NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

DEVELOPMENT OF FACULTYS' STUDENTS iNFORMATION SYSTEM

Graduation Project
COM- 400

Student: Mahmoud ALMASSRI (991195)

Supervisor: Assoc. Prof. Dr Rahib ABEYV

Nicosia - 2003

ACKNOWLEDGMENT

"First, I would like and foremost to thank Allah whom its accomplishment would not have been possible.

Second, I would like also to deeply thank my supervisor Ass. Prof. Dr RAHIB ABEYV for his invaluable advice and belief in my work and myself over the course of this Graduation Project.

Third, I am deeply indebted to my parents for their love and financial support. They have always encouraged me to pursue my interests and ambitions throughout life.

Especially My Father: Atta Almassri(Abu Mohammad), My uncle: Abd Elhamead Almassri(Abu Mohammad), My uncle: Yahya Almassri(Abu Amer)

Forth, I thank my family for their constant encouragement and support during the preparation of this project.

Last but not least I would also like to thank all of my friends especially Reyad Bader and Kaled Almassri they were always available for my assistance throughout this project."

ABSTRACT

The purpose of this project is the development of faculty's student information system. The main problems which we have come across in faculty student's information system have been analyzed. The algorithms for student registration, course registration, GPA, CGPA calculations are described. The main structures and elements of database system for these problems are clarified. The operation principles of each blocks of the information system are modeled in Delphi programming language. The developed system allows to make registration of courses and students, calculation of GPA and CGPA easily and decreasing time response of the system. Over the past decades people have transferred in maintaining records through paper and pen, and now we are evolving into the technology aria.

This project has taken a lot of time and effort to send out a very clear and simple program in Delphi concerning any university.

This system has been designed in a way that it would work more speedy than the normal record keeping system.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

INTRODUCTION

The aim of the project is the development of faculty's students' information system using Delphi programming. The intended audience for this project includes the follow:

(1) Codes – any codes that are responsible for creating and maintaining the data elements and file description specified in this project.

(2) Screens – those individuals who wish to view the data collected and processed as part of the Development Students Courses from "summary" or "subtotaled" point of view.

In this project, I use one of the programming languages that we are learned in our university - Delphi programming language. In this language there are many things that we can use to create any kind of project. But in this project I use some standard components and database components to create this project. In this language there is special procedure called Database Desktop to create some tables that used in the project. We will see this later, regarding this program, which basically divided, into tow main sections: Registration and Calculation of GPA. A section for Registration, which consist of student's information, courses, grades. Another section is the Calculation, which includes calculation of GPA of students in semester and CGPA of the students in all semesters. Each member has been assigned a special form in this program. These forms are updated consistently depending on his working hours. Another important think about this project is to searching student Courses in any semester or all Courses of student during his studies and also searching student grades in any semester with GPA of that semester or grades of student for all semesters with CGPA during his studies and also got out a report about them. (Described in chapter 3). These are the main aides about this project.

CHAPTER 1

DESCRIPTION OF THE PROBLEMS OF FACULTY'S STUDENTS INFORMATION SYSTEM

1.1. Introduction

Every university has programs that register the student information, the student courses, and student grades and calculate the GPA and CGPA for the students. Because that I make this project to do these schedules. We know when we want to do the program by using database information there are two main factors one is data definition language that create table and index and view, second is data manipulation language that used special sentences to control and remote the tables and indexes.

In this project the three main problems are solved:

- Registration of student's
- Registration of Courses and Grades
- Calculation the GPA and CGPA of the students

1.2 Registration of student's

Here, at the beginning we enter the student information and keep it in the file, these information's are:

Student No. : Every student takes special identification number in the university.

Name: The first name of student.

Surname: The last name of student.

Country: The name of country for the student.

Birthday: The date of birth for the student.

Birthplace: The name of birthplace of the student.

Date: The registration date of student.

Department: The name of department which chosen by the student.

Transfer: here, if the student is transfer from any university it will record the name of that university and exempted courses.

1.3. Registration of Courses and Grades

1.3.1. Courses

Here, after registration of the student's information, we check all these information and save it in data file. Registration of Courses mode will records the courses of students. We know there is a system for the university about records the Courses and there are three semesters Fall Term, Spring Term and Summer Term and also there is limitation according the number of courses that will be taken. For these things we must sure that the student is registered in the university or not. It is realized by searching the number of student. We use some information when we record the courses of students. These are:

Year: The year of registration of courses.
Semester: The name of semester (fall, spring, summer).
Course name: The name of course.
Course code: The title of the course.
Credit: The degree of the course.

1.3.2. Grades

After finishing the records of the courses and checking and saving they, we will record the grades of the courses at the end of each semester. This is realized by choosing the course, which was taken by the students in the semester, and recording the corresponding grades of these students.

1.4. How we can calculate the:
GPA of students, CGPA of students

We know each grade has a value for example AA=4.0, BA=3.5, BB=3.0, CB=2.5 and so on, and each course has credit for example COM101=4 and COM312=3 and so on. For these we can calculate the GPA and CGPA for each student.

1.4.1. GPA of Students

As we know after finishing the semester and all the grades of the students, we can prepare the results of the students and also calculate the GPA of the semester. There

is some steps and equations, how calculate the GPA for each semester, these equations are defined as:

$D = Xi * Yi$

Here, i =1 … n, n - number of courses in the semester, Xi - Course credit, Yi - Value of grade

S = sum of D

$GPA = D / Z$

Here Zi - credits of the courses , Z - sum of credit

1.4.2. CGPA of Students

Like the calculation of the GPA for each semester, we can also calculate CGPA for whole courses that were taken in different semesters by student.

In the CGPA calculation there are some steps and equations. These equations are defined as:

$Dj = Xj * Yj$

Here j = 1…N, N - number of courses, Xj - course credit, Yj - the value of grade

F = sum of D

$CGPA = F / Z$

Here Zj - credits of the courses, Z - sum of whole courses credits

CHAPTE 2

STRUCRURE OF THE PROGRAM OF FACULTY'S STUDENTS INFORMATION SYSTEM

2.1. Program Structure:

The program includes the following blocks (figure 2.1)

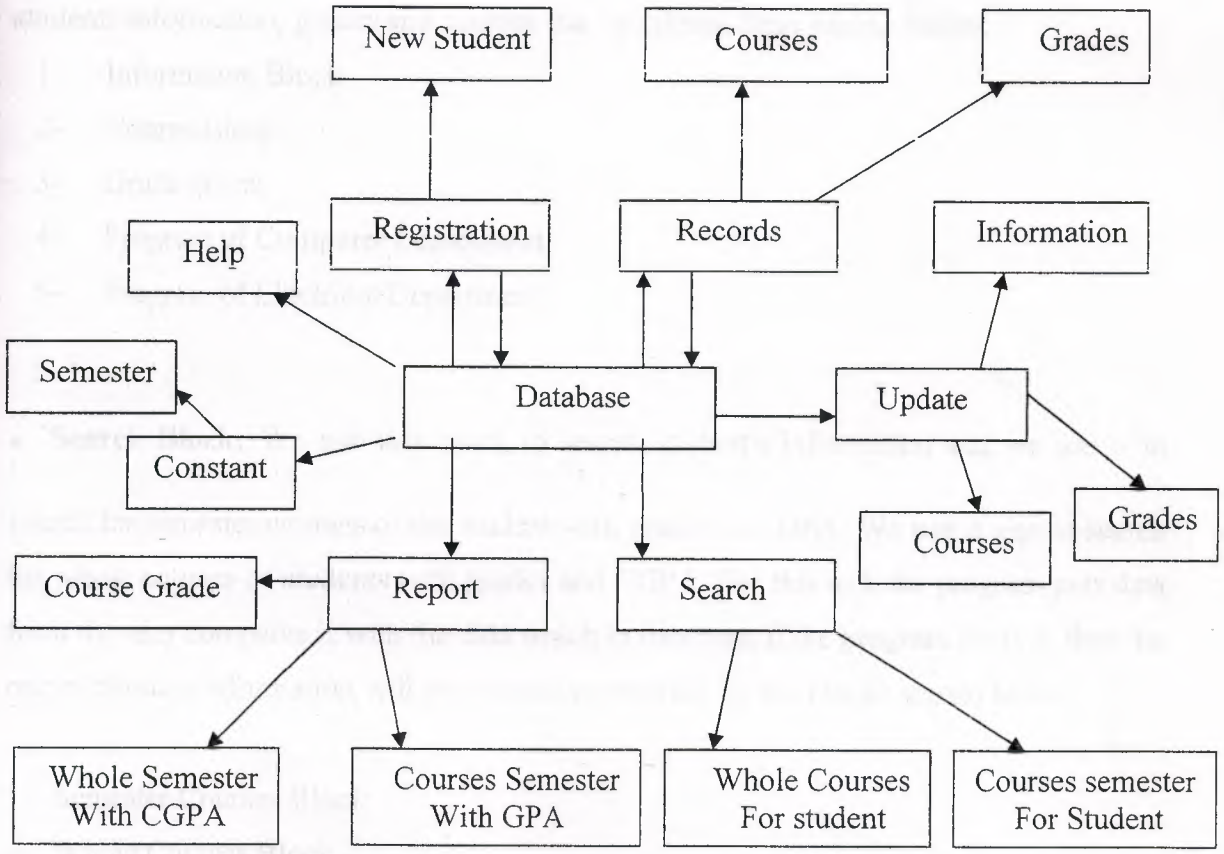

Figure 2.1 Structure of the program

2.1.1. Explanation of the Block Diagram:

- **Registration Block:** In this block we enter the student's information. It is realized by choosing the New Student block.

- **Records Block:** In this block we enter the courses and grades for the student by choosing the modes that are given below:

1- Students Courses Block.
2- Students Grades Block.

- **Update Block:** In this block we shows the student's information , students courses and students grades have been entered and also can be update these information about the students information, grades and courses that by chosen these blocks below:

1- Information Block
2- Course Block
3- Grade Block
4- Program of Computer Department
5- Program of Electrical Department

- **Search Block:** We use this block to search student's information and we use it to search for semester courses of the student with grades and GPA. We use it also to search for whole courses of students with grades and CGPA. For this task the program gets data from the user compares it with the data which in database, if the program finds it, then the rest of the data information will appear on the monitor, by the blocks shown below:

1- Semester Courses Block
2- Whole Courses Block

- **Report Block:** In this block the student's information and students courses with the grades and GPA in the any selected semester is printed, and also we can print the whole student courses with grades and CGPA and courses of students in the selected semester, these done by chosen the blocks shown below:

1- Semester Courses for the student with GPA Block
2- Whole Course of the student with CGPA Block
3- Courses of the students.

•**Constant Block:** In this block we display the studying year and name of semester before used any form or block and it consist one sub block called semester.

• **Help Block:** This block is designed to support the usage of the operators for the whole program. Unless you are going to answer questions of person. A help file is provided for that application.

2.2. Program Flow Chart:

2.2.1. Explanation of add new student Flow-Chart:

Firstly we open the data base and display the information to register the student details, after that we save these details, if the number of the student (A) that entered is equal to the number in the data base then we must change the number of the student until it accepts it, and we know this procedure by showing the message. After that we save these details. If we want to register another student details (K), we can do that as long as the form is still opened, otherwise we close the form. We can see this procedure presented here in the (figure 2.4) below.

2.2.2. Explanation of records courses Flow-Chart:

At the beginning the user should enter the number of student, if the number of the student is in data base then we can register the student courses and save it in the data file, after that we can close the data base or register another student courses (K), we can see it in (figure 2.6) below.

2.2.3. Explanation of records grades of student Flow-Chart:

At the beginning the user should enter the student number, name of semester, and the year, then if these data are in data base the list of students will appear in the form and register the grades of students if not we rewrite the entered data, after that we save these grades and close the data base we can see it in (figure 2.7) below.

2.2.4. Explanation of update student information Flow-Chart:

At the beginning the user should enter the student number, if this number is in data base then the information of student will be appear, after that we can rewrite some information and save it, then we close the data base or enter another student number to update. We can see it in (figure 2.9), and also we can see another update like update students courses that is in (figure2.10), and also update students grades in (figure 2.11) below.

2.2.5. Explanation of calculates GPA of students Flow-Chart:

At the beginning as we know the formula of GPA is the summation of the courses credits of semester and the summation of the result of credits multiply the value of grades of courses like (3*3.5,3*2.5,4*4… and so on) after that we enter the student no, the name of semester, and year if these date are in the file then the result will appear if not we rewrite again correct data, after we close the data base or enter another student data to see the result of students. We can see it in (figure 2.14), and the same procedure will used when we calculate the CGPA of students but we need whole courses of student.

2.2.6. Explanation of the Search Flow Chart:

The required data, which is entered to the program to search, will be checked to whether the data is matched or not according to the database. If the data is not matched then alert message will occurs otherwise the program continue the process that shows the rest information. We can see it in (figure 2.13) below.

2.2.7. Explanation of the GPA Report Flow Chart:

At the beginning the user should enter the number of student, name of semester and year if these data in database then we see the courses of students with grades and GPA if not we enter correct data again, after that the GPA of student display to printed if you want to print press print button if not exit from form and close the data base. These operations we see it in (figure 2.16) and also the same procedures happen in (figure 2.17) for CGPA report flow chart.

