfo
      ?'LAST NAME        FIRST NAME     STUDENT NUMBER'
      ?
      display all slastnam, sfirstnm, ssn off
      ?
      ? '          Hit Any Key To Continue'
      wait ' '
      loop
    case choice = '4'
      save to mdeletest
      return
    otherwise
      * for all poo poo entries
      clear
      @9,20 say 'Invalid Choice. Hit Any Key to Continue.'
      wait ' '
      clear
      loop
endcase choice
if choice = '1'
  * key to index file is ssn
  use sinfo index ndxsinfo
  clear
  reindex
  find &skey
endif
```

```
 if choice = '2'
   * key to index file is slastnam - sfirstnm
   use sinfo index ndxsname
   clear
   reindex
   find &name
 endif
 if eof() .and. found = 'f'
   clear
   @9,20 say 'No Record On File For This Student'
   @16,20 say 'Hit Any Key To Continue.'
   wait ' '
   loop
 else

* confirm correct student
 clear
 @7,20  say 'This Student Is In The Database As: '
 @9,20 say  'NAME:'
 @9,35 say trim(sfirstnm) + " " + smidinit + " " + slastnam
 @10,20 say 'NUMBER:'
 @10,35 say ssn
 @13,20 say '1. To DELETE'
 @14,20 say '2. To ENTER Another'
 @15,20 say '3. To QUIT'
 @17,20 say 'Enter Choice: '
 mansw = ' '
 @17,36 get mansw picture '9'
 read
 do case
 case  mansw = '1'
     clear
     @9,20 say 'DELETE This Person (Y/N)? '
     mdele = ' '
     @9,49 get mdele picture 'a'
     read
     if mdele = 'Y'.or. mdele = 'y'
        delete
        pack
        reindex
        @11,20 say 'This Person Has Been DELETED.'
        @15,20 say 'Hit Any Key To Continue.'
        wait ' '
     else
        @11,20 say 'This Person Has NOT Been DELETED.'
        @15,20 say 'Hit Any Key To Continue.'
        wait ' '
     endif
 case mansw = '2'
     loop
```

```
case mansw = '3'
    save to mdeletest
    return
```

```
  otherwise
      clear
      @10,10 say 'Invalid Choice. Hit Any Key to Continue.'
      wait ' '
      loop
  endcase mansw
endif
enddo
save to mdeletest
return
* return control back to student menu
```

```
*********************************************************************
*  This program, chgstud.prg, updates a student file.
*********************************************************************
*  Database Used:  Sinfo
*  Memory File:    maddstud
*  Index File:     ndxsinfo
*                  ndxsname
*********************************************************************

* set up
clear all
set safety off
set talk off
set exact on
set heading off
set confirm on
restore from maddstud

*  maintenance on index files
use sinfo
index on ssn to ndxsinfo
set index to ndxsinfo
use
use sinfo
index on trim(slastnam) - trim(sfirstnm) to ndxsname
set index to ndxsname
use

manswer = ' '
set device to screen
do while upper(manswer) = ' '
 clear
 @3,13 say 'C H A N G E   S T U D E N T   I N F O R M A T I O N'
 @10,20 say '1. CHANGE By Student Number'
 @11,20 say '2. CHANGE By Student Name'
 @12,20 say '3. VIEW All Students In Database'
 @13,20 say '4. QUIT'
 @15,20 say 'Enter Choice: '
 achoice = ' '
 @15,36 get achoice picture '9'
 read
 choice = achoice

 do case

    case choice = '1'
    * changing by student number
    set confirm off
```

```
    clear
    @9,20 say 'Enter Student Number: '
    skey = '          '
    @10,20 get skey picture '@r 999-99-9999'
    read
    set confirm on
    case choice = '2'
    * changing by student name
    clear
    @9,15 say 'Enter Student First Name: '
    fkey = '                '
    @9,44 get fkey picture 'aaaaaaaaaaaaaaaaaa'
    read
    fkey = upper(fkey)
    @11,15 say 'Student Last Name: '
    nkey = '                '
    @11,44 get nkey picture 'aaaaaaaaaaaaaaaaaa'
    read
    nkey = upper(nkey)
    name = trim(nkey) - trim(fkey)

    case choice = '3'
     clear
     use sinfo index ndxsname
     ? 'LAST NAME       FIRST NAME     STUDENT NUMBER'
     ?
     display all slastnam, sfirstnm, ssn off
     ?
     ? '          Hit Any Key To Continue'
     use
     Wait ' '
     loop

    case choice = '4'
     save to mchgstud
     return

    otherwise
     clear
     @9,20 say 'Invalid Number. Hit Any Key To Continue.'
     wait ' '
     loop
endcase choice

if choice = '1'
     * key for this index file is ssn
     use sinfo index ndxsinfo
     find &skey
endif
```