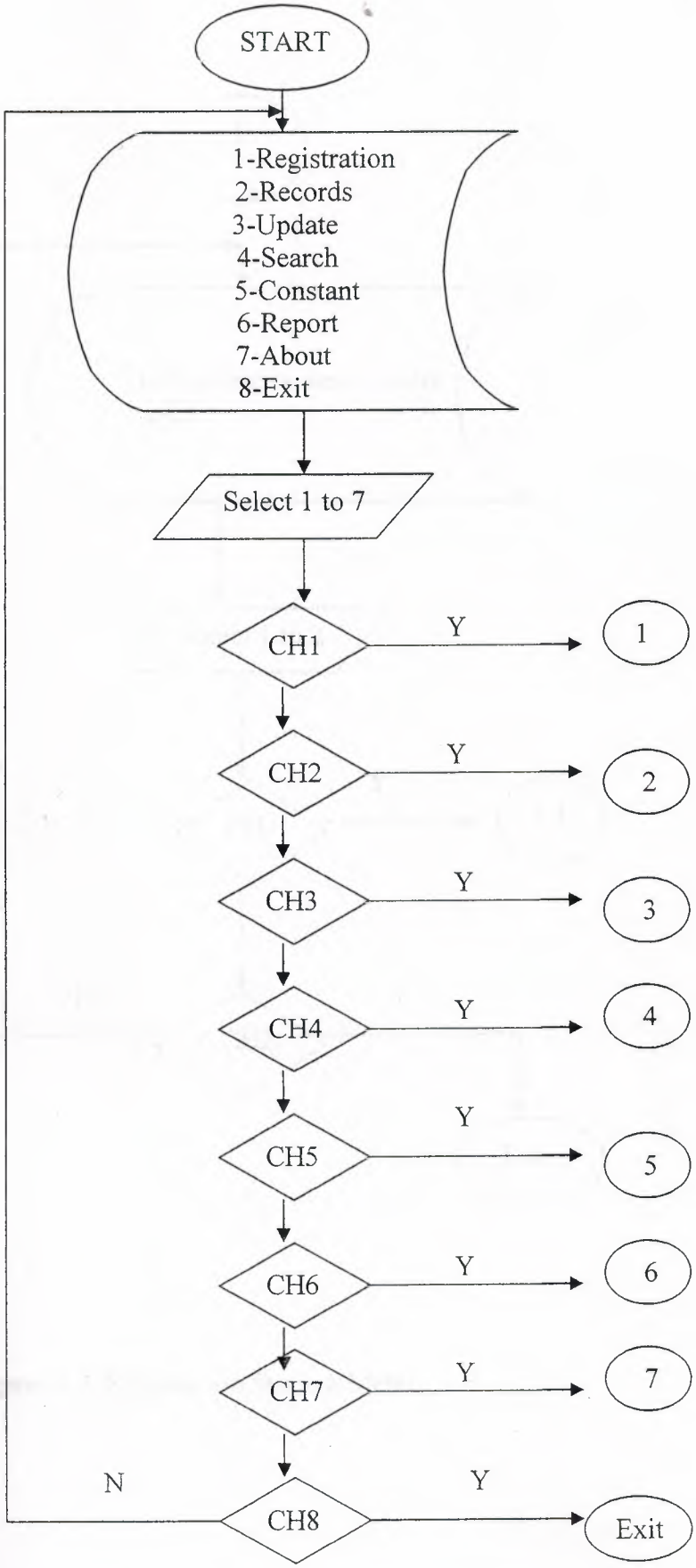

Figure 2.2 Main menu Flow Chart

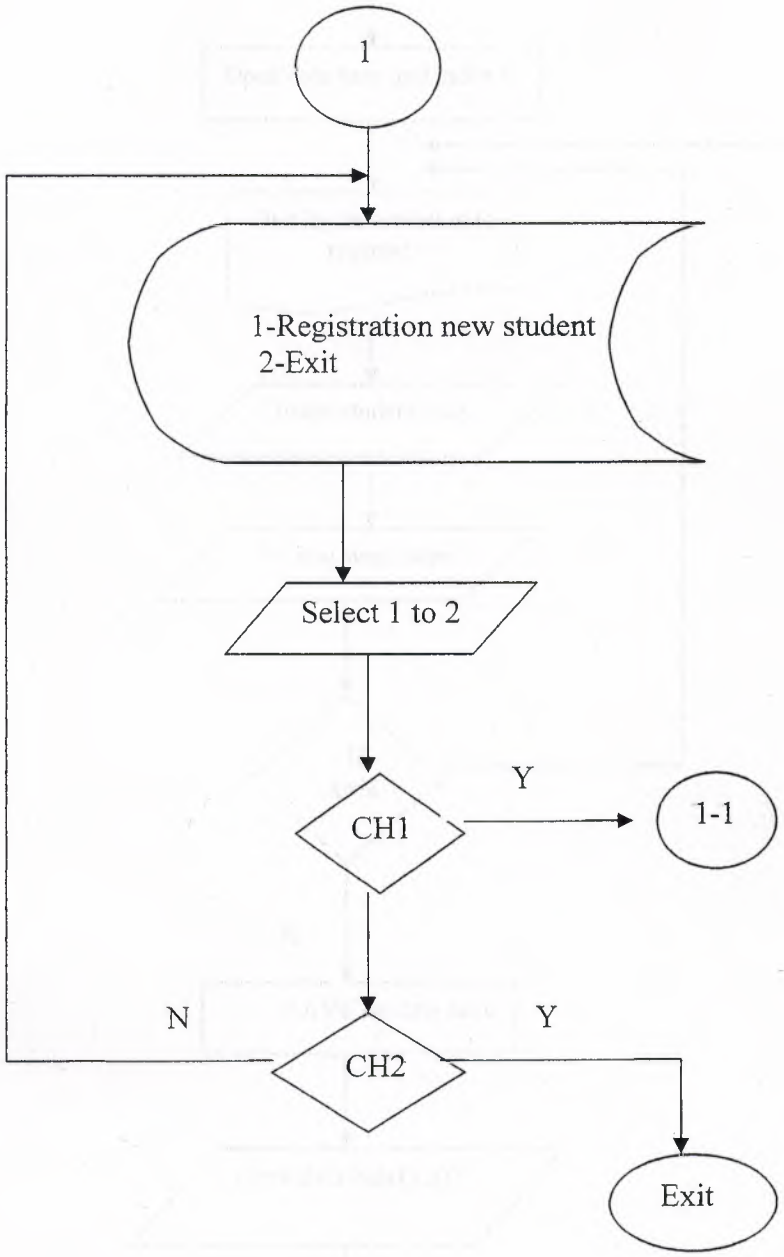

Figure 2.3 Registration Student Menu

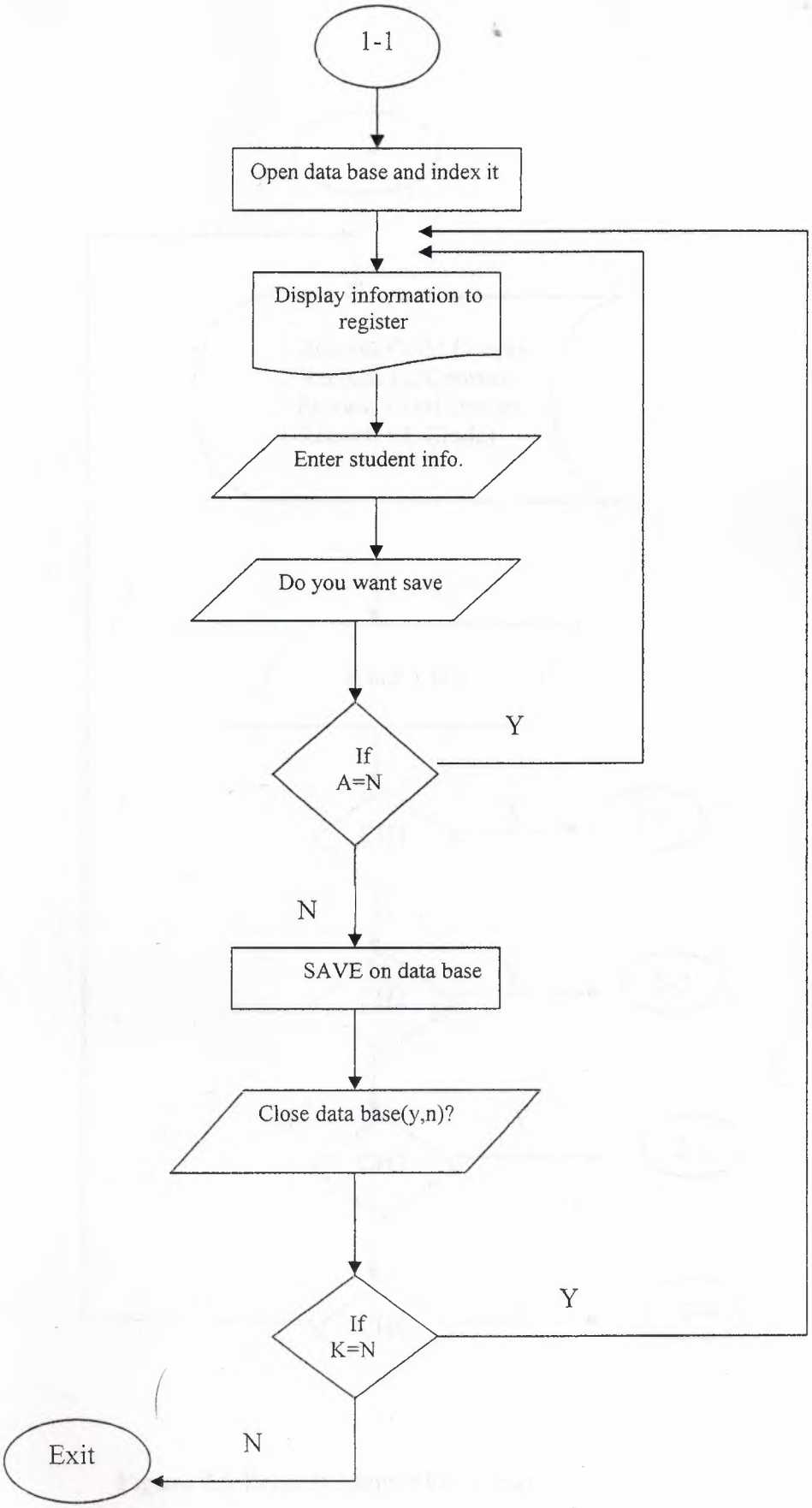

Figure 2.4 Add New Student Flow Chart

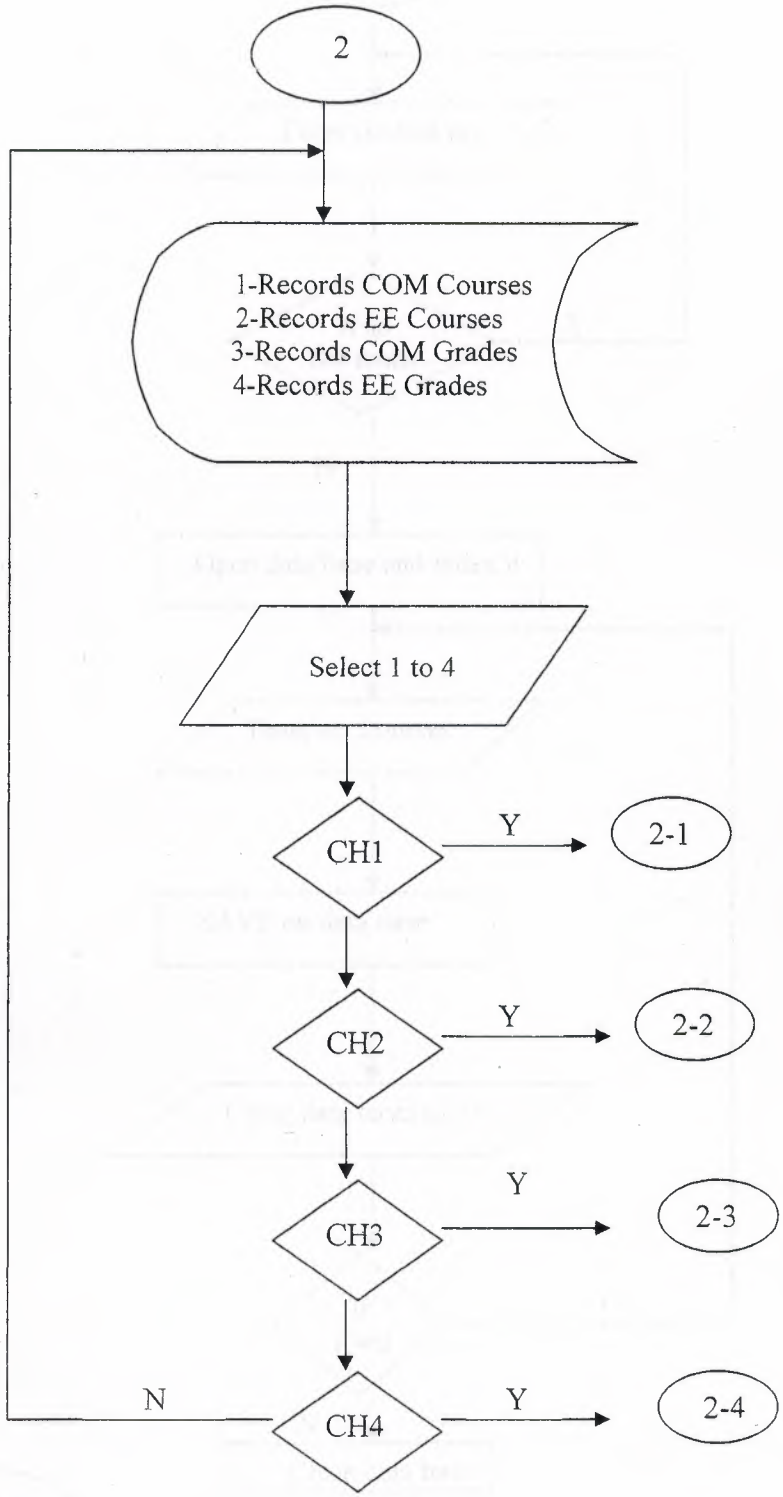

Figure 2.5 Records Menu Flow Chart

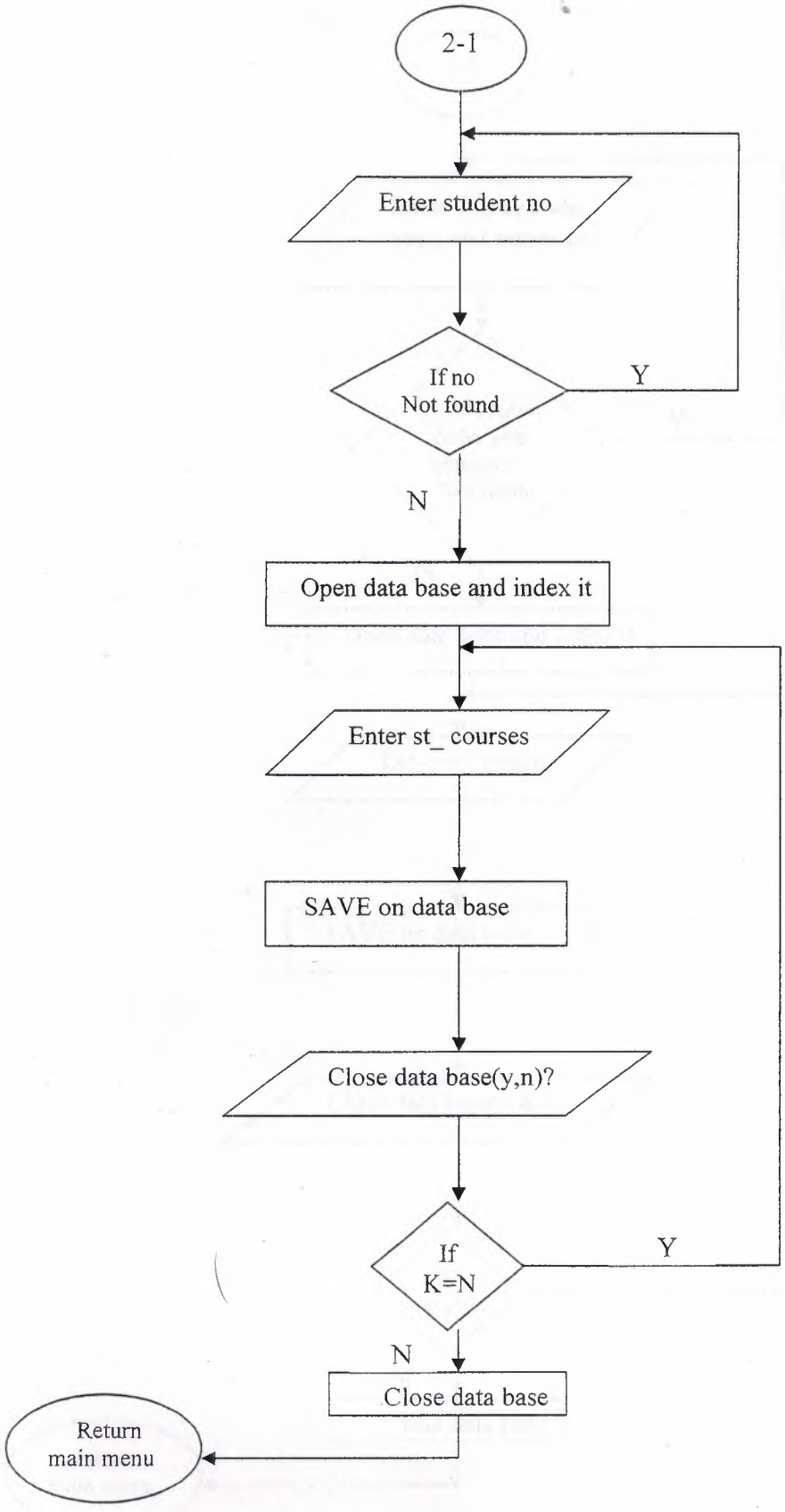

Figure 2.6 Records Courses Flow Chart

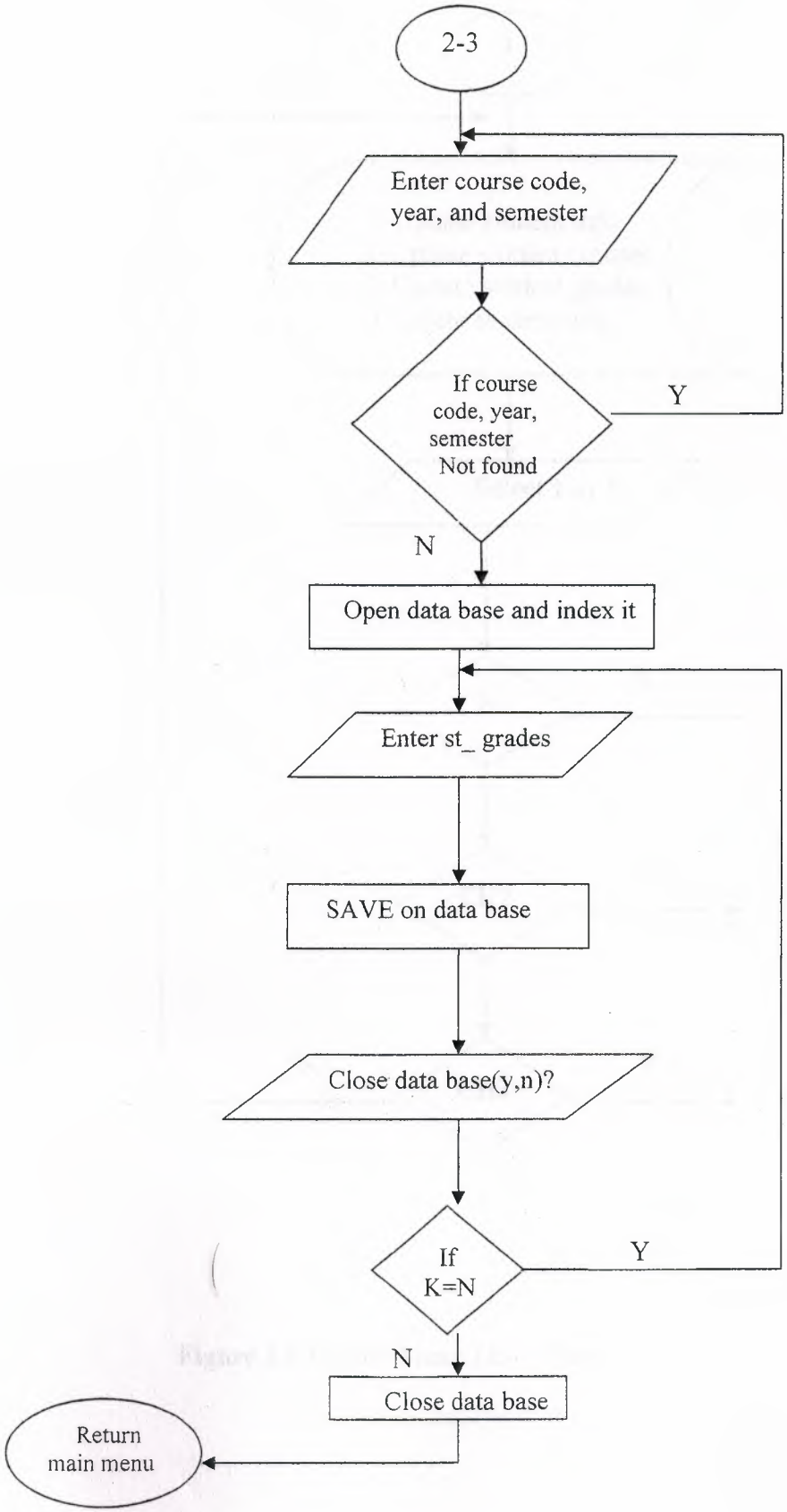

Figure 2.7 Records Grades Flow Chart

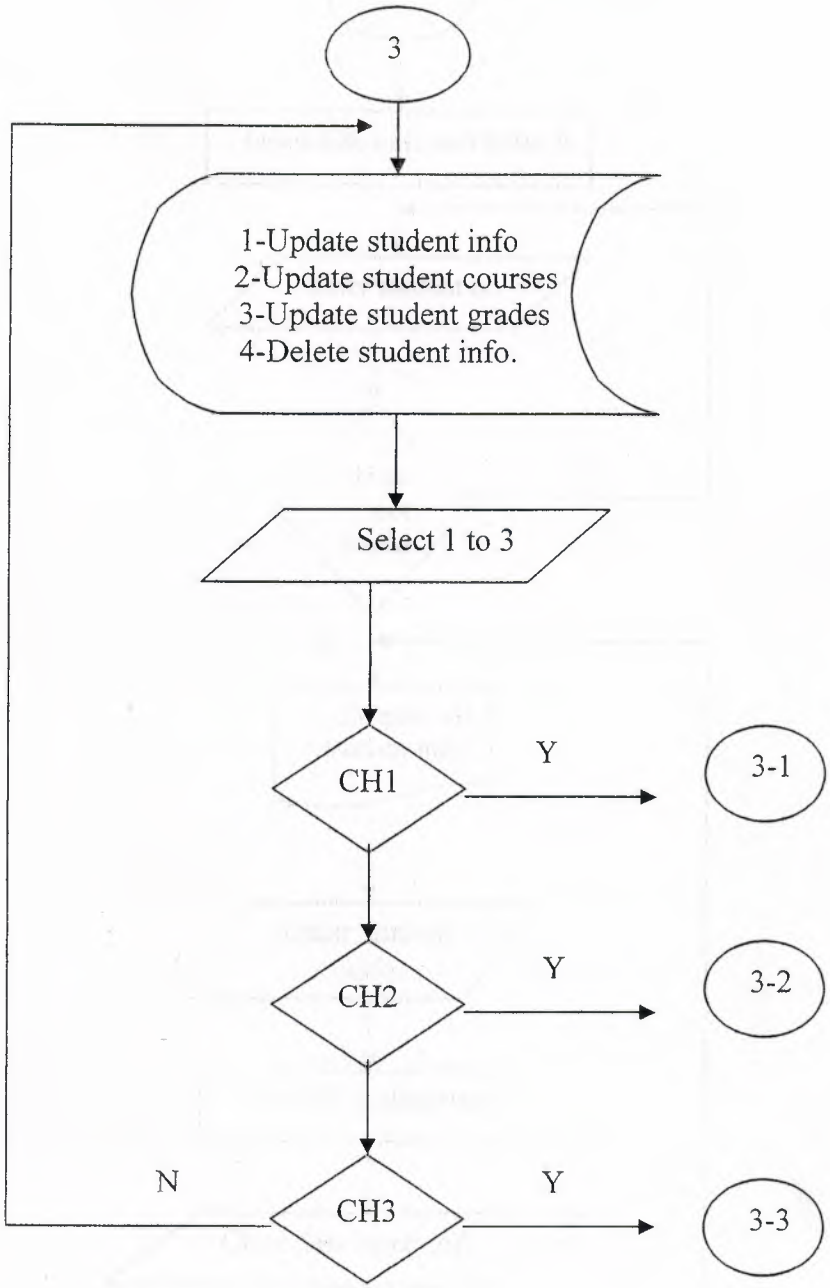

Figure 2.8 Update Menu Flow Chart

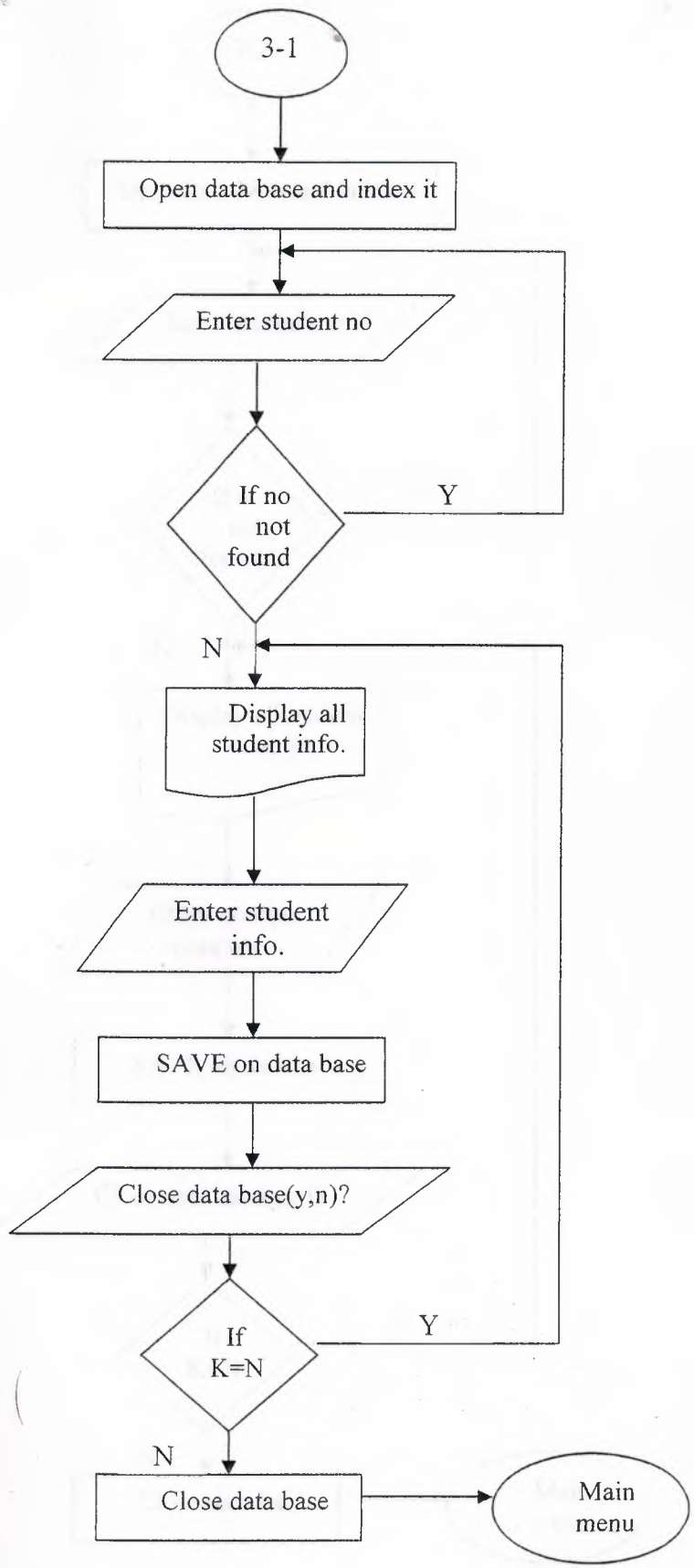

Figure 2.9 Update Student Info.

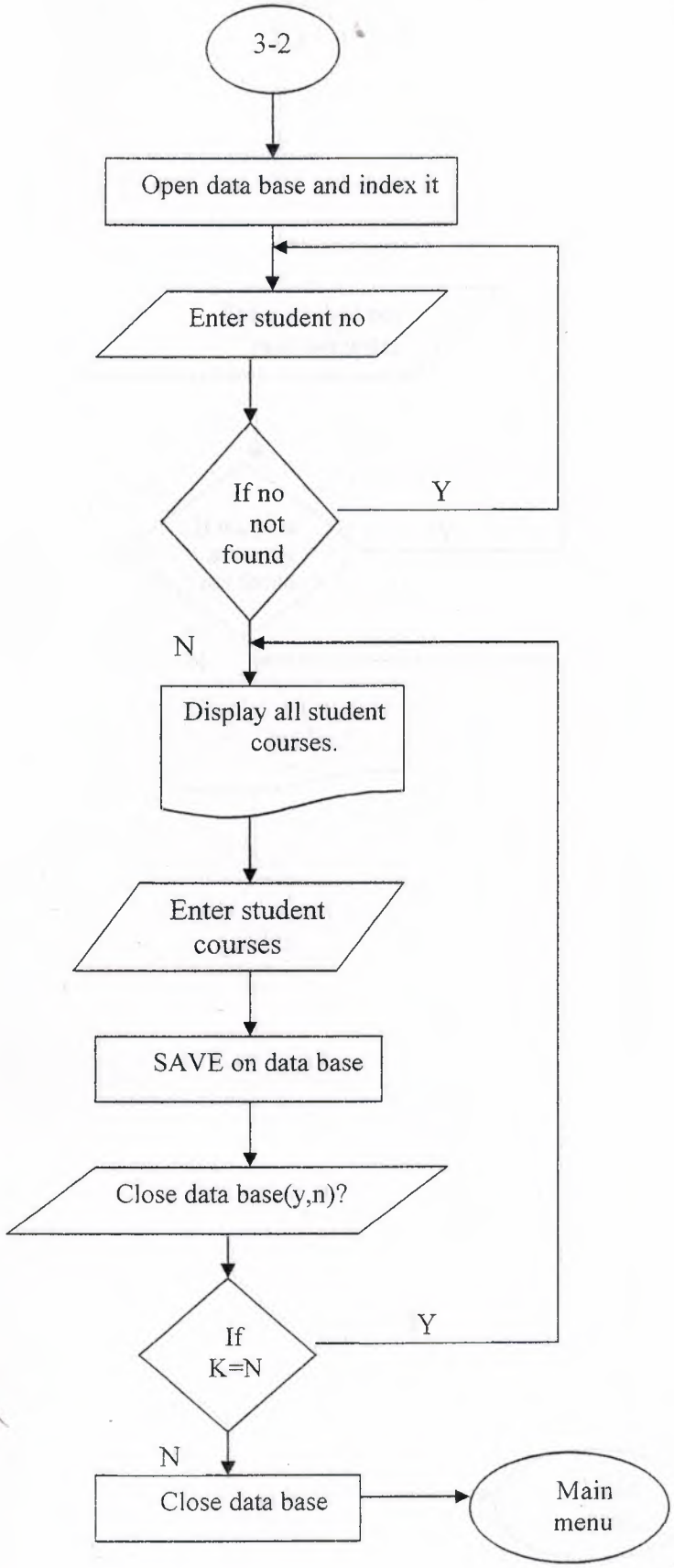

Figure 2.10 Update student Courses

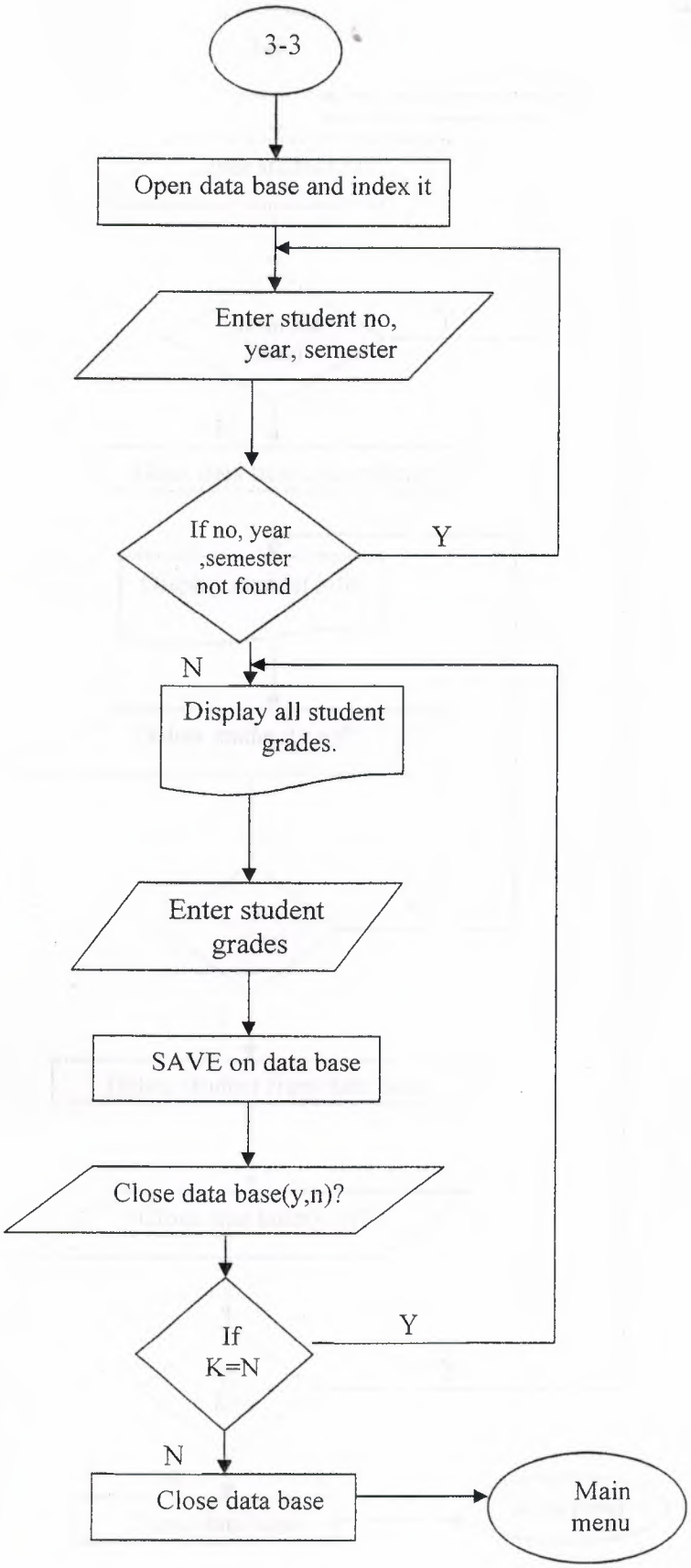

Figure 2.11 Update Student Grades

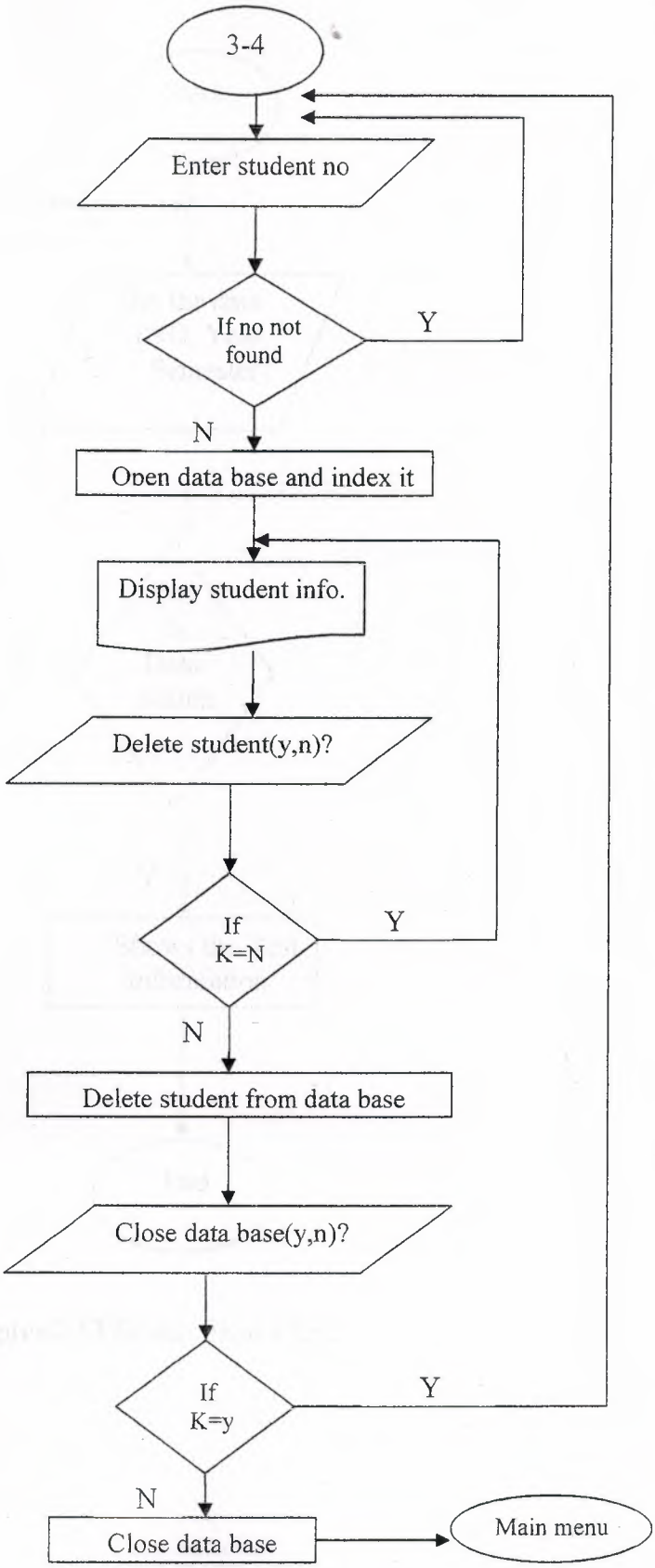

Figure 2.12 Delete Student Info.