```
if choice = '2'
     * key for this index file is slastnam - sfirstnm
     use sinfo index ndxsname
     find &name
endif

if eof()
 clear
 @10,20 say 'No Record On File For This Student '
 @16,20 say 'Hit Any Key To Continue.'
 wait ' '
 loop

else
 * confirm correct student for changing info
 clear
 @7,20 say 'This Student Is In The Database As: '
 @9,20 say 'NAME:'
 @9,35 say  trim(sfirstnm) + ' ' + smidinit + ' ' + slastnam
 @10,20 say 'NUMBER:'
 @10,35 say ssn
 @13,20 say '1. To CHANGE Info On This Student'
 @14,20 say '2. To ENTER Another'
 @15,20 say '3. To QUIT'
 @17,20 say 'Enter Choice: '
 mchange = ' '
 @17,36 get mchange picture '9'
 read
 last = slastnam
 initial = smidinit
 first = sfirstnm
 skey = ssn
 clear

 do case
  case mchange = '1'
   mrecno = recno()
   use
   use sinfo
   set format to sinfo
   change record mrecno
   close format
   use
   clear
```

```
    @10,13 say 'These Changes Have Been Saved. Hit Any Key To Continue.'
    wait ' '
   case mchange = '2'
    use
    loop
   case mchange = '3'
    use
    save to mchgstud
    return
   otherwise
    use
    @10,20 say 'Invalid Number. Hit Any Key To Continue.'
    wait ' '
    loop
  endcase mchange
 endif
enddo

* return control back to student menu
```

```
*****************************************************************
* This program, emreport.prg, allows the adding and
* updating of the employment report form for students.
*****************************************************************
* Database Used:  sinfo
*                 ereport
* Memory File:    maddstud
* Index File:     ndxsname
*                 ndxename
*****************************************************************

set talk off
set safety off
set confirm on
set heading off
set exact on
set date american
restor from maddstud
clear all

* maintenance on index file
use sinfo
index on trim(slastnam) - trim(sfirstnm) to ndxsname
set index to ndxsname
use
use ereport
index on trim(slastnam) - trim(sfirstnm) to ndxename
set index to ndxename

count to mrec2
if recno() = 1
  insert blank
  insert blank before
endif
use
manswer = ' '
set device to screen
do while upper(manswer) = ' '
  clear
  number = ' '
  @3,25 say 'Employment Report'
  @10,25 say '1. To ADD An Employment Report '
  @11,25 say '2. To VIEW An Employment Report'
  @12,25 say '3. QUIT'
  @14,25 say 'Enter Choice: '
  @14,41 get number picture '9'
  read
```

```
choice = number
do case
  case choice = '3'
    save to maddstud
    return
  case choice = '2'
    clear
    first = '                    '
    last = '                     '
    use ereport index ndxename
    @7,17 say 'View An Employment Report'
    @10,10 say 'Enter Student First Name:'
    @10,38 get first
    read
    @12,10 say '              Last Name:'
    @12,38 get last
    read
    first = upper(first)
    last = upper(last)
    nkey = trim(last) - trim(first)
    find &nkey

    if eof()
      @14,17 say 'No Employment Report For This Student.'
      @17,17 say 'Hit Any Key To Continue.'
      wait ' '
    else
      place = recno()
      use
      use ereport
      set format to e2report.fmt
      change record place
      close format
    endif

    use
    loop

 case choice = '1'
  clear all
  use sinfo index ndxsname
  first = '                    '
  last = '                     '
  clear
  @9,15 say 'Enter Student First Name'
  @9,41 get first picture 'aaaaaaaaaaaaaaaaaaa'
  read
  @11,15 say 'Enter Student Last Name'
  @11,41 get last picture 'aaaaaaaaaaaaaaaaaaa'
  read
```