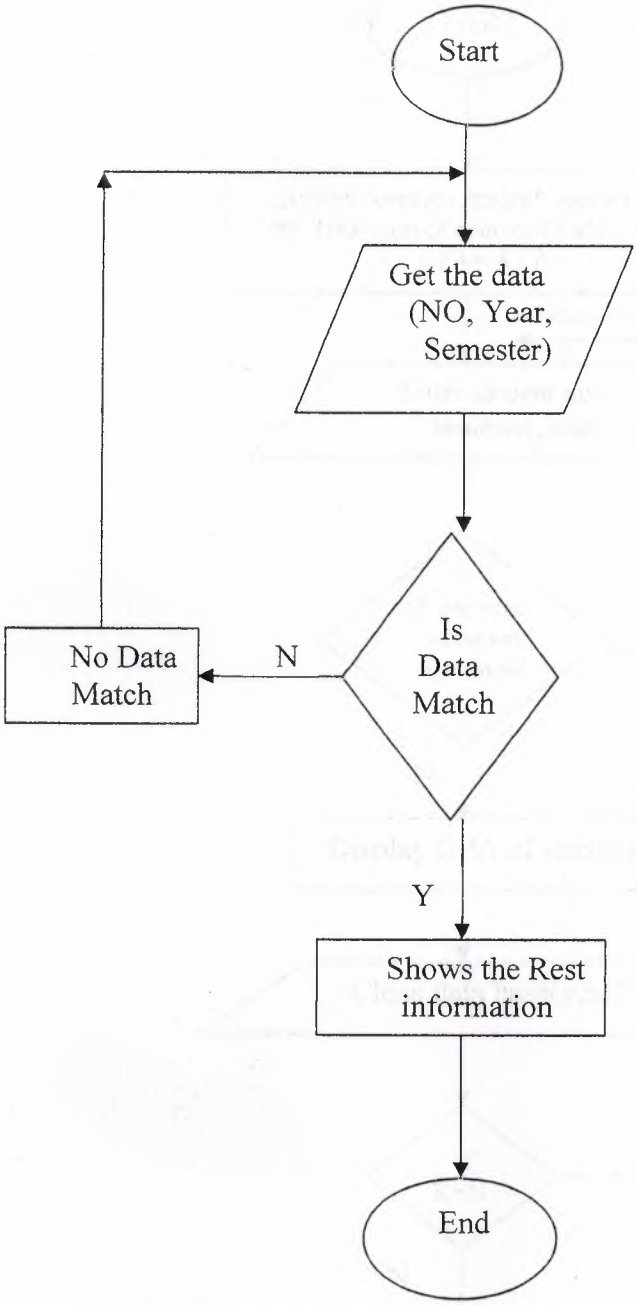

Figure 2.13 Search Flow Chart

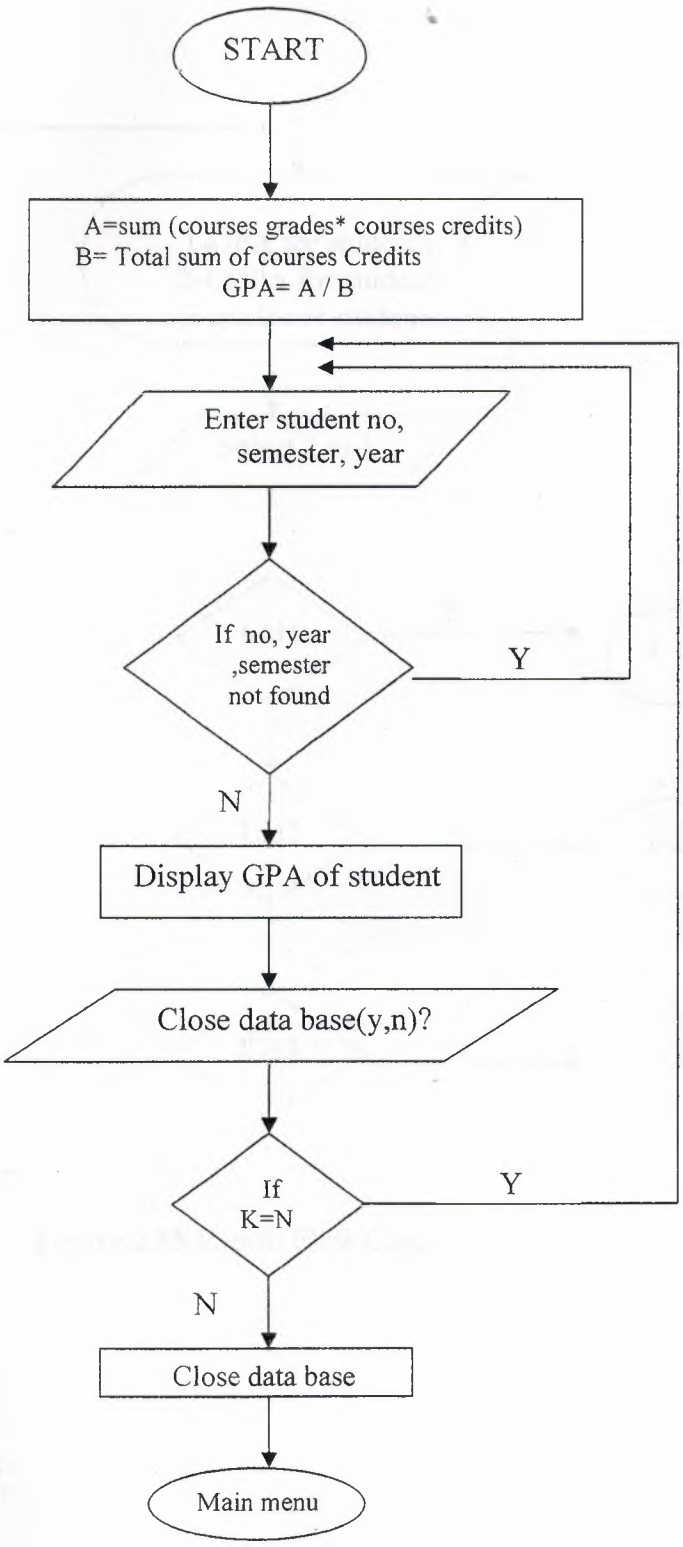

Figure 2.14 GPA Calculation Flow Chart

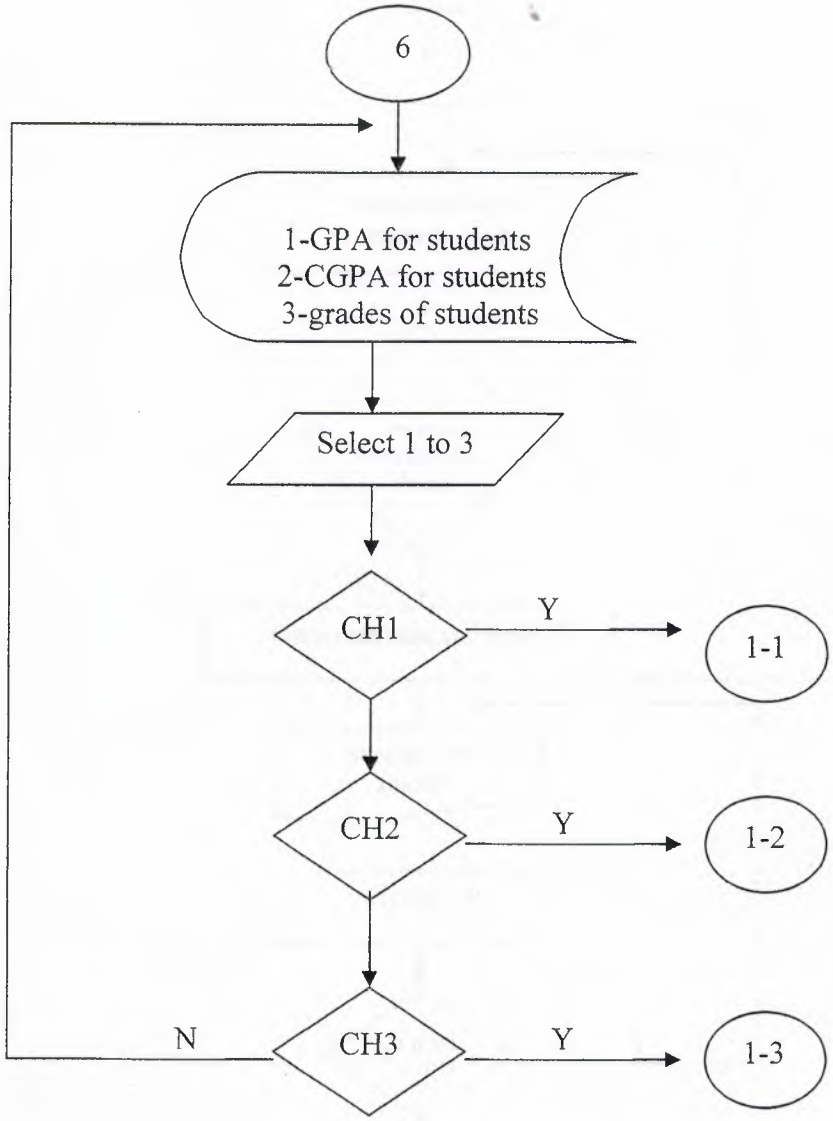

Figure 2.15 Report Flow Chart

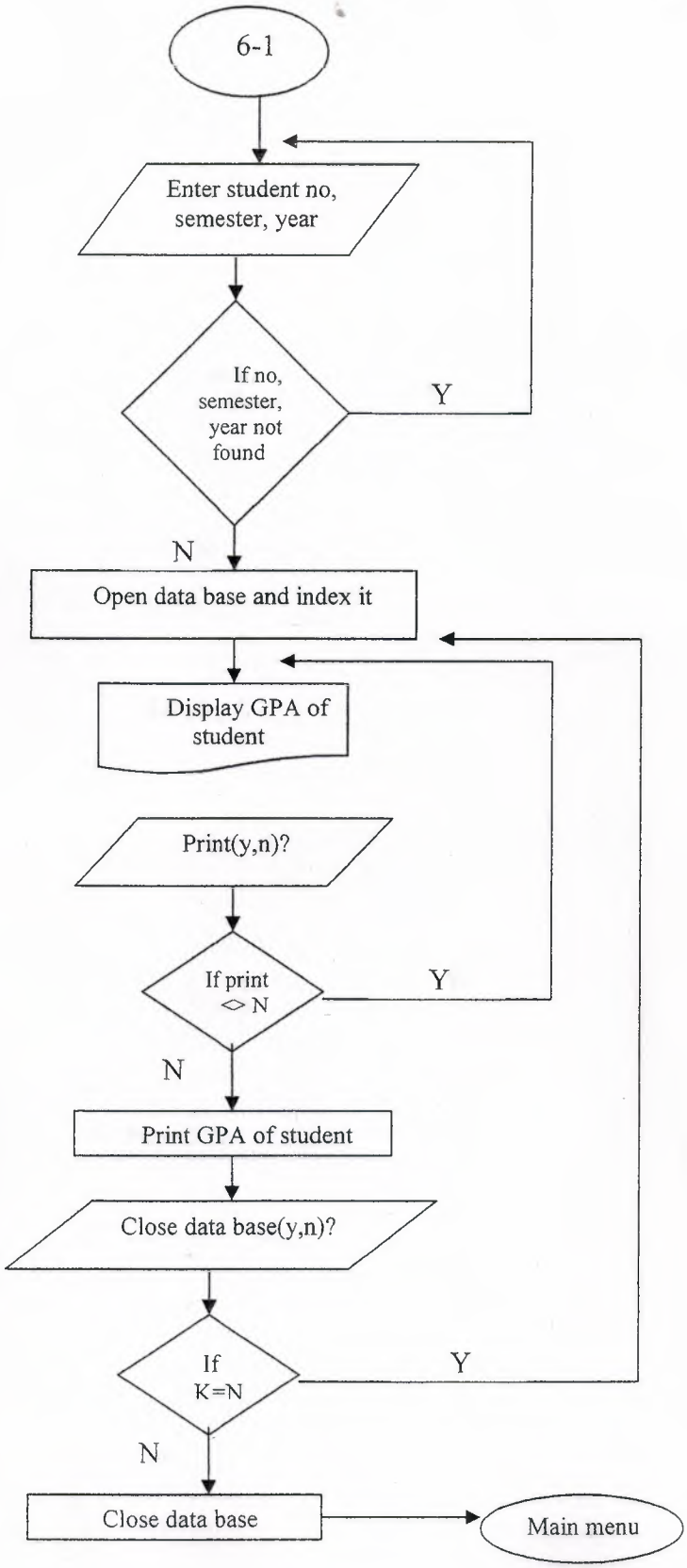

Figure 2.16 GPA Report Flow Chart

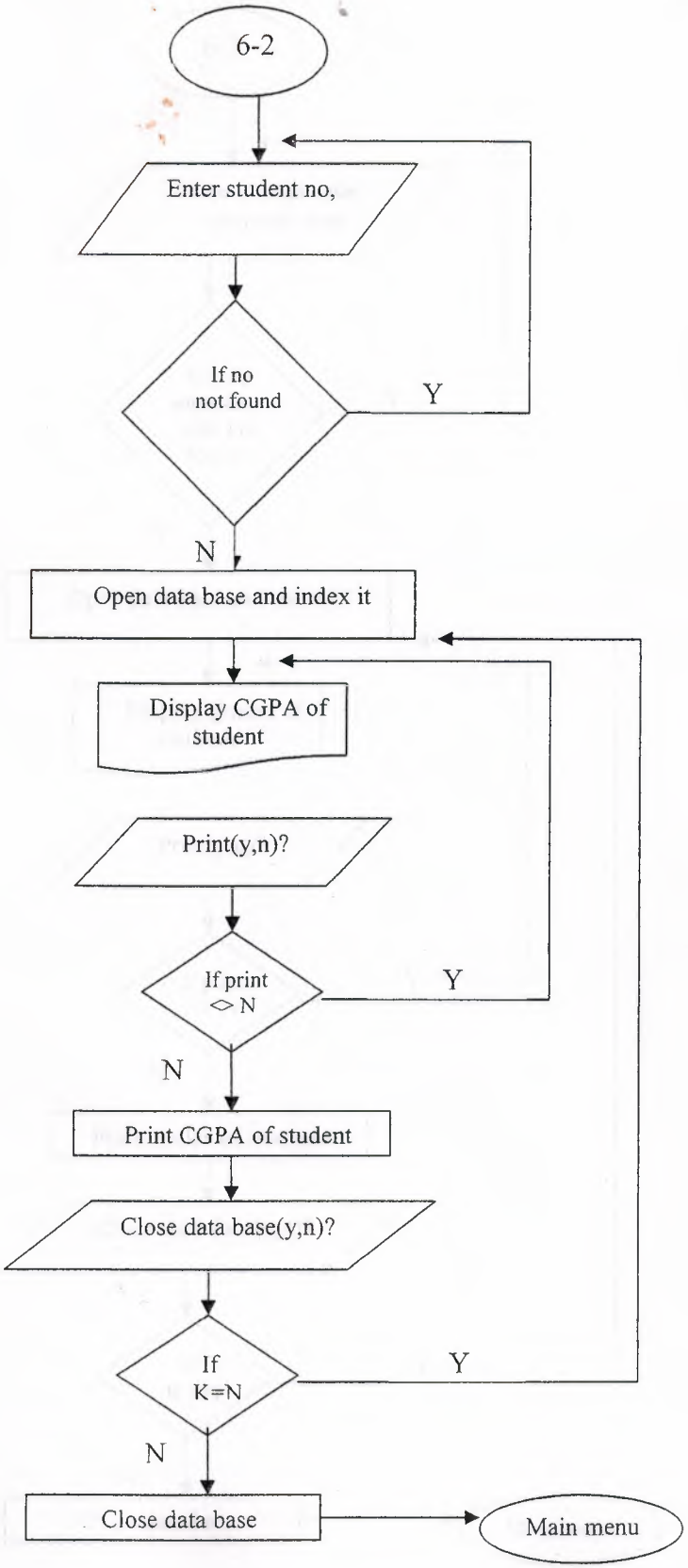

Figure 2.17 CGPA Report Flow Chart

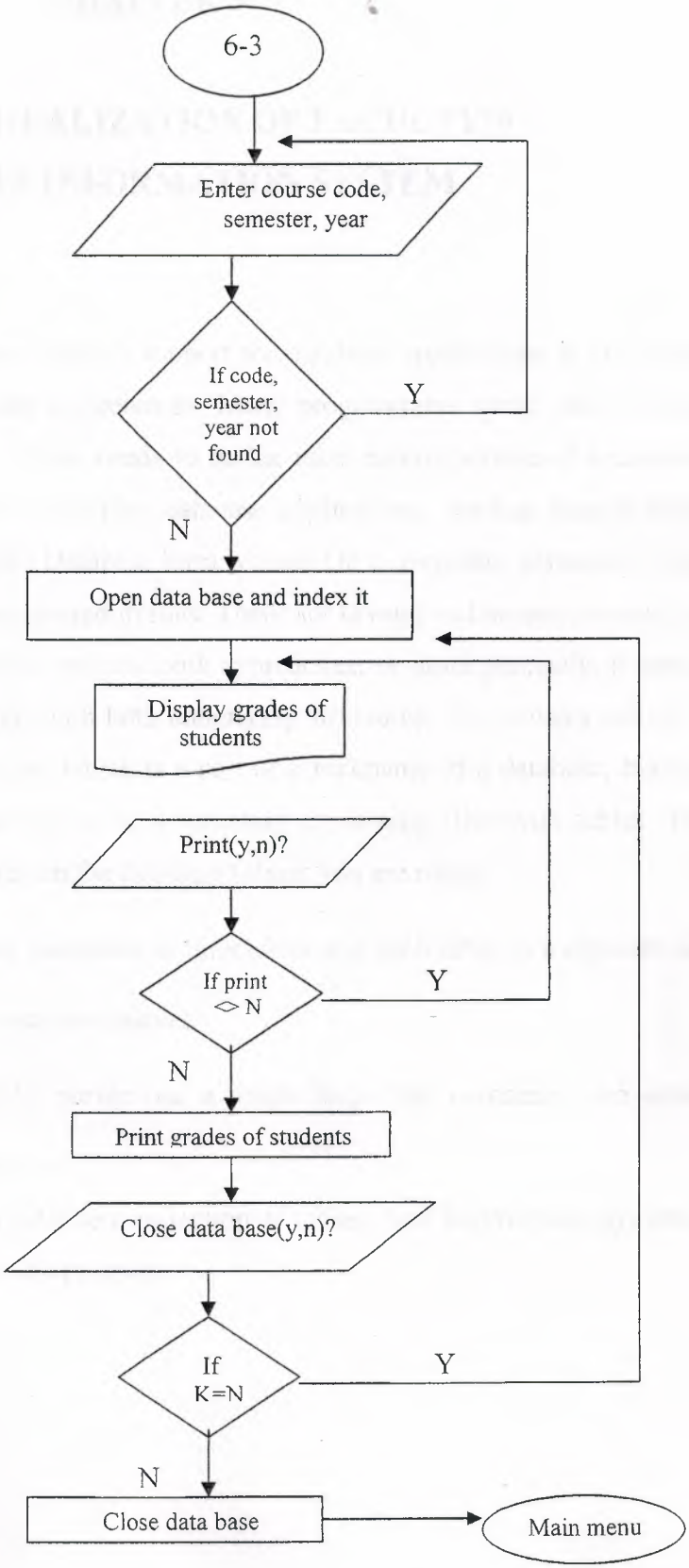

Figure 2.18 Grades Report Flow Chart

CHAPTER 3

COMPUTER REALIZATION OF FACULTY'S STUDENTS INFORMATION SYSTEM

3.1. Database Structure:

First thing as we know Delphi's support for database applications is one of the key feathers of the programming environment. Many programmers spend most of their time writing data-access code, which needs to be the most robust portion of a database application. You can create very complex database applications, starting from a blank form or one generated by Delphi's Database form wizard. On a computer, permanent data-including database data is always stored in files. There are several techniques you can use to accomplish this storage. Delphi can use both approaches; or more precisely, it uses a custom approach that works well with both underlying structures. You always refer to a database with its name or an alias, which is a sort of a nickname of a database, but this reference can be to a database file or to a directory containing files with tables. The approach used by Delphi depends on the database format you are using:

- Paradox and dBASE tables define databases as directories and each table as a separate file or actually multiple files if you include indexes.
- Access, interBase, and most SQL server use a single huge file containing the entire database, with all tables and indexes.

In general we use term database to refer to a collection of tables. And bellow there are some database tables, which I used in this program:

Field Name	Data Type	Field Size
Student No	Number	8
Student Name	Alpha	20
Student Surname	Alpha	20
Birthday	Date	Short Date
Birthplace	Alpha	20
Date	Date/Time	Short Date
Department	Alpha	20
Country	Alpha	20
Transfer	Alpha	25

Table 3.1 Students Information

Field Name	Data Type	Field Size
Student No.	Number	8
Student Name	Alpha	20
Student Surname	Alpha	20
Course Code	Alpha	10
Course Name	Alpha	35
Grades	Alpha	2
Academic Year	Alpha	4
Semester	Alpha	10
Credit	Number	2
Credit * grade	Number	4
Gr	Number	4

Table 3.2 Students Courses

Field Name	Data Type	Filed Size
Course Code	Alpha	8
Course Name	Alpha	35
Credit	Number	2

Table 3.3 Undergraduate Curriculum

Field Name	Data Type	Filed Size
Semester	Alpha	10
Year	Alpha	4

Table 3.4 Semesters

Field Name	Data Type	Filed Size
Pass_no	Number	6

Table 3.5 Password

3.2. Define Relationships Between Tables:

When we create a relationship, the related fields don't have the same names. However, related fields must have the same data type unless the primary key field is an AutoNumber field. We can match an AutoNumber field with a number field only if the fieldsize property of both of the matching fields is the same. For examples, we can match an AutoNumber field and a field number field if the fieldSize property of both fields is Long Integer. Even when both matching fields are Number fields, they must have the same fieldSize property setting. We can see bellow the relationships between the tables of this project:

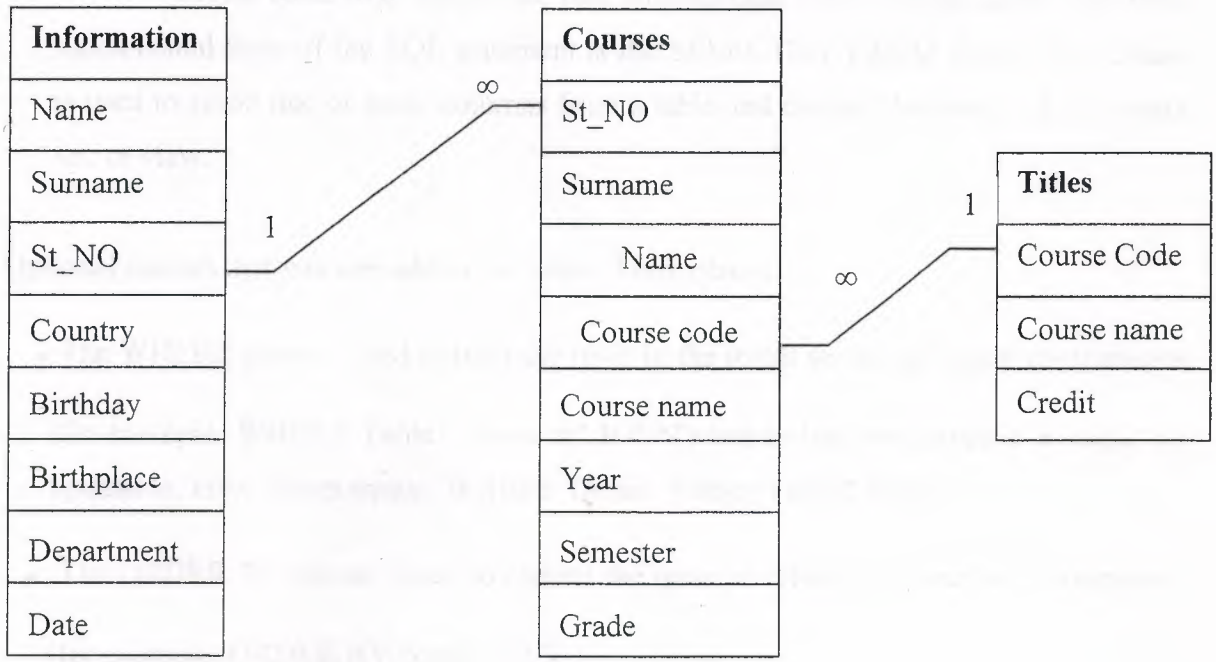

Figure 3.1 Relationships

3.3. Delphi Database Components

Delphi includes a number of components related to database. The data access page pf the component Palette contains components used to interact to databases. Most of them are nonvisual components, since they encapsulate database connection, tables, queries, and similar elements. Fortunately, Delphi also provides a number of predefined components you can used to view and edit database data. In the data control page, there are visual components used to view and edit the data in a form. These controls are called data-aware controls. To access a database in Delphi, you generally need a data source, identified by the data source component. The data source component, however, does not indicate the data directly; it refers to a data set component. This can be a table, the result of a query, the result of stored procedure, the data fetched from a remote server, or some other custom data set. As soon as you have placed a table or query component on the form, you can use the dataset property of the data source component to refer to it. For this property, the object inspector lists available data sets of the current form, or of other forms connected with the current one(using the File> Used Form command).

3.4. Working with SQL

Create basic SQL statements that selected data from existing tables. The most fundamental form of the SQL statement is the SELRCTED_FROM clause. This clause is used to select one or more columns from a table and display the results of in a result set, or view.

Optional clauses that you can add to the select_Form clause:

- The WHERE clause: Used to limit the rows in the result set using logical comparisons (for example, WHERE Table1. Name = ' JOUN') and to link two tables in a single, no updatable, view (for example, WHERE Table1. Name=Table2.Name).
- The ORDER BY clause: Used to control the order in which the result set is displayed (for example, ORDER BY Name ASC)

- The GROUP BY clause: Used to create a subtotal result set based on a break column (for example, GROUP BY Name).

- The HAVING clause: Used only with the GROUP BY clause, the HAVING clause acts as a WHERE clause for the GROUP BY subtotal clause (for example, GROUP BY Name HAVING SUM(Grades Total) > 65).

- The INNER JOIN clause: Used to join two tables together into a single, updatable result set. The INNER JOIN results rows that have a corresponding match in both tables.

- The LEFT JION and RIGHT JOIN: Used to join two tables into a single, updatable set. The LEFT JOIN includes all records from the first (Left hand) table and all rows from the second table that have a corresponding match the RIGHT JOIN works in reveres.

- The UNION clause: Used to combine two or more complete SQL queries into a single result set(for example, SELECT * FROM Table1 UNION SELECT * FROM Table2).

- The TRANSFORM_PIVOT clause: Used to create across- tab query as a result set (for example, TRANSFORM SUM (Credit Value) FROM Credit Table GROUP BY Grades PIVOT Credit).

Additional SQL keywords that you can use to control the contents of the result sel:

- BETWEEN _ AND

- DISTINCT and DISTINICTROW

- AS

- TOP n and TOP n PERCENT

- AVG, COUNT, MAX, MIN, and SUM

3.5. Building Database Application:

Delphi database applications do net have direct access to the data sources that they reference. Delphi interfaces with the Borland Database Engine (BDE), which does have direct access to a number of data sources, including dBASE , Paradox , ASCII , FoxPro, and Access tables. The BDE can also interface with Borland's SQL links, a tool that allows access to a number of local and remote SQL servers. The fact Delphi applications generally don't access data directly but use the BDE basically means that you will need to install the BDE along with your applications on your clients' computers. This is not difficult, since Delphi includes the "Lite" version of an installation program (Install Shield) that can be used to prepare installation disks for the BDE, along with your own application. The BDE files are required your Delphi database applications won't work without them but you can distribute them freely. Delphi now includes a ClientDataset component you can use to access data from an OLE server running on a different computer.

3.6. Handling Database Error:

Another important element of database programming is handling database error in custom ways. Of course, you can let Delphi show an exception message each time a database error occurs, but you might want to try to connect the errors or simply show more details. There are basically three approaches you can used to handle database errors:

- You can wrap a try-except block around risky database operations, such as a call to the open method of a query or the post method of a data set. This is not possible when the operation is generated by the interaction a data aware control.
- You can install a handler for the OnExeception event of the global application object.
- You can handle specific events of the data sets related to errors, as OnPostError, OnEditError, OnDeleteError, and OnUpdateError.

While most of the exception classes in Delphi simply deliver an error message with database exceptions you see a list of error, showing local BDE error codes and also the native error codes of the SQL server you are connected with.

Besides this error-handling code, the program has a table and a query, along with the error related event handlers. As already mentioned you can install an event handler

related to specific errors of a dataset. The three events OnPostError, OnDeleteError, and OnEditError have the same structure. Their handlers receive as parameters the dataset, the error itself, and an action you can request from the system.

3.7. Update Databases with SQL:

To add, delete, and edit data within tables using DML (Data Manipulation Language) SQL keywords by using DML statements you can quickly create test data for tables and load default values into startup tables. DML statements-such as Append queries, Make Table queries, and Delete queries can outperform equivalent Delphi code versions of the same operations.

Managing data within the tables using the following DML keywords:

- The INSERT INTO statement can be used to add new rows to the table using the VALUES clause.
- You can create an Append query by using the INSERT INTO_FROM syntax to copy data from one table to another. You can also copy data from one database to another using the IN clause on an INSERT INTO_FROM statement.
- You can create new tables by coping the structure and some of the data using the SELECT_INTO statement. This statement can incorporate WHERE, ORDER BY, GROUP BY, and HAVING clauses to limit the scope of the data used to populate the new table you create.
- You can use DELETE FROM clause to remove one or more records from an existing table. You can even create customized views of the database using the JOIN clause, and remove only records that are the result of a JOIN statement.

3.8. Layout of the Application:

3.8.1. Main menu screen:

It consists of seven Buttons, each button has a specific mission, and these missions will be explaining as follow:

1- Registration Button: we can use this button to transfer to another form we wanted, and it has two sub buttons one for new student and another for Exit button.

2- Records Button: this button has four sub buttons these are records COM courses, records EE courses, grades for COM students, and grades for EE students, and we can chose of these sub buttons to do want we want.

3- Update Button: this button used when we want to update or rewrite some information or grades or courses of students, and it has three sub buttons, update information, update grades, and update courses of students.

4- Searching Button: the usage of this button is to find a certain data that we entered for the student before and it has four sub buttons: the first one is search for information of students, second search and calculate the CGPA of students, third search and calculate GPA of COM students, and last one is to search and calculate the GPA of the EE students.

5- Constant Button: we use this button to chose the year and name of semester.

6- Report Button: we use this button to print out the student data like GPA of students, CGPA of students, the grades of students, and also some data related for the students. Report button has six sub buttons.

7- Help Button: this button used to give information about my project.

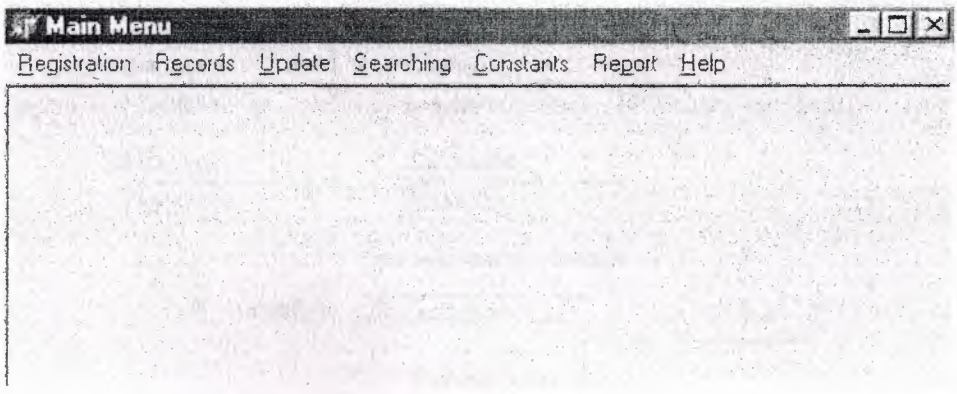

Figure 3.2 Main Menu Screen

3.8.2. Students Information Screen:

This screen allow us to enter information such as: Student Number, date, student name, student surname, student birthday, student birthplace,

department of student, country of student, The name of university if he or she transfer, and there are tow tasks buttons and radio buttons. Their functions are shown bellow:

1- Save Button: This button used to keep and save the information of students which entered in the DBedits components to a file.
2- Exempted Courses Button: when this button is pressed the form that used to transfer student courses and also the student number and year are appeared and I make the EX grade in the DBedit component as nonanable.
3- DBradio Button: this used to select data; here we use it if the student is transferred from any university. When we select the DBedit the form will be appeared to write the name of university for the student transfer.