```
last = upper(last)
first = upper(first)
nkey = trim(last) - trim(first)
find &nkey

if eof()
 @13,15 say 'Need to Input A Student Data Sheet First.'
 @17,15 say 'Hit Any Key To Continue.'
 use
 wait ' '
 loop
else
 use
 use ereport index ndxename
 reindex
 find &nkey
 if eof()
  use
  use ereport
  set format to ereport
  count to mreccount
  goto top
  insert
  close format
  count to mrec2
  clear
  * if mrec2 < > mrecount then user inserted data and we
  * want to know if they want to save it
  if mrec2 < > mreccount
    myn = ' '
    @10,18  say 'Do you want to SAVE this (Y/N)? '
    @10,53 get myn picture 'a'
    read
    if myn = 'Y' .or. myn = 'y'
     replace record 2 sfirstnm with upper(first)
     replace record 2 slastnam with upper(last)
     replace record 2 curriculum with upper(curriculum)
     replace record 2 location with upper(location)
    else
     if myn = 'n' .or. myn = 'N'
      delete record 2
      pack
     endif
    endif
    use
```

```
      else
        clear
        @10,18 say 'Please Enter Information Before You Hit <RETURN>'
        @16,18 say 'Hit Any Key To Continue.'
        use
        wait ' '
        loop
      endif

    else
      @13,17 say 'Existing Employment Form, Cannot Add Another.'
      @17,17 say 'Hit Any Key To Continue.'
      use
      wait ' '
      loop
    endif
  loop
  endif
  * end of choice 1

otherwise
  clear
  @10,18 say 'Invalid Number. Hit Any Key to Continue.'
  wait ' '
  loop
endcase choice
enddo
save to maddstud
return
* return control back to student menu
```

```
*********************************************************************
* This program, stat1.prg, calculates the percentage of
* registered students that are graduating, seeking employmt,
* or furthering their education.
*********************************************************************
* Database Used:  ereport
* Memory File:    mstat
* Index File:     None
*********************************************************************

set talk off
set exact off
set safety off
restore from mstat
clear all
clear
@4,17 say 'Calculation of Total and Percentage of '
@5,17 say '        Graduating Students'

use ereport
today = ctod("01" + "/" + "01" + "/" + "80")
@8,17 say 'Enter Date (mm/da/yr):'
@8,50 get today picture '99999999'
read
numb = 0
@10,17 say 'Enter Total Number of Graduates:'
@10,56 get numb picture '9999'
read
number = numb
count to total
total1 = 0
total2 = 0
total3 = 0
total4 = 0
goto top
position = recno()
do while position < > total
  graddate = ctod(mdegdate + "/" + "01" + "/" + ydegdate)
  if (graddate - today < 365) .and. (graddate - today < > 0)
     total1 = total1 + 1
     if hired
        total2 = total2 + 1
     endif
     if seekempl
        total3 = total3 + 1
     endif
```

```
    if moreeduc
      total4 = total4 + 1
    endif
  endif
  position = position + 1
  goto position
enddo
* calculation time
total1 = (total1 / number) * 100
total2 = (total2 / number) * 100
total3 = (total3 / number) * 100
total4 = (total4 / number) * 100

use
* initialize values in percent database
use percent
replace studreg with 0
replace studcnt with 0
replace hired with 0
replace seeking with 0
replace gradsch with 0
* store values into percent database
replace studcnt with number
replace studreg with total1
replace hired with total2
replace seeking with total3
replace gradsch with total4

@12,17 say 'Total Number Of Graduating Students:'
@12,60 say studcnt picture '@r 9999'
@13,17 say 'Graduating Students Registered:'
@13,60 say studreg picture '@r 999%'
@14,17 say '                    Hired:'
@14,60 say hired picture '@r 999%'
@15,17 say '          Seeking Employment:'
@15,60 say seeking picture '@r 999%'
@16,17 say '              Graduate School:'
@16,60 say gradsch picture '@r 999%'

@18,17 say 'Calculations Have Been Completed.'
@21,17 say 'Hit Any Key To Continue.'
wait ' '
use
save to mstat
return
```

```
*****************************************************************
* This program, stat2.prg, calculates the semester
* recruiting statistics and the company cancellation
* rate for all companies.
*****************************************************************
* Database Used:  recruiter
*                 compinfo
* Memory File:    mstat
* Index File:     None
*****************************************************************



clear
set talk off
set safety off
set exact off
set device to screen
restore from mstat