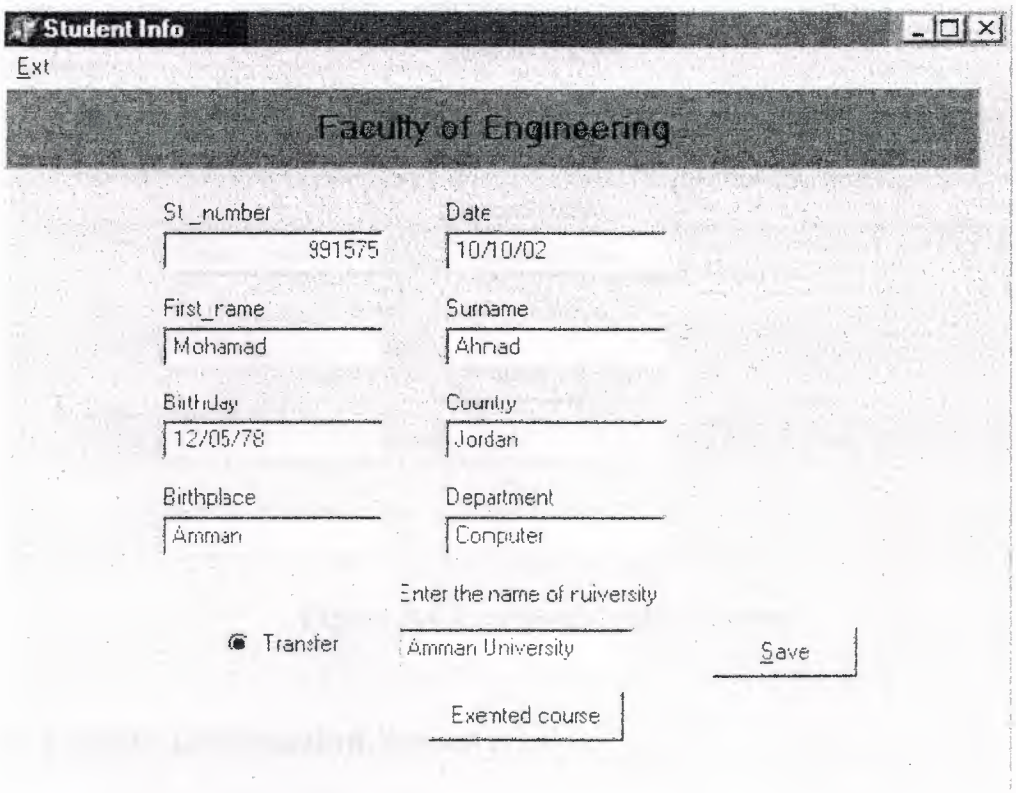

Figure 3.3 Student info. Screen

3.8.3. Student Transfer screen:

This form especially for the transfer students, it consists:

1- Dblookupcombobox Component: this component used to get data from table and set data to another table and in this component there are whole courses of electrical department and computer department separately.
2- Save Button: the usage of this button is to keep or save the data which entered to the file.
3- Back Button: since we click this button it will be returned to the student information screen.
4- Five DBedit components: these components are used to insert the data on it.

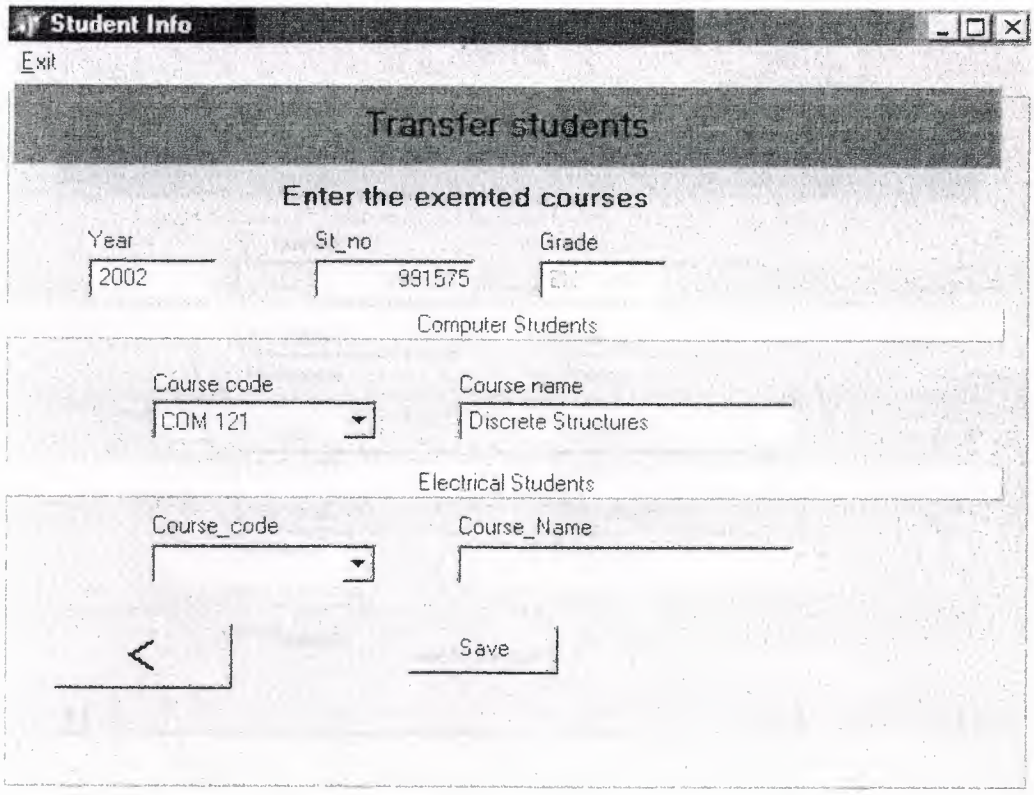

Figure 3.4 Exempted Courses Screen

3.8.4. Update Information Screen:

We use this form when we want to update or correct some of student information. It consists four buttons and nine DBedit components and radio button.

1- Search Button: This button used to get the student information, which is saved in the file, by using student number. I use filter property for searching.
2- Save Button: this button for saving or keeping the update information of students in the file.

3- Delete Button: this button for erase or delete information and when we click this button there is a message appears about ask you do you want to delete or not if yes click ok if not click no.

4- Exit Button: this button has one task which is signing out of the form.

5- DBRadio button: this used to select data.

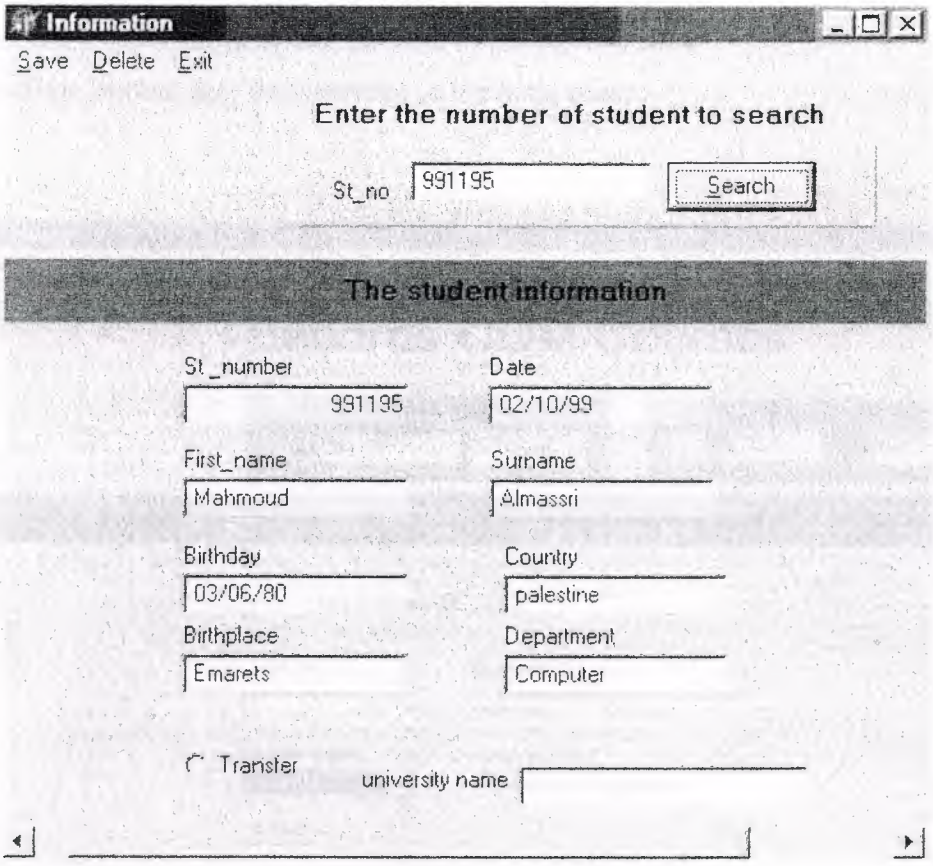

Figure 3.5 Update Student Info. Screen

3.8.5. Registration Courses for Computer students Screen:

This screen allows us to enter courses of computer department. It consists:

1- Search Button: the usage of this button is to be sure the student register or not if he/she registers then his/here name will be appear in the DBedits as we show bellow and in this button we search by using filter property and used number of student.

2- Small Button: this button used for returning to the form, which includes the name of semester and year to change the year and semester if we want.

3- Dblookupcombobox component: this component is to get the data from table and sent or set the data to another table and this component consists the whole courses of the computer department.

4- Save Button: this button used to keep or save the data, which entered in to the file.

5- Six DBedit components: these are used to enter data.

6- Exit Button: this for returning to the main menu.

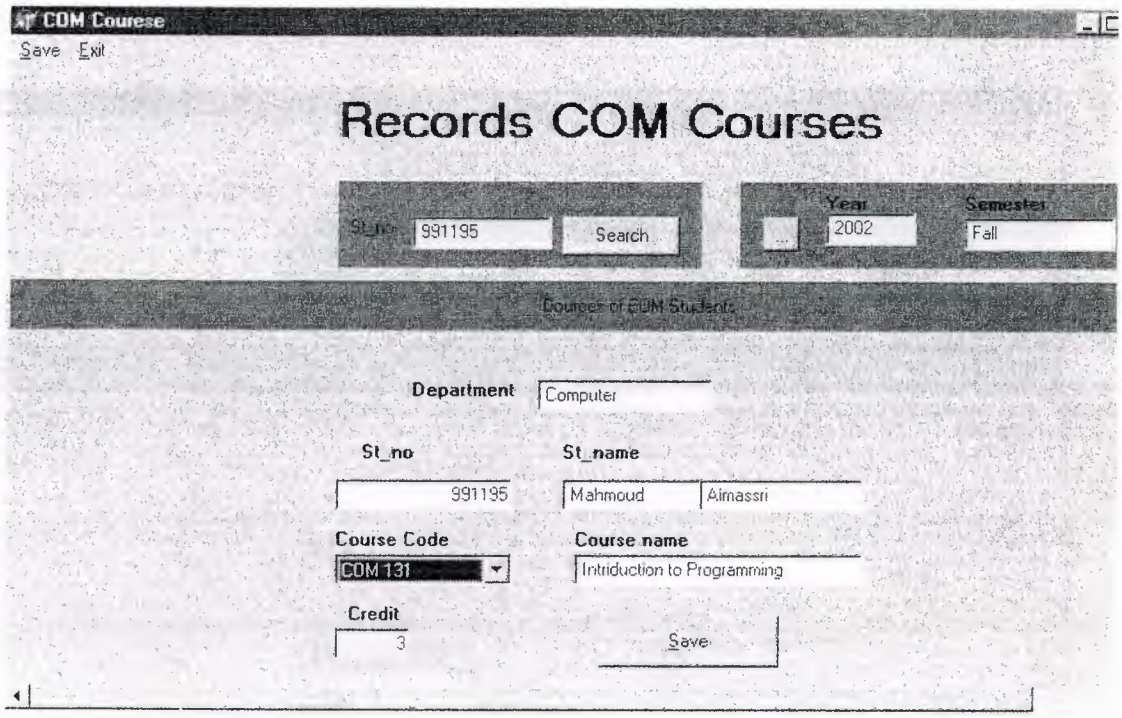

Figure 3.6 Records COM Courses Screen

3.8.6. Registration Courses for Electrical and Electronic students Screen:

This screen allows us to enter courses of Electrical and electronic department. It consists of:

1-Search Button: the usage of this button is to be sure the student is registered or not if he/she is registered then his/here name will be appear in the DBedits as we show bellow by using student number.

2- Small Button: this button is used to return to the form which includes the name of semester and year to change the year and semester if we want

3- Dblookupcombobox component: this component is to get the data from table and sent or set the data to another table and this component consists the whole courses of the electrical and electronic department.

4- Save Button: this button used to keep or save the data, which is entered to the file.

5- Six DBedit components: these are used to enter data.

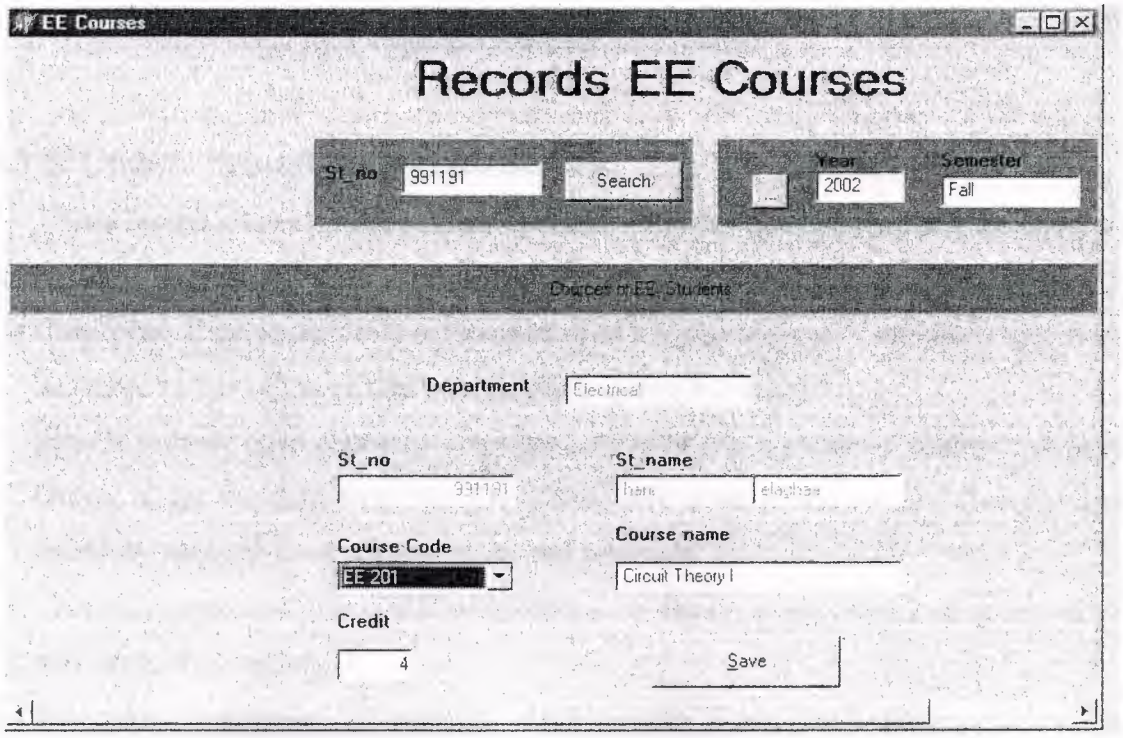

Figure 3.7 Records EE Courses Screen

3.8.7. Searching Information Screen:

It consists two main components:

1- Search Button: this button is used to get the information student by entering the number of student in the edit component as you see bellow, if the number of student in the file which keep it then the information will be appeared.

2- DBgrid component: this component capable to display a whole table at once and you can edit the grid's contents.

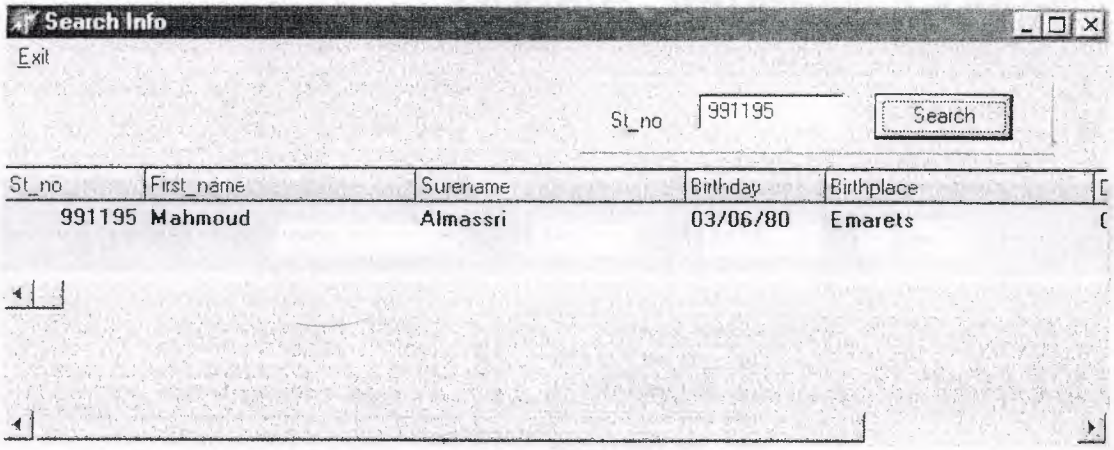

Figure 3.8 Search info. Screen

3.8.8. Update Semester Courses Screen:

This screen allows us the student courses in the any semester we will be chosen and it consists of:

1- Combobox Component: this component used to select a single value from specified set. It has three values of the semester name (fall, spring, summer).

2- Search Button: after choosing semester and enter the number of student, we press this button to get the data from the file which save it on it and we use Query property to search by using student number, year, and semester.

3- DBgrid component: this component capable to display a whole table at once and you can edit the grid's contents.

4- Navigator Component: it consists eight buttons First, Last, Previous Next, Delete, Refresh, Edit, and Post.

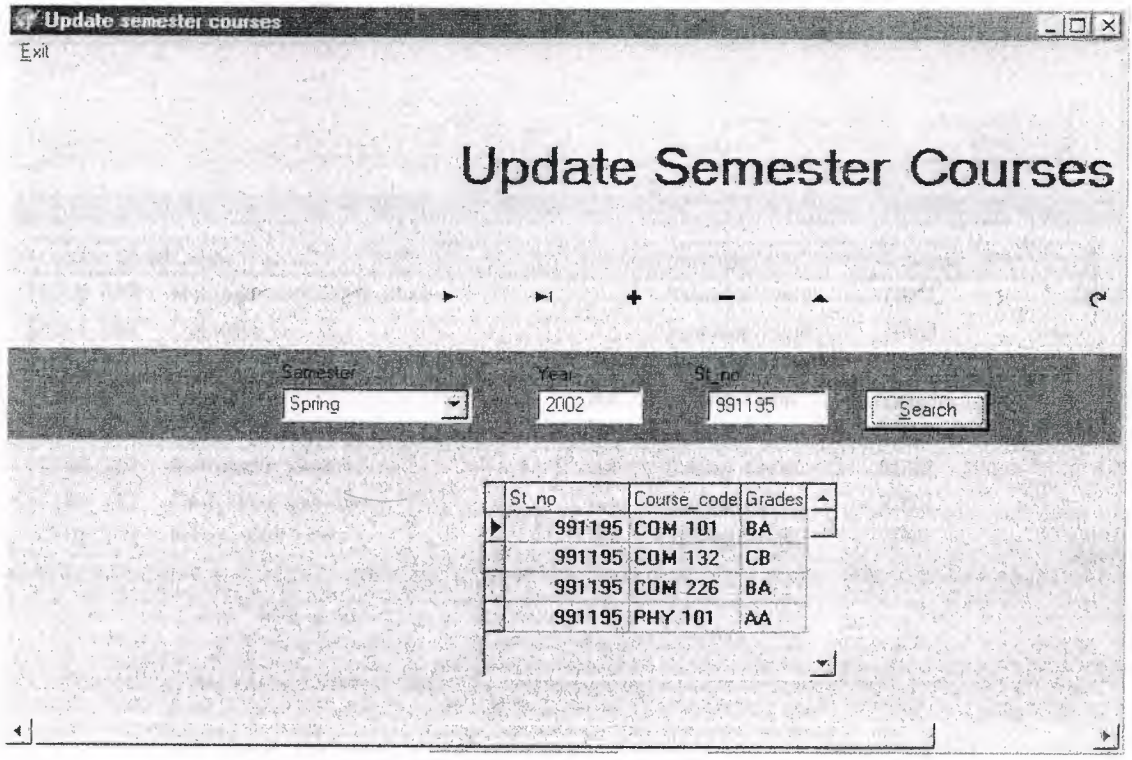

Figure 3.9 Update Semester courses Screen

3.8.9. Update whole Courses of student screen:

This screen for updating the student courses, it consists of:

1- Search button: the task of this button is to get the student courses from the file that save it on it, after we write the student number in the edit component as we see bellow.

2- DBgrid component: we talk about this component in the previous screen, and of course whole data of student are appearing in this component.

3- Navigator component: this component has some task buttons, these buttons are First, Previous, Last, Next, and Delete unnecessary data.