@7,17 say  'Calculation of recruiting statistics and '
@8,17 say  '  cancellation rate for all companies'

use recruiter
*  initialize all values to 0
  replace schedules with 0
  replace recruits with 0
  replace cancels with 0
  replace total with 0
  replace percent with 0
use

use compinfo
*  totals for number_of_schedules made by all companies
*  totals of all recruiters sent by all companies
*  totals of all cancellations made by all companies
sum all no_sched to temp
sum all recruit1 to temp1
sum all recruit2 to temp2
sum all recruit3 to temp3
sum all recruit4 to temp4
sum all recruit5 to temp5
sum all recruit6 to temp6
sum all recruit7 to temp7
sum all cancelcnt to temp8
use
```

```
use recruiter
* store totals into recruiter dbase to print out when desired
replace schedules with temp
replace recruits with temp1 + temp2 + temp3 + temp4 + temp5 + temp6 + temp7
replace cancels with  temp8
replace total with (temp - temp8)
replace percent with ((total / schedules) * 100)
@13,17 say 'Calculations Have Been Completed.'
@16,17 say 'Hit Any Key To Continue.'
wait ' '
clear
display schedules,cancels,recruits, total,percent off
@16,17 say 'Hit Any Key To Continue.'
wait ' '
use
save to mstat
return
```

```
*****************************************************************
* This program, stat4.prg, calculates the number of
* companies that recruit by curriculum, gpa or both.
*****************************************************************
* Databases Used:  compinfo
*                  gpa
* Memory File:     mstat
* Index File:      None
*****************************************************************

set talk off
set exact off
set safety off
restore from mstat

clear
@7,17 say 'Calculation Of Companies Recruiting By'
@8,17 say '     Curriculum and/or GPA'

use compinfo
* get the totals for companies that recruit by curriculum,
*by gpa, and by both
count for bycurr  to temp1
count for bygpa  to temp2
count for bycurr .and. bygpa  to temp3
use

use gpa
* clear out old totals, then replace with new counts
replace bycurr with 0
replace bygpa with 0
replace both with 0
replace bycurr with temp1
replace bygpa with temp2
replace both with temp3
@13,17 say 'Calculations Have Been Completed.'
@16,17 say 'Hit Any Key To Continue.'
wait ' '
clear
@7,17 say 'Company Recruiting Information:'
set heading off
@11,1 say ' '
display bycurr off
@12,12 say ': Companies Recruiting by Curriculum '
display bygpa off
@13,12 say ': Companies Recruiting by GPA'
```

```
display both off
@14,12 say ': Companies Recruiting by Both'
@18,17 say 'Hit Any Key To Continue.'
wait ' '
save to mstat
return
```

```
*********************************************************************
* This program, stat5.prg, calculates the yearly placement
* of registered students by curriculum.
*********************************************************************
* Database Used:  curr
*                 ereport
* Memory File:    ndxcurr
*                 ndxename
* Index File:     ndxcurr
*********************************************************************


clear
set talk off
set safety off
set exact off
set heading off
set device to screen
restore from mstat

@7,10 say 'Calculation of Placement Information by Curriculum'

* maintenance on index files, and also, initialize all totals to 0
use curr
replace all htotal with 0
index on trim(curriculum) to ndxcurr
set index to ndxcurr
use
use ereport
index on trim(slastnam) - trim(sfirstnm) to ndxename
set index to ndxename
reindex
use
use ereport
position = recno()
count to total_no
do while position <= total_no
* loop until no more students in employment report
  goto position
  if hired
    curr_key = curriculum
    curr_key = trim(curr_key)
    use
    use curr index ndxcurr
    * indexed by  curriculum
```

```
    find &curr_key
    if .not. eof()
        replace htotal with htotal + 1
    endif
    use
  endif
  position = position + 1
  use ereport
enddo

* suming the number of placements per college
do stat6
clear
@9,20 say 'Calculations Have Been Completed.'
@16,24 say 'Hit Any Key To Continue.'
wait ' '
flag = .t.
do while flag
 clear
 @5,20 say 'View The Statistics?'
 @7,20 say '1. To View Totals By Colleges'
 @8,20 say '2. To View Totals By Curriculum'
 @9,20 say '3. To Quit'
 choice = ' '
 @11,20 say 'Enter Choice: '
 @11,36 get choice picture '9'
 read
 do case
  case choice = '1'
   set device to screen
   clear
   use curr
   set index to ndxcurrname
   set heading off
   ? '    *****  PLACEMENT TOTATLS BY CURRICULUM & COLLEGE  *****'
   ?
   ? '                                    College'
   ? '          Colleges                    Totals'
   ?
   display all '       ', name, col_total for "All" $ name off
   ?
   ?
   ?'                  Hit Any Key To Continue'
   wait ' '
   loop
```