4- To return back to the main menu click on the exit button.

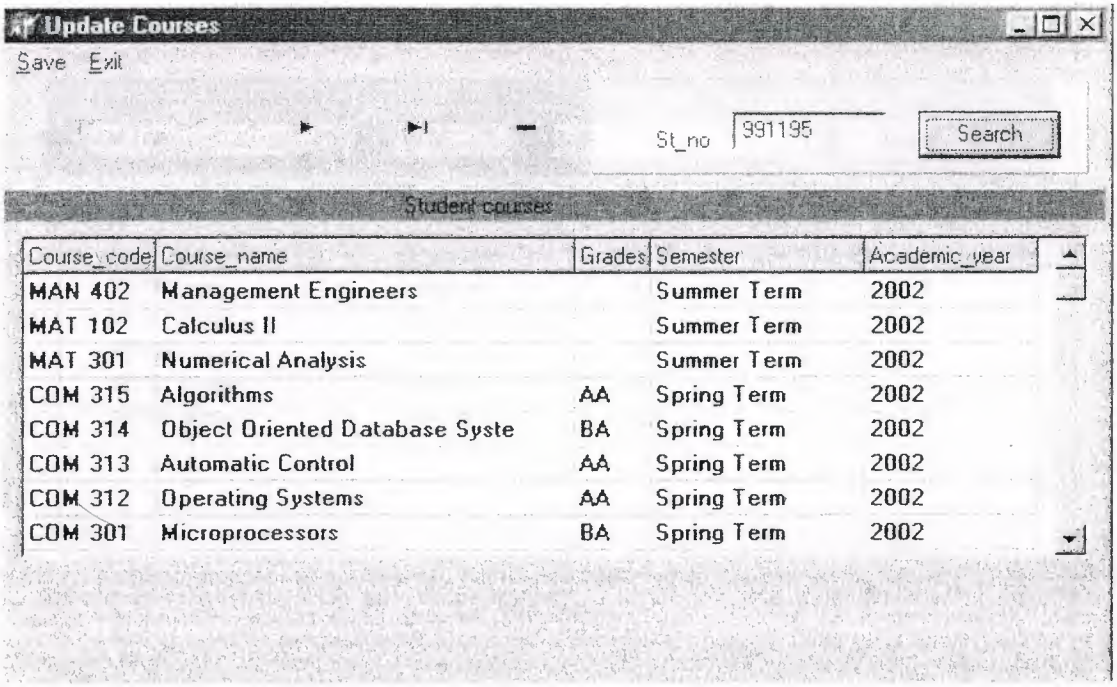

Figure 3.10 Update Whole Courses Screen

3.8.10. Record Grades Screen:

This screen for the computer department that records the grades of students, it consist three buttons, lookupcombobox, and combobox which are:

1- Lookupcombobox component: this component is getting the data from table and set these data to another table and here the data is courses of computer department.

2- Combobox component: this component used to select one of the values which it consist and here there are three value for semester name (fall, spring, summer).

3- Search Component: this button has only one task to get the data from the file, which saves the data on it, and here the data is the students' courses. By using Query component and using three information course codes, year, and semester.

4- Save button: this button is used to keep or save the data, after records the grades of the students to the suitable file.

5- DBgrid component: in this component there is operation that when we record the grade of student in the field which consist of the grades since we must double click to the next field in the dbgrid that called (Grt), to compute the value of the grades. For example if the grade is BB that means the value of this grade is 3.0, BA means 3.5 and so on.

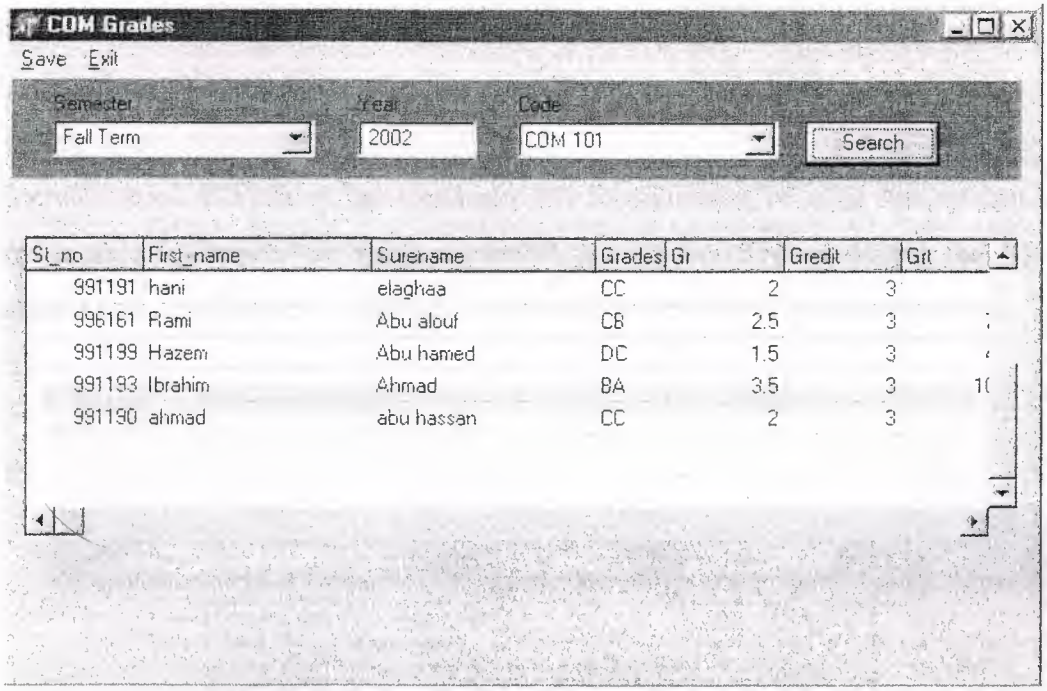

Figure 3.11 Records COM Grades Screen

3.8.11. CGPA Screen:

This screen or form for CGPA calculation of the students and we see how to calculate the CGPA in chapter one. In this form we use search button by using student number and DBgrid for displaying whole courses of student and by using search button the CGPA for the student will be appeared in the edit component as you see in the screen bellow.

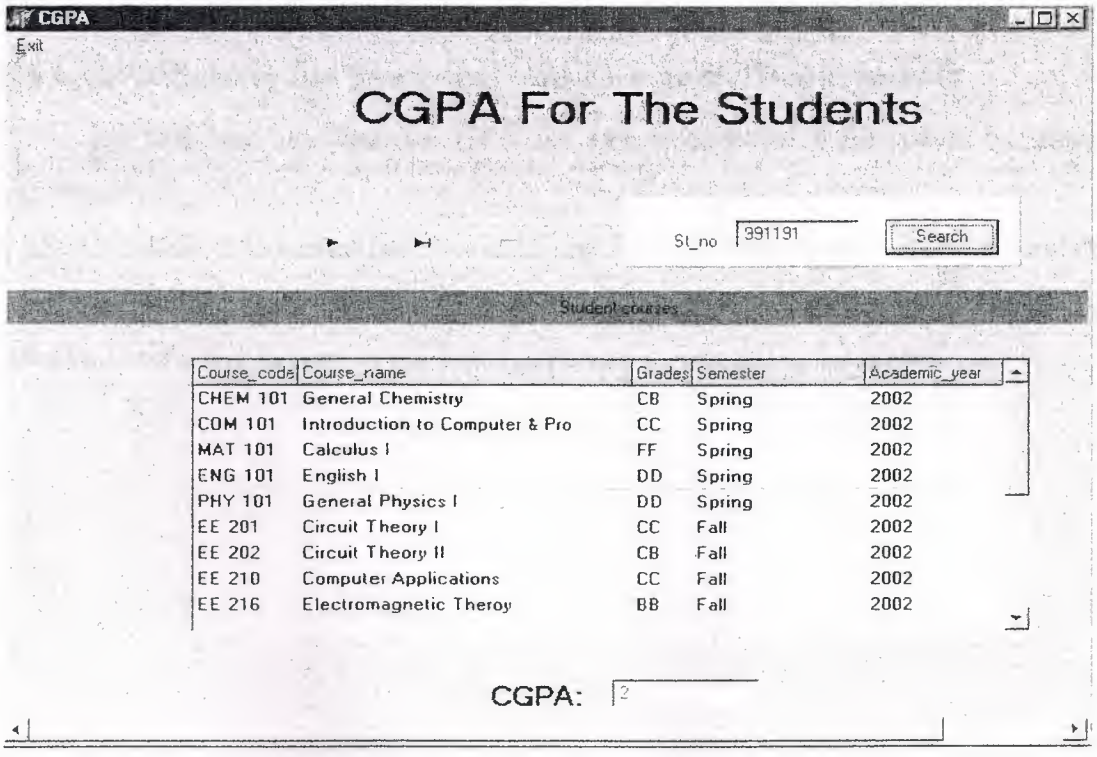

Figure 3.12 CGPA of Students Screen

3.8.12. GPA Screen for Computer Department:

In this form or screen we calculate the GPA for any semester of the students, that by using some components:

1- Search button: this button has tow tasks one for searching by used student number, year, and name of semester by using combobox list, second for calculating the GPA of the student and it will appear in the edit component as you see in the screen bellow:

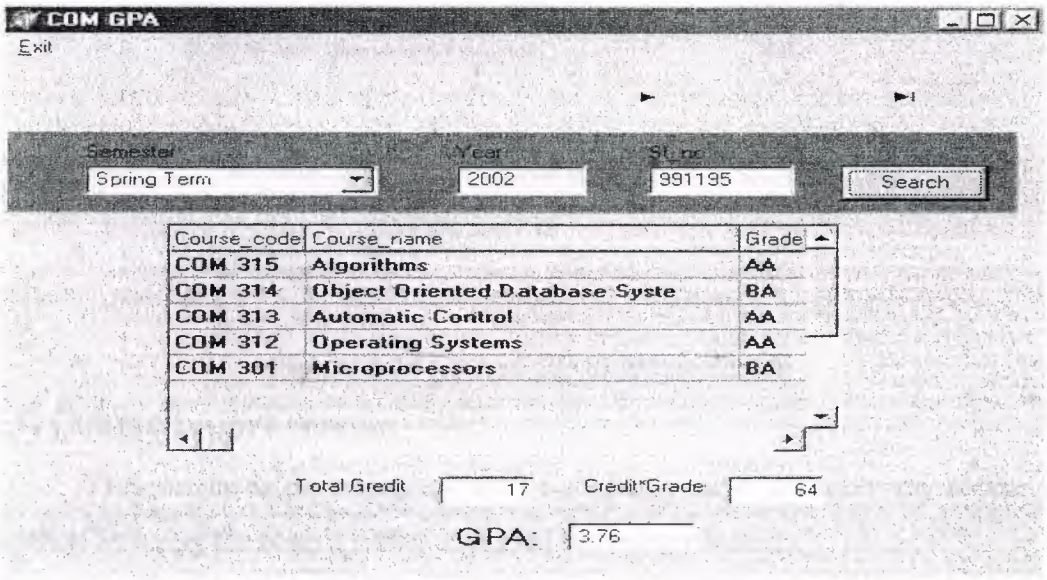

Figure 3.13 GPA of COM Student Scree

3.8.13. GPA Screen for Electrical and Electronic Department:

In this form we calculate the GPA for any semester of the students by using some components:

1- Search button: this button has tow tasks one for searching by used student number, year, and name of semester by using combo box list, second for calculate the GPA of the student and it will appear in the edit component as you see in the screen bellow:

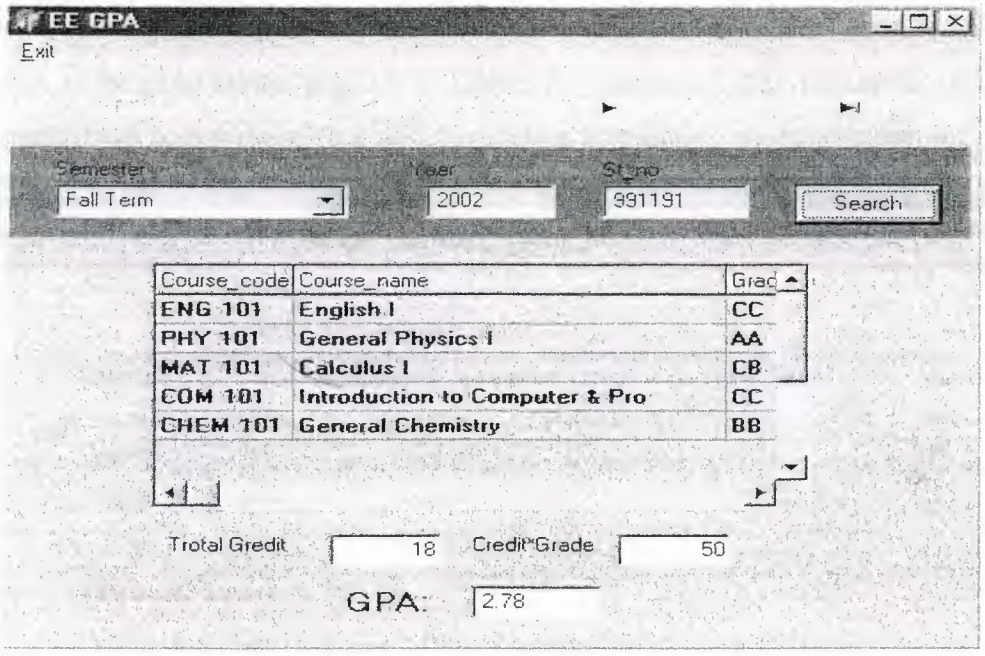

Figure 3.14 GPA of EE Students Screen

3.8.14. Course report screen:

This screen is preparing the list of students, these are taken the course which mention by using the search button and used the printer button.

Search button: we used this button to bring the list of students by using the course code, year, and name of semester and also used the query property.

Printer preview button: this button used to display the information of students these are student number, student name, student surname, and the grade of student as you see in the (figure3.16).

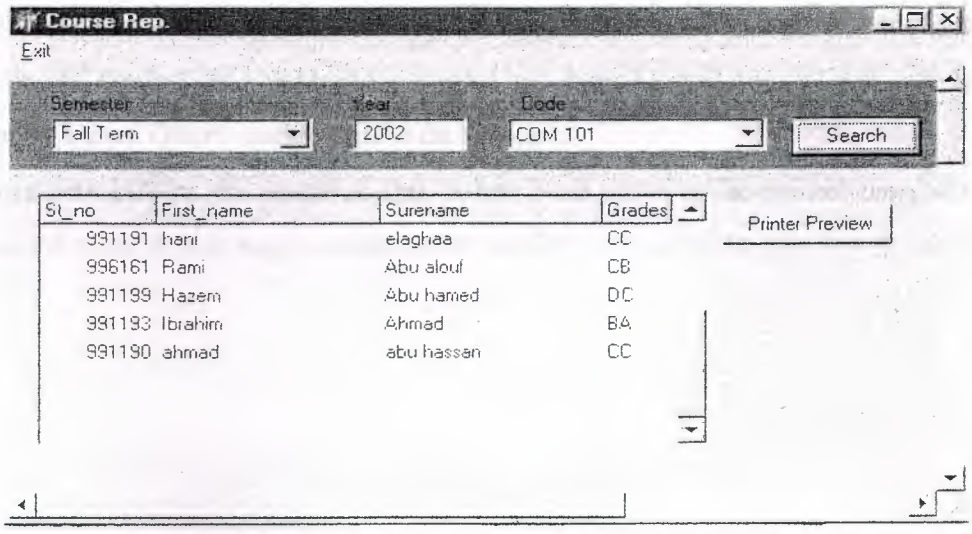

Figure 3.15 View Grades Screen

3.8.15. Printer Preview screen:

This screen is the page printer preview in Delphi programming, that display the information on it and if we accept these data we can printout by pressing the printer button.

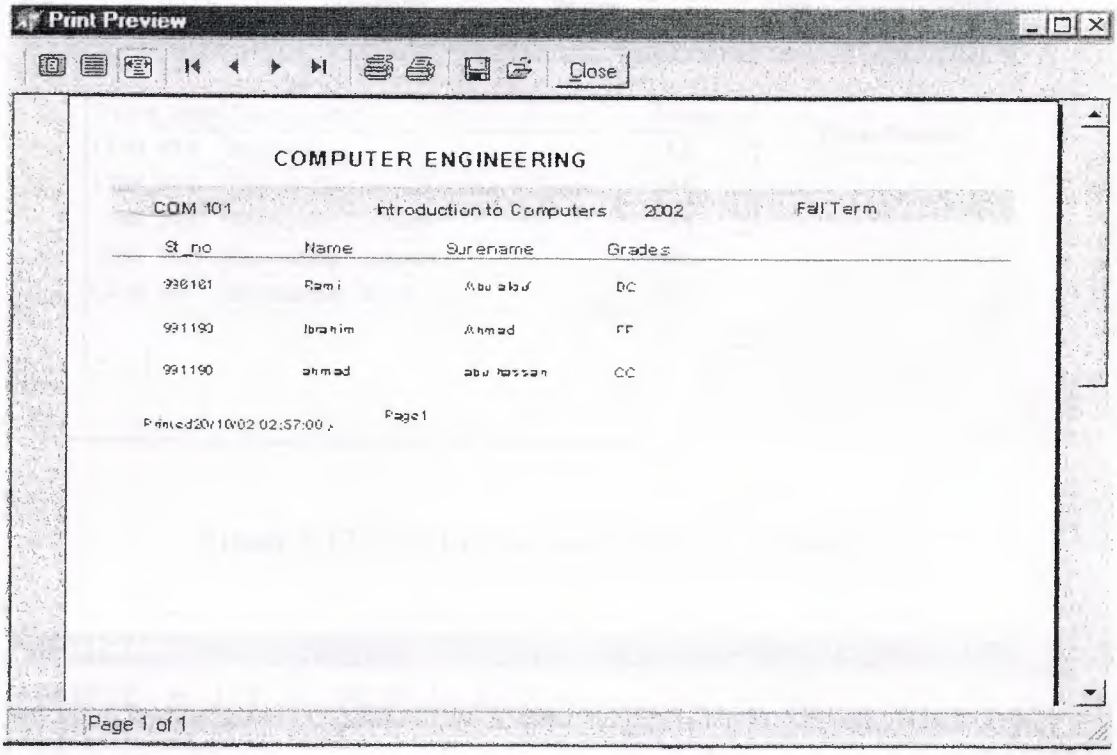

Figure 3.16 Grades of students Page

3.8.16. Grades of student in a semester screen:

In this form or screen there are the grades of student in one semester, we get these information by using search button and there is printer preview button:

Search button: the usage of this button to looking for the student information by chose the number of student and year and name of semester as you see bellow, and for search property I used Query component to do this task.

Printer preview button: the usage of this button is to show us the printer page after get the grades of student, this page contains also the GPA of the student as you see in figure().

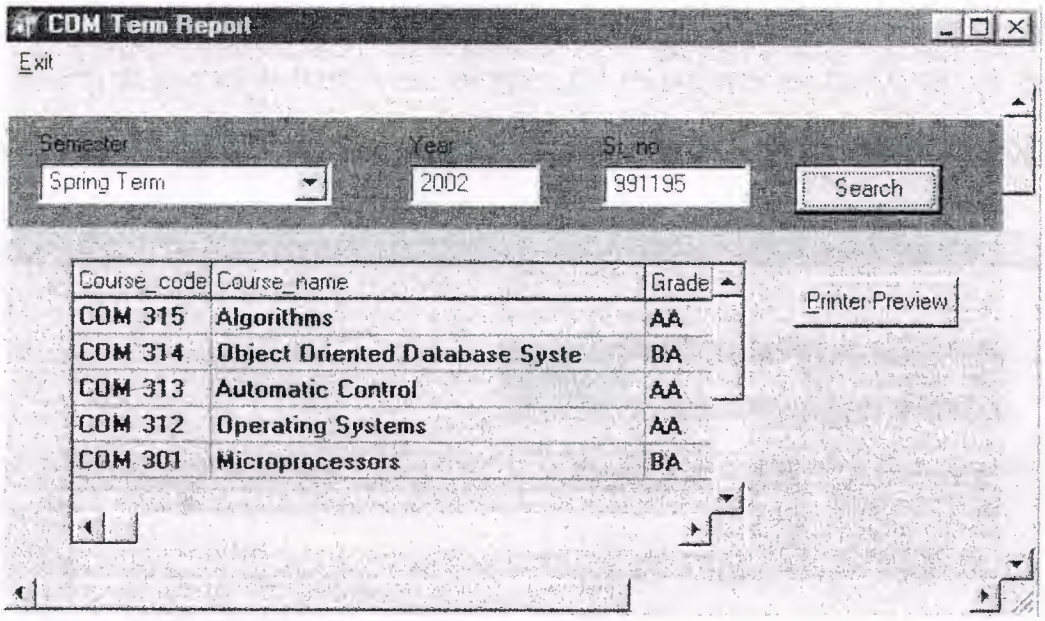

Figure 3.17 View Courses and Grades to Printed Screen

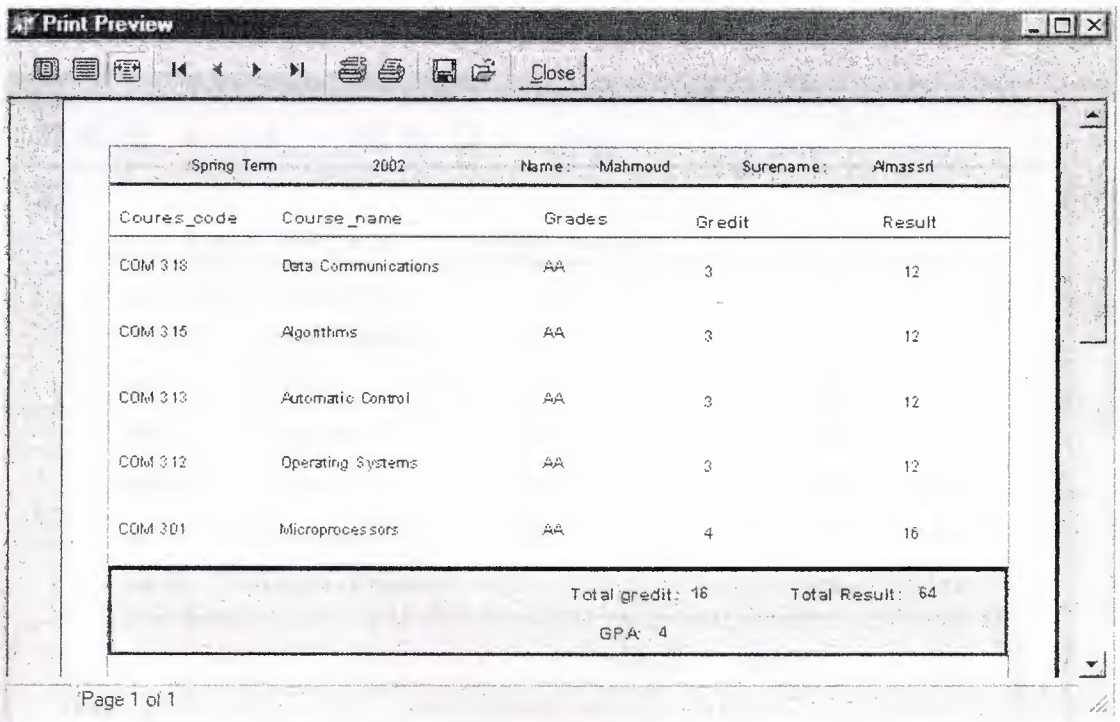

Figure 3.18 Grades and GPA Page Screen

3.8.17. Whole courses of student screen:

This screen gives us the whole courses of student by using search button for the number of student. By using filter property, and there is a printer preview button that

used to prepare the printer page of these courses with CGPA of the whole courses of the student as you see bellow, when we press the printer preview button we see the printer page that in (figure 3.20).

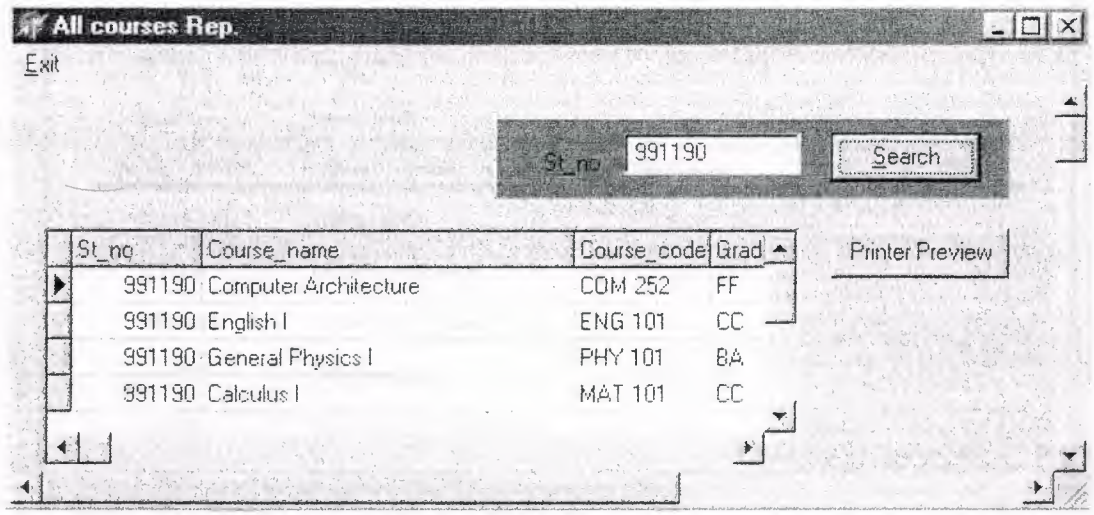

Figure 3.19 View All Courses Screen

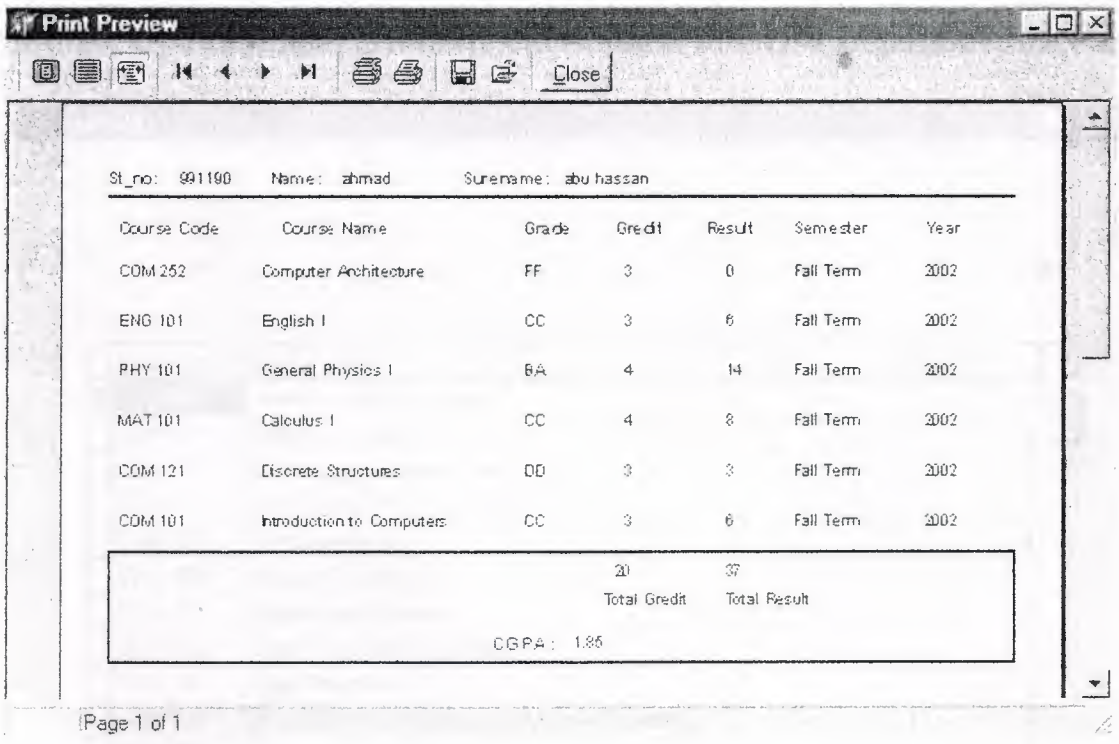

Figure 3.20 All Course and CGPA of students Screen

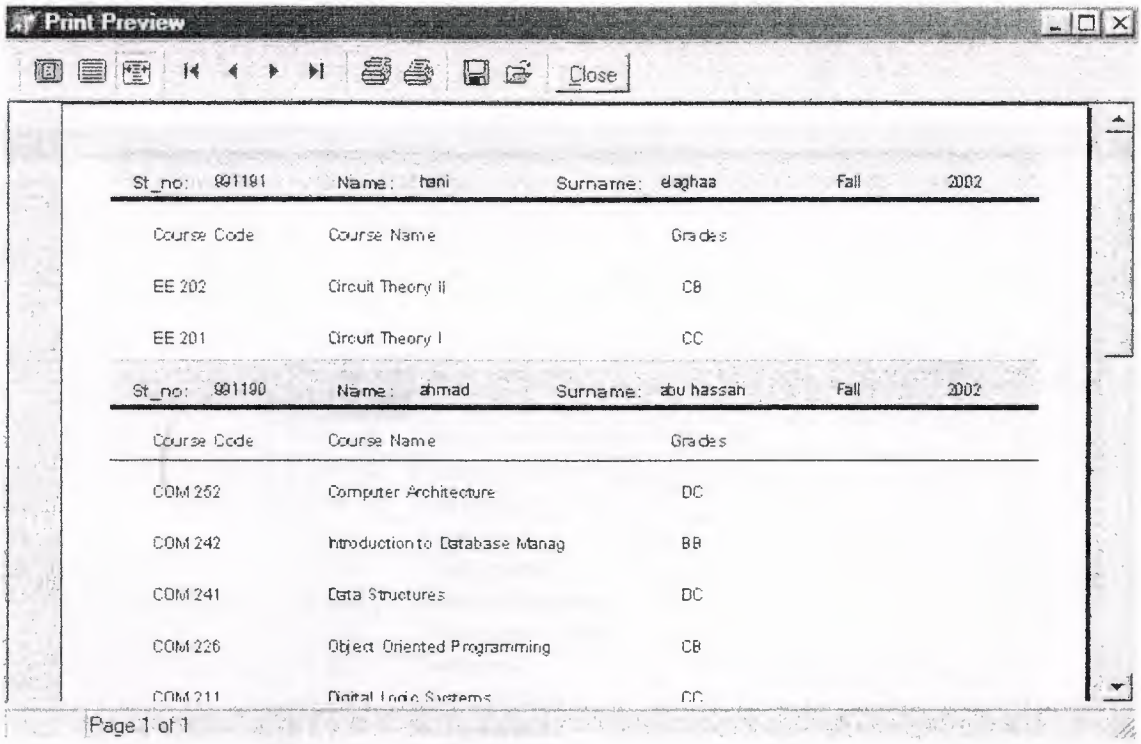

Figure 3.21 Courses of Students in One Semester

3.8.18. Update undergraduate curriculum of COM Department screen

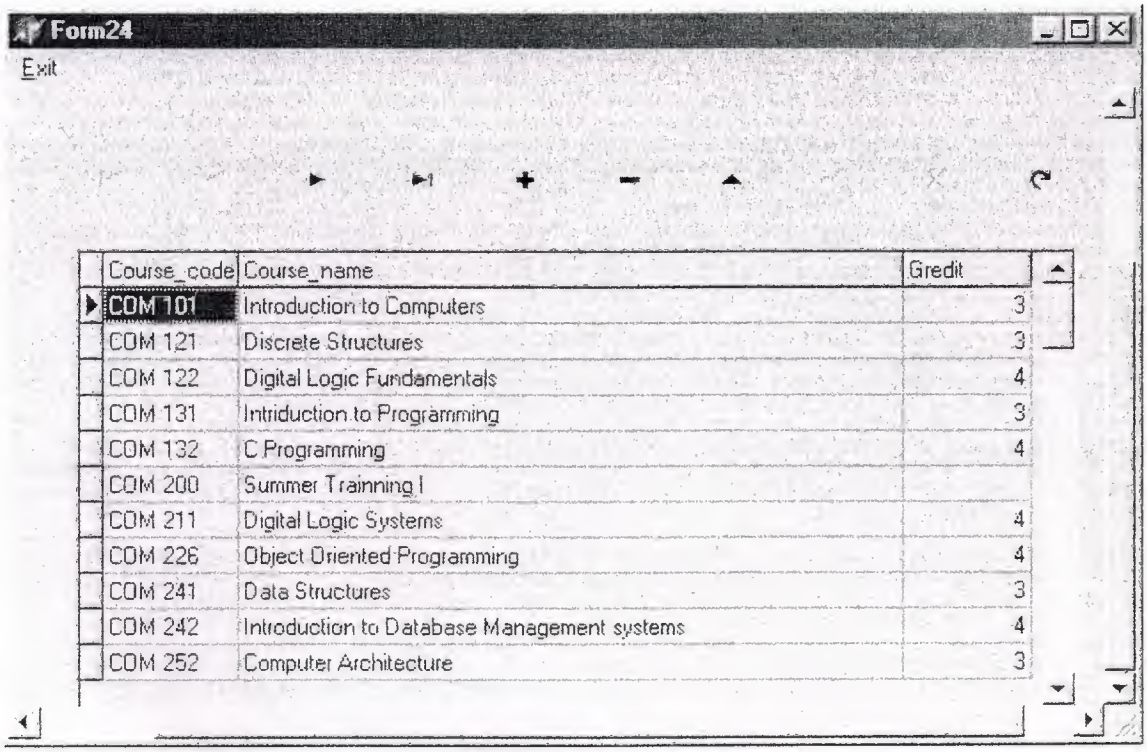

Figure 3.22 Update Undergraduate Curriculum of COM Dept. screen

3.8.19. Update Undergraduate Curriculum of EE Department Screen

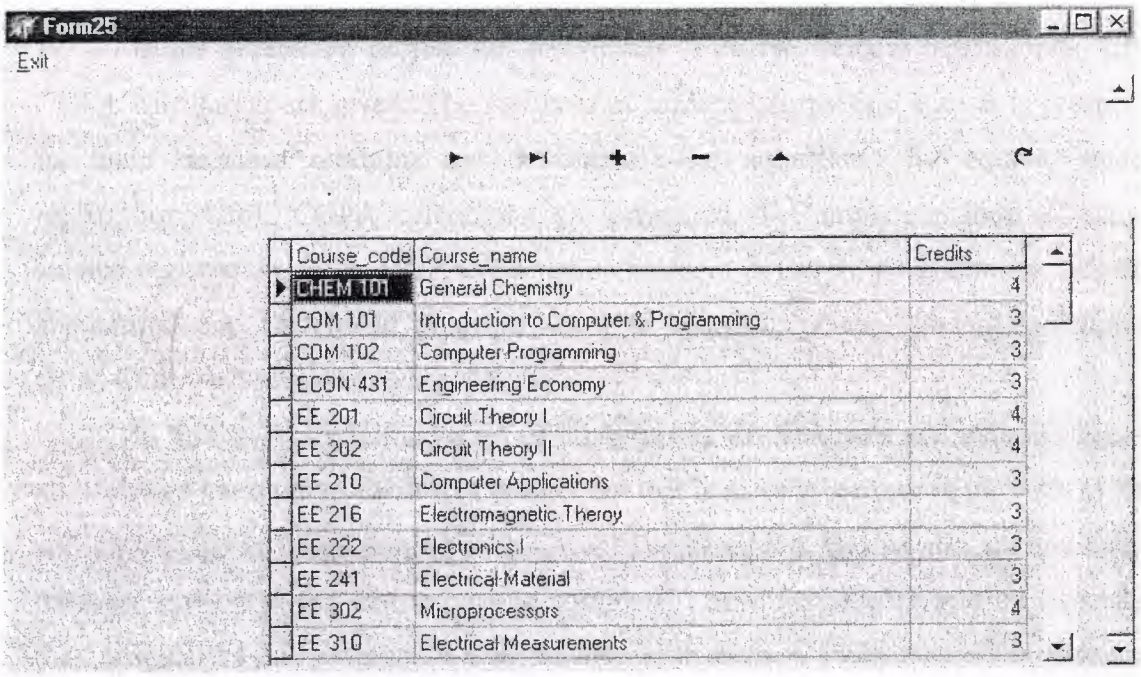

Figure 2.23 Update Undergraduate Curriculum of EE Dept. screen

CONCLUSION

In the graduation project the description of course, student registrations, GPA, CGPA calculation are given. The structure of student information system is presented. Its main database modules are developed. The algorithms for course, student registration, GPA, CGPA calculation are presented. The implementation of course, student registration, GPA, CGPA calculation problem in Delphi programming language are carried out. Developed program allow to automate course, student registration, GPA, CGPA calculation process.

In this project I learned a lot of thing that in the first time and even though not all of things I wanted to do in this project but this is mainly because of the lack of time and knowledge in programming with Delphi Programming. But we can say the Delphi database support is very extensive and complete. I have very high hopes on expanding the capability of this program in near future and from there I will take-off in mastering Delphi to design any project. I will try to take a lot of experience which is very important tool that I will need to take any obstacles being faced in the future.

REFERENCES

Books

- Jeff Duntemann, Jim Mischel, and Don Taylor, "Delphi Programming Explorer", the coriolis group inc. 1995.

- Marco Cantu, "Mastering Delphi", SYBEX, second edition.

- Gary Cornell,"Delphi nuts & Bolts for Experienced Programmers", McGrow-Hill, second edition.

Websites

- www.sybex.com

-www.marcocantu.com

- www.kdtool.net

APPENDIX

1- Main menu source code:

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Menus, StdCtrls, ExtCtrls;

type
  TForm1 = class(TForm)
    MainMenu1: TMainMenu;
    Registration1: TMenuItem;
    Registernewstudent1: TMenuItem;
    information1: TMenuItem;
    registercourses1: TMenuItem;
    Reports1: TMenuItem;
    Search1: TMenuItem;
    Update1: TMenuItem;
    Searchbystno1: TMenuItem;
    Searchbycourses1: TMenuItem;
    About1: TMenuItem;
    About2: TMenuItem;
    Searchforsemester1: TMenuItem;
    RegisterEECourses1: TMenuItem;
    EEstudentSemester1: TMenuItem;
    courses1: TMenuItem;
    Grades1: TMenuItem;
    Semester1: TMenuItem;
    Studentrep1: TMenuItem;
    CourseGradeforEE1: TMenuItem;
    StudentCourses1: TMenuItem;
    Exit1: TMenuItem;
    StudentInformation1: TMenuItem;
    studentCouses1: TMenuItem;
    Help1: TMenuItem;
    StudentSemesterCourses1: TMenuItem;
    constantes1: TMenuItem;
    Semester2: TMenuItem;
    WholeStudentsGrades1: TMenuItem;
    ProgramOfCOMDepartment1: TMenuItem;
    ProgramofEEDepartment1: TMenuItem;
    procedure Registernewstudent1Click(Sender: TObject);
    procedure registercourses1Click(Sender: TObject);
    procedure Search1Click(Sender: TObject);
    procedure Update1Click(Sender: TObject);
```

```
    procedure Searchbystno1Click(Sender: TObject);
    procedure Searchbycourses1Click(Sender: TObject);
    procedure Searchforsemester1Click(Sender: TObject);
    procedure RegisterEECourses1Click(Sender: TObject);
    procedure EEstudentSemester1Click(Sender: TObject);
    procedure courses1Click(Sender: TObject);
    procedure Grades1Click(Sender: TObject);
    procedure Semester1Click(Sender: TObject);
    procedure Studentrep1Click(Sender: TObject);
    procedure CourseGradeforEE1Click(Sender: TObject);
    procedure StudentCourses1Click(Sender: TObject);
    procedure Exit1Click(Sender: TObject);
    procedure StudentInformation1Click(Sender: TObject);
    procedure studentCouses1Click(Sender: TObject);
    procedure StudentSemesterCourses1Click(Sender: TObject);
    procedure Semester2Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure WholeStudentsGrades1Click(Sender: TObject);
    procedure ProgramOfCOMDepartment1Click(Sender: TObject);
    procedure ProgramofEEDepartment1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

uses Unit2, Unit3, Unit5,unit4,unit6,unit7,unit8,unit10,unit11,unit12,unit13,unit9
,unit14,unit15,unit16,unit17,unit18,unit19,unit20,unit21,unit22,unit23,unit24,unit25;

{$R *.DFM}

procedure TForm1.Registernewstudent1Click(Sender: TObject);
begin
  Application.CreateForm(TForm2, Form2);
  form2.show;

end;

procedure TForm1.registercourses1Click(Sender: TObject);
begin
Application.CreateForm(TForm5, Form5);
form5.show;
//if Edit1.Text<>'' then
//form5.dbedit6.Text:=edit1.Text;
```

```
end;

procedure TForm1.Search1Click(Sender: TObject);
begin
Application.CreateForm(TForm6, Form6);
form6.show;
//if Edit1.Text<>'' then
//form6.edit2.Text:=edit1.Text;
end;

procedure TForm1.Update1Click(Sender: TObject);
begin
application.Createform(TForm4,form4);
form4.show;
//if Edit1.Text<>'' then
//form4.edit1.Text:=edit1.Text;
end;

procedure TForm1.Searchbystno1Click(Sender: TObject);
begin
 Application.CreateForm(TForm7, Form7);
 form7.show;
end;

procedure TForm1.Searchbycourses1Click(Sender: TObject);
begin
Application.CreateForm(TForm8, Form8);
form8.show;
end;

procedure TForm1.Searchforsemester1Click(Sender: TObject);
begin
Application.CreateForm(TForm10, Form10);
form10.show;
//if Edit1.Text<>'' then
//form10.edit2.Text:=edit1.Text;
end;

procedure TForm1.RegisterEECourses1Click(Sender: TObject);
begin
 Application.CreateForm(TForm11, Form11);
 form11.show;
 //if Edit1.Text<>'' then
//form11.dbedit6.Text:=edit1.Text;
end;

procedure TForm1.EEstudentSemester1Click(Sender: TObject);
begin
 Application.CreateForm(TForm12, Form12);
```