```
case choice = '2'
  clear
  @5,23 say 'College of Agriculture'
  ?
  ?'     Curriculum                          Number'
  ?'       Code          Name               Employed'
  ?
  display all '       ', curriculum, name, htotal for college = '1' .and. htotal > 0 off
  ?
  ?
  ?'     Hit Any Key To Continue To Next College Curriculums'
  wait ' '
  clear
  @5,20 say 'College of Architecture And Design'
  ?
  ?'     Curriculum                          Number'
  ?'       Code          Name               Employed'
  ?
  display all '       ', curriculum, name, htotal for college = '2' .and. htotal > 0 off
  ?
  ?
  ?'     Hit Any Key To Continue To Next College Curriculums'
  wait ' '
  clear
  @5,22 say 'College of Arts and Sciences'
  ?
  ?'   Curriculum                            Number'
  ?'     Code              Name              Employed'
  ?
  display all '       ', curriculum, name, htotal for college = '3' .and. htotal > 0
  ?
  ?
  ?'     Hit Any Key To Continue To Next College Curriculums'
  wait ' '
  clear
  @5,22 say 'College of Business'
  ?
  ?'   Curriculum                            Number'
  ?'     Code              Name              Employed'
  ?
  display all '       ', curriculum, name, htotal for college = '4' .and. htotal > 0
  ?
  ?
  ?'     Hit Any Key To Continue To Next College Curriculums'
  wait ' '
  clear
```

```
    @5,22 say 'College of Education'
    ?
    ?'  Curriculum                              Number'
    ?'     Code              Name              Employed'
    ?
    display all '       ', curriculum, name, htotal for college = '5'
    @20,6 say 'Hit Any Key To Continue To Next College Curriculums'
    wait ' '
    clear
    @5,22 say 'College of Engineering'
    ?
    ?'  Curriculum                              Number'
    ?'     Code              Name              Employed'
    ?
    display all '       ', curriculum,name,htotal for college = '6' .and. htotal > 0
    ?
    ?'       Hit Any Key To Continue To Next College Curriculums'
    wait ' '
    clear
    @5,22 say 'College of Human Ecology'
    ?
    ?'  Curriculum                              Number'
    ?'     Code              Name              Employed'
    ?
    display all '       ', curriculum,name,htotal for college = '7' .and. htotal > 0
    ?
    ?'                   Hit Any Key To Continue'
    wait ' '
    loop
  case choice = '3'
    save to mstat
    return
  otherwise
    @14,20 say 'Not A Valid Selection, Hit Any Key To Continue'
    wait ' '
    loop
  endcase
enddo
```

```
*************************************************************************
* This program, stat7.prg, calculates the number of registered
* students geographical placement within the 50 united states.
* ***********************************************************************
* Database Used:  geolocation
* Memory File:    mstat
* Index File:     none
*************************************************************************

set talk off
restore from mstat
set safety off

*maintenance on index files, and also, initialize all totals to 0
use geolocation
* all state counters are zero
replace all al with 0, ak with 0, az with 0, ar with 0, ca with 0
replace all co with 0, ct with 0, de with 0, fl with 0, ga with 0
replace all hi with 0, id with 0, il with 0, in with 0, ia with 0
replace all ks with 0, ky with 0, la with 0, me with 0, ma with 0
replace all md with 0, mi with 0, mn with 0, ms with 0, mo with 0
replace all mt with 0, ne with 0, nv with 0, nh with 0, nj with 0
replace all nm with 0, ny with 0, nc with 0, nd with 0, oh with 0
replace all ok with 0, or with 0, pa with 0, ri with 0, sc with 0
replace all sd with 0, tn with 0, tx with 0, ut with 0, vt with 0
replace all va with 0, wa with 0, wv with 0, wi with 0, wy with 0

hired_cnt = 0
menu = .t.
do while menu
clear
choice = ' '
@5,12 say 'Computing Statistics For Geographic Placement By College'
?
?
?'          1. College of Agriculture'
?'          2. College of Architecture & Design'
?'          3. College of Arts & Sciences'
?'          4. College of Business Administration'
?'          5. College of Education'
?'          6. College of Engineering'
?'          7. College of Human Ecology'
?'          8. Quit'
@17,18 say 'Enter Choice: '
@17,36 get choice picture '9'
read
```

```
if choice = '8'
  save to mstat
  return
endif
if choice = '9' .or. choice = '0'
  clear
  @7,17 say 'Invalid Selection.  Hit Any Key To Continue.'
  wait ' '
  loop
endif

coll = choice
use
use ereport
position = recno()
count to emp_cnt
goto top
do while position <= emp_cnt
  goto position
  if hired
    state = location
    tempcurr = curriculum
    hired_cnt = hired_cnt + 1
    use
    do case
      case choice = '1'
        use geolocate
        index on college1 to ndxcoll1
        set index to ndxcoll1
        menu = .f.
      case choice = '2'
        use geolocate
        index on college2 to ndxcoll2
        set index to ndxcoll2
        menu = .f.
      case choice = '3'
        use geolocate
        index on college3 to ndxcoll3
        set index to ndxcoll3
        menu = .f.
      case choice = '4'
        use geolocate
        index on college4 to ndxcoll4
        set index to ndxcoll4
        menu = .f.
```

```
      case choice = '5'
        use geolocate
        index on college5 to ndxcoll5
        set index to coll5
        menu = .f.
      case choice = '6'
        use geolocate
        index on college6 to ndxcoll6
        set index to coll6
        menu = .f.
      case choice = '7'
        use geolocate
        index on college7 to ndxcoll7
        set index to coll7
        menu = .f.
      otherwise
        clear
        @12,22 say 'Ivalid Selection, Hit Any Key To Continue'
        wait ' '
        loop
    endcase


    find &tempcurr
    if .not. eof()
      temprec = recno()
      replace &state with &state + 1
    endif
  endif
  * not hired
 position = position + 1
 use
 use ereport
enddo
enddo
* print stats
clear
do while .t.
 choice = ' '
 @8,20 say 'Computation Is Complete'
 @10,18 say '1. VIEW Statistics'
 @11,18 say '2. QUIT'
 @13,18 say 'Enter Choice:'
 @13,36 get choice picture '9'
 read
```

```
do case
  case choice = '1'
   clear
   @8,12 say 'STATISTICS FOR GEOGRAPHIC PLACEMENT OF REGISTERED STUDENTS'
   @12,20 say 'Students With An Employment Report: '
   @12,55 get emp_cnt
   @14,20 say 'Number Of Students Hired: '
   @14,55 get hired_cnt
   @18,20 say 'Hit Any Key To Continue'
   wait ' '
   set heading on
   use geolocate
   do case
     case coll = '1'
      collg = 'college1'
     case coll = '2'
      collg = 'college2'
     case coll = '3'
      collg = 'college3'
     case coll ='4'
      collg = 'college4'
     case coll = '5'
      collg = 'college5'
     case coll = '6'
      collg = 'college6'
     case coll = '7'
      collg = 'college7'
   endcase
      clear


display all &collg,AL,AK,AZ,AR,CA,CO,CT,DE,FL,GA,HI,ID,IL,IN,;
IA,'          ',KS,KY,LA,ME,MD,MA,MI,MN,MS,MO,MT,NE,NV,NH,;
NJ,'          ',NM,NY,NC,ND,OH,OK,OR,PA,RI,SC,SD,TN,TX,UT,;
VT,'          ',VA,WA,WV,WI,WY while &collg < > '    '
   ?
   ? '                    Hit Any Key To Continue '
   wait ' '
   clear
   loop
  case choice = '2'
   save to mstat
   return
  otherwise
   @15,18 say 'Invalid Choice. Hit Any Key To Try Again'
   wait ' '
   loop
  endcase
enddo
```

Design And Implementation Of A Computerized System For A Manual Artisan Office

by

Jeanie L. Jester Gay

B.S. Kansas State University, 1986

An Abstract of a
MASTER'S REPORT

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1987

Abstract

The quality of placement programs at universities and colleges determine to a great extent, the types of jobs opportunities their graduates are afforded. A good computer placement program offers many advantages to a placement center. This paper presents the functional evaluation, design and implementation of a computer placement program for the Career Planning and Placement Center at Kansas State University, an organization which has an artisan type of office. A relational database was designed and implemented in in dBaseIII. Several application programs were also implemented; these help ensure user friendliness, and generate various statistics. A users manual is included to aid in executing the program. The challenges, problems and solutions to those problems in the design of a computerized system within the framework of an artisan office are discussed.