```
 form12.show;
 //if Edit1.Text<>'' then
//form12.edit2.Text:=edit1.Text;
end;

procedure TForm1.courses1Click(Sender: TObject);
begin
Application.CreateForm(TForm13, Form13);
form13.show;
form13.QuickRep1.Preview;
form13.Visible:=false;
end;

procedure TForm1.Grades1Click(Sender: TObject);
begin
 Application.CreateForm(TForm9, Form9);
 form9.show;
 form9.QuickRep1.Preview;
 form9.visible:=false;
end;

procedure TForm1.Semester1Click(Sender: TObject);
begin
Application.CreateForm(TForm17, Form17);
form17.show;
//if Edit1.Text<>'' then
//form17.edit2.Text:=edit1.Text;
end;

procedure TForm1.Studentrep1Click(Sender: TObject);
begin
Application.CreateForm(TForm16, Form16);
form16.show;
//if Edit1.Text<>'' then
//form16.edit2.Text:=edit1.Text;

end;

procedure TForm1.CourseGradeforEE1Click(Sender: TObject);
begin
 Application.CreateForm(TForm18, Form18);
 form18.show;
 //if Edit1.Text<>'' then
//form18.edit2.Text:=edit1.Text;
end;

procedure TForm1.StudentCourses1Click(Sender: TObject);
begin
Application.CreateForm(TForm19, Form19);
form19.show;
```

```
end;

procedure TForm1.Exit1Click(Sender: TObject);
begin
close;
end;

procedure TForm1.StudentInformation1Click(Sender: TObject);
begin
Application.CreateForm(TForm3, Form3);
form3.show;
end;

procedure TForm1.studentCouses1Click(Sender: TObject);
begin
Application.CreateForm(TForm20, Form20);
 form20.show;
end;

procedure TForm1.StudentSemesterCourses1Click(Sender: TObject);
begin
 Application.CreateForm(TForm21, Form21);
 form21.show;
// if Edit1.Text<>'' then
//form21.edit2.Text:=edit1.Text;
end;

procedure TForm1.Semester2Click(Sender: TObject);
begin
Application.CreateForm(TForm22, Form22);
form22.show;
end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
action:=cafree;
end;

procedure TForm1.WholeStudentsGrades1Click(Sender: TObject);
begin
Application.CreateForm(TForm23, Form23);
form23.show ;
form23.QuickRep1.Preview;
form23.Visible:=false;
end;

procedure TForm1.ProgramOfCOMDepartment1Click(Sender: TObject);
begin
Application.CreateForm(TForm24, Form24);
form24.show;
```

```
end;

procedure TForm1.ProgramofEEDepartment1Click(Sender: TObject);
begin
Application.CreateForm(TForm25, Form25);
form25.show;
end;
```

2- Register New Student source code:

```
unit Unit2;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Db, DBTables, StdCtrls, DBCtrls, Mask, Menus, ExtCtrls;

type
  TForm2 = class(TForm)
    DBEdit1: TDBEdit;
    Label2: TLabel;
    DBEdit2: TDBEdit;
    Label5: TLabel;
    DBEdit4: TDBEdit;
    Label6: TLabel;
    DBEdit5: TDBEdit;
    Label7: TLabel;
    DBEdit6: TDBEdit;
    Label8: TLabel;
    Label9: TLabel;
    Table1: TTable;
    DataSource1: TDataSource;
    MainMenu1: TMainMenu;
    Label10: TLabel;
    DBEdit9: TDBEdit;
    RadioButton1: TRadioButton;
    DBEdit7: TDBEdit;
    Label3: TLabel;
    DBEdit3: TDBEdit;
    Label1: TLabel;
    Button1: TButton;
    Panel1: TPanel;
    GroupBox1: TGroupBox;
    Label4: TLabel;
    Label11: TLabel;
    Label12: TLabel;
    Label13: TLabel;
    Label14: TLabel;
    Label15: TLabel;
```

```
    DBEdit10: TDBEdit;
    DBEdit11: TDBEdit;
    Panel2: TPanel;
    DBLookupComboBox2: TDBLookupComboBox;
    DBEdit12: TDBEdit;
    Button2: TButton;
    DataSource2: TDataSource;
    Table2: TTable;
    DataSource3: TDataSource;
    Table3: TTable;
    MainMenu2: TMainMenu;
    Save1: TMenuItem;
    MenuItem1: TMenuItem;
    Exit2: TMenuItem;
    Table4: TTable;
    DataSource4: TDataSource;
    Button3: TButton;
    Exit1: TMenuItem;
    Button4: TButton;
    DBEdit13: TDBEdit;
    DataSource5: TDataSource;
    Table5: TTable;
    DBLookupComboBox1: TDBLookupComboBox;
    DBEdit14: TDBEdit;
    Panel3: TPanel;
    Panel4: TPanel;
    Label16: TLabel;
    Label17: TLabel;
    DBEdit15: TDBEdit;
    DBEdit16: TDBEdit;
    Table6: TTable;
    Label18: TLabel;
    DBComboBox1: TDBComboBox;
    procedure Button1Click(Sender: TObject);
    procedure Exit2Click(Sender: TObject);
    procedure New1Click(Sender: TObject);
    procedure Save1Click(Sender: TObject);
    procedure RadioButton1Click(Sender: TObject);
    procedure Exit1Click(Sender: TObject);
    procedure Edit1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure DBLookupComboBox2Click(Sender: TObject);
    procedure DBLookupComboBox1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure GroupBox1Click(Sender: TObject);
  private
    { Private declarations }
  public
```

```
  { Public declarations }
 end;

var
 Form2: TForm2;

Implementation

uses Unit1;

{$R *.DFM}

procedure TForm2.Button1Click(Sender: TObject);
begin
//Application.CreateForm (TForm9, Form9);
 //form9.show;
 groupbox1.Visible:=true;
 table2.Insert;
 dbedit11.Text:=dbedit1.Text;
 dbedit12.Text:='EX';
 dbedit15.text:='0.0';
  dbedit16.Text:='0.0';
  dbedit10.Text:=table6.Fields[1].text;

end;

procedure TForm2.Exit2Click(Sender: TObject);
begin
close;
end;

procedure TForm2.New1Click(Sender: TObject);
begin
table1.Insert;
end;

procedure TForm2.Save1Click(Sender: TObject);
begin
table1.Post;
table1.insert;
end;

procedure TForm2.RadioButton1Click(Sender: TObject);
begin
if radiobutton1.Checked then
label3.Visible:=true;
dbedit7.Visible:=true;

end;
```

```
procedure TForm2.Exit1Click(Sender: TObject);
begin
close;
end;

procedure TForm2.Edit1Click(Sender: TObject);
begin
table1.Post;
table1.Insert;
end;

procedure TForm2.Button2Click(Sender: TObject);
begin
form2.Visible:=true;
form9.visible:=false;
groupbox1.Visible:=false;
end;

procedure TForm2.Button3Click(Sender: TObject);
begin
table2.Post;
table2.Insert;
dbedit11.Text:=dbedit1.Text;
  dbedit12.Text:='EX';
  dbedit15.Text:='0.0';
  dbedit16.Text:='0.0';
  dbedit10.Text:=table6.Fields [1].text;
end;

procedure TForm2.Button4Click(Sender: TObject);
begin
table1.Post;
table1.Insert;
dbedit1.SetFocus;
end;

procedure TForm2.DBLookupComboBox2Click(Sender: TObject);
begin
if dblookupcombobox2.Text<>'' then
dbedit13.Text:=table4['course_name'];
end;

procedure TForm2.DBLookupComboBox1Click(Sender: TObject);
begin
if dblookupcombobox1.Text<>'' then
dbedit14.Text:=table5 ['course_name'];
end;

procedure TForm2.FormClose(Sender: TObject; var Action: TCloseAction);
begin
```

```
action:=cafree;
end;

procedure TForm2.GroupBox1Click(Sender: TObject);
begin
end;
end.
```

3- Update student Information source code:

```
unit Unit3;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Db, DBTables, StdCtrls, DBCtrls, Mask, ExtCtrls, Menus, Grids, DBGrids;

type
  TForm3 = class(TForm)
    Panel1: TPanel;
    Button1: TButton;
    Edit1: TEdit;
    MainMenu1: TMainMenu;
    File1: TMenuItem;
    Save1: TMenuItem;
    Delete1: TMenuItem;
    Label3: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    DBEdit4: TDBEdit;
    DBEdit5: TDBEdit;
    DBEdit6: TDBEdit;
    DBEdit8: TDBEdit;
    DBEdit9: TDBEdit;
    RadioButton1: TRadioButton;
    DBEdit7: TDBEdit;
    Table2: TTable;
    DataSource2: TDataSource;
    Label4: TLabel;
    Panel2: TPanel;
    Label12: TLabel;
```

```
    DBEdit3: TDBEdit;
    Label2: TLabel;
    Label1: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Save1Click(Sender: TObject);
    procedure Update1Click(Sender: TObject);
    procedure Exit1Click(Sender: TObject);
    procedure Delete1Click(Sender: TObject);
    procedure RadioButton1Click(Sender: TObject);
    procedure File1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form3: TForm3;

implementation

uses Unit1;

{$R *.DFM}

procedure TForm3.Button1Click(Sender: TObject);
begin
table2.Filtered:=false;
table2.Filter:='st_no='+edit1.Text;
table2.Filtered:=true;
end;

procedure TForm3.Button2Click(Sender: TObject);
begin
form1.visible:=true;
form3.Visible:=false;
end;

procedure TForm3.Save1Click(Sender: TObject);
begin
table2.Post;
table2.insert;
end;
procedure TForm3.Update1Click(Sender: TObject);
begin
table2.Insert;
end;
```

```
procedure TForm3.Exit1Click(Sender: TObject);
begin
close;
end;

procedure TForm3.Delete1Click(Sender: TObject);
begin
if messagedlg('Are you sure you want to delete the current reecords?',
mtconfirmation,[mbyes,mbno],0)=idyes then
table2.delete;
end;

procedure TForm3.RadioButton1Click(Sender: TObject);
begin
if radiobutton1.Checked then
label11.Visible:=true;
dbedit7.Visible:=true;
end;

procedure TForm3.File1Click(Sender: TObject);
begin
close;
end;

procedure TForm3.FormClose(Sender: TObject; var Action: TCloseAction);
begin
action:=cafree;
end;
end.
```

4- Records Students Courses Source Code:

```
unit Unit5;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Db, DBTables, Mask, DBCtrls, StdCtrls, ExtCtrls, Menus, Grids, DBGrids,
  Buttons
  ;
type
  TForm5 = class(TForm)
    Label1: TLabel;
    Table1: TTable;
    DataSource1: TDataSource;
    MainMenu1: TMainMenu;
    Label8: TLabel;
    DataSource2: TDataSource;
```

```
    Table2: TTable;
    Button1: TButton;
    Panel1: TPanel;
    Edit1: TEdit;
    Label4: TLabel;
    Table3: TTable;
    DataSource3: TDataSource;
    Label3: TLabel;
    DBEdit1: TDBEdit;
    Exit1: TMenuItem;
    DBEdit3: TDBEdit;
    Panel2: TPanel;
    DBEdit5: TDBEdit;
    DBEdit6: TDBEdit;
    Label2: TLabel;
    Label7: TLabel;
    Exit2: TMenuItem;
    DBLookupComboBox1: TDBLookupComboBox;
    DBEdit2: TDBEdit;
    Panel3: TPanel;
    Label6: TLabel;
    DBEdit4: TDBEdit;
    DBEdit7: TDBEdit;
    Label5: TLabel;
    Table4: TTable;
    DataSource4: TDataSource;
    DBEdit8: TDBEdit;
    Label9: TLabel;
    Button2: TButton;
    SpeedButton2: TSpeedButton;
    Label10: TLabel;
    procedure Button3Click(Sender: TObject);
    procedure Exit1Click(Sender: TObject);
    procedure Save1Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Edit2Click(Sender: TObject);
    procedure Exit2Click(Sender: TObject);
    procedure DBLookupComboBox1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form5: TForm5;
```

```
implementation

uses unit1,unit22;

{$R *.DFM}

procedure TForm5.Button3Click(Sender: TObject);
begin
table1.Insert;
end;
procedure TForm5.Exit1Click(Sender: TObject);
begin
table1.Post;
table1.Insert;
if dbedit7.Text<>'' then
dbedit3.Text:=table3['st_no'];
if dbedit7.Text<>'' then
dbedit1.Text:=table3['first_name'];
if dbedit7.Text<>'' then
dbedit2.Text:=table3['surename'];
dbedit6.text:=table4.Fields[1].Text;
dbedit8.Text:=table4.Fields[0].Text;
end;
procedure TForm5.Save1Click(Sender: TObject);
begin
 table1.Post;
end;

procedure TForm5.Button1Click(Sender: TObject);
begin
table3.filtered:=false;
table3.filter:='st_no='+edit1.text;
table3.filtered:=true;
table1.insert;
if dbedit7.Text<>'' then
dbedit3.Text:=table3['st_no'];
if dbedit7.Text<>'' then
dbedit1.Text:=table3['first_name'];
if dbedit7.Text<>'' then
dbedit2.Text:=table3['surename'];
dbedit6.text:=table4.Fields[1].Text;
dbedit8.Text:=table4.Fields[0].Text;

end;
procedure TForm5.Button2Click(Sender: TObject);
begin
table1.insert;
end;
```

```
procedure TForm5.Edit2Click(Sender: TObject);
begin
table1.insert;
if dbedit7.Text<>'' then
dbedit3.Text:=table3['st_no'];
if dbedit7.Text<>'' then
dbedit1.Text:=table3['first_name'];
if dbedit7.Text<>'' then
dbedit2.Text:=table3['surename'];
end;

procedure TForm5.Exit2Click(Sender: TObject);
begin
close;
end;

procedure TForm5.DBLookupComboBox1Click(Sender: TObject);
begin
if dblookupcombobox1.Text<>'' then
dbedit4.Text:=table2['course_name'];
if dblookupcombobox1.Text<>'' then
 dbedit5.Text:=table2['gredit'] ;
end;

procedure TForm5.FormCreate(Sender: TObject);
begin
dbedit6.text:=table4.Fields[1].Text;
 dbedit8.Text:=table4.Fields[0].Text;
end;

procedure TForm5.SpeedButton1Click(Sender: TObject);
begin
Application.CreateForm(TForm22, Form22);
form22.show;
form5.Close;
end;
procedure TForm5.FormClose(Sender: TObject; var Action: TCloseAction);
begin
action:=cafree;
end;

end.
```

5- Records Students Grades Source Code:

```
unit Unit6;

interface
```

```
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Grids, DBGrids, Db, DBTables, ExtCtrls, StdCtrls, DBCtrls, Menus, Mask;

type
  TForm6 = class(TForm)
    Panel1: TPanel;
    Button1: TButton;
    Label1: TLabel;
    Table1: TTable;
    DBGrid1: TDBGrid;
    Edit2: TEdit;
    MainMenu1: TMainMenu;
    Edit4: TMenuItem;
    Exit1: TMenuItem;
    DataSource2: TDataSource;
    Label2: TLabel;
    Label3: TLabel;
    Table2: TTable;
    DataSource3: TDataSource;
    DataSource1: TDataSource;
    Query1: TQuery;
    ComboBox1: TComboBox;
    DBLookupComboBox1: TDBLookupComboBox;
    Table3: TTable;
    Label4: TLabel;
    Button2: TButton;
    procedure Button1Click(Sender: TObject);
    procedure Exit1Click(Sender: TObject);
    procedure Save1Click(Sender: TObject);
    procedure Edit4Click(Sender: TObject);
    procedure DBGrid1EditButtonClick(Sender: TObject);
    procedure DBGrid1DblClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form6: TForm6;

implementation

uses Unit1;

{$R *.DFM}
```

```
procedure TForm6.Button1Click(Sender: TObject);
VAR
str: string;

begin

str:='select*from grade where';

if DBLookupComboBox1.Text <> '' then
str:=str+' course_code='"+ DBLookupComboBox1.Text+'" and ';
if edit2.Text <>'' then
str:=str+' academic_year='"+ edit2.Text+'" and';
if combobox1.Text <> '' then
str:=str+' semester='"+ combobox1.Text+'" ';
query1.Close;
query1.sql.Clear;
query1.sql.add(str);
query1.Open;
 dbgrid1.Visible:=true;

 end;
procedure TForm6.Exit1Click(Sender: TObject);
begin
 close;
end;

procedure TForm6.Save1Click(Sender: TObject);
begin
table1.Post;
end;

procedure TForm6.Edit4Click(Sender: TObject);
begin
query1.Edit;
query1.Post;
edit2.text:=table3.Fields[1].Text;
end;
procedure TForm6.DBGrid1EditButtonClick(Sender: TObject);
begin
if dbgrid1.Fields[3].text='BB' then
dbgrid1.Fields[4].text:='4.0';
end;
procedure TForm6.DBGrid1DblClick(Sender: TObject);
begin
if dbgrid1.Fields[3].text='AA' then
dbgrid1.Fields[4].value:=4.0;
if dbgrid1.Fields[3].text='BA' then
dbgrid1.Fields[4].value:=3.5;
if dbgrid1.Fields[3].text='BB' then
dbgrid1.Fields[4].value:=3.0;
```

```
if dbgrid1.Fields[3].text='CB' then
dbgrid1.Fields[4].value:=2.5;
if dbgrid1.Fields[3].text='CC' then
dbgrid1.Fields[4].value:=2.0;
if dbgrid1.Fields[3].text='DC' then
dbgrid1.Fields[4].value:=1.5;
if dbgrid1.Fields[3].text='DD' then
dbgrid1.Fields[4].value:=1.0;
if dbgrid1.Fields[3].text='FD' then
dbgrid1.Fields[4].value:=0.5;
if dbgrid1.Fields[3].text='FF' then begin
dbgrid1.Fields[4].value:=0.0;
end;
dbgrid1.Fields[6].Value:=(dbgrid1.Fields[4].Value)*(dbgrid1.Fields[5].Value);
end;
procedure TForm6.FormCreate(Sender: TObject);
begin
edit2.text:=table3.Fields[1].Text;

end;

procedure TForm6.FormClose(Sender: TObject; var Action: TCloseAction);
begin
action:=cafree;
end;

procedure TForm6.Button2Click(Sender: TObject);
begin
query1.Edit;
query1.Post;
edit2.text:=table3.Fields[1].Text;
end;

end
```

6- Search Student Information Source Code:

```
unit Unit7;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls, Db, DBTables, Grids, DBGrids, Menus;

type
  TForm7 = class(TForm)
    DBGrid1: TDBGrid;
    Table1: TTable;
```

```
    DataSource1: TDataSource;
    Panel1: TPanel;
    Button1: TButton;
    Edit1: TEdit;
    Label1: TLabel;
    MainMenu1: TMainMenu;
    Exit1: TMenuItem;
    Label2: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure Exit1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form7: TForm7;

implementation

{$R *.DFM}
procedure TForm7.Button1Click(Sender: TObject);
begin
table1.Filtered:=false;
table1.Filter:='st_no='+edit1.Text;
table1.Filtered:=true;
end;
procedure TForm7.Exit1Click(Sender: TObject);
begin
close;
end;

procedure TForm7.FormClose(Sender: TObject; var Action: TCloseAction);
begin
action:=cafree;
end;
 end
```

7- CGPA Calculation Source Code:

```
unit Unit8;
interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Grids, DBGrids, Db, DBTables, StdCtrls, ExtCtrls, Menus, DBCtrls, Mask;
```

```
type
  TForm8 = class(TForm)
    Panel1: TPanel;
    Button1: TButton;
    Edit1: TEdit;
    Label1: TLabel;
    Panel2: TPanel;
    Table1: TTable;
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
    MainMenu1: TMainMenu;
    Exit1: TMenuItem;
    DBNavigator1: TDBNavigator;
    DataSource2: TDataSource;
    Query1: TQuery;
    Query2: TQuery;
    Query3: TQuery;
    DataSource3: TDataSource;
    Edit2: TEdit;
    Label4: TLabel;
    Label5: TLabel;
    Query4: TQuery;
    procedure Button1Click(Sender: TObject);
    procedure Exit1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form8: TForm8;

implementation

{$R *.DFM}

procedure TForm8.Button1Click(Sender: TObject);
var se,sw,st,sr:string ;
s,d,f:double;
begin
sw:='select * from grade where';
if edit1.Text <> '' then
sw:=sw+' st_no='''+ edit1.Text+''' ';
query1.Close;
query1.sql.Clear;
```

```
query1.sql.add(sw);
query1.Open;

se:='select  g1.st_no,sum(grt)/sum(gredit) as cgpa'    ;
se:=se+' from  grade g1,grade g2 '                     ;
se:=se+' where g1.ukey=g2.ukey'                        ;
se:=se+' and   g1.gr=(select max(g3.gr) from grade g3' ;
se:=se+'          where g1.st_no=g3.st_no'             ;
se:=se+'          and   g1.course_code=g3.course_code)' ;
se:=se+'          and g1.ukey not in(select distinct g5.ukey' ;
se:=se+'                        from  grade g5,grade g6'      ;
se:=se+'                        where g5.ukey<>g6.ukey'       ;
se:=se+'                        and   g5.st_no=g6.st_no'      ;
se:=se+'                        and   g5.course_code=g6.course_code' ;
se:=se+'                        and   g5.gr=g6.gr'            ;
se:=se+'                        and   g5.ukey<g6.ukey)'       ;
if edit1.Text <> '' then
se:=se+'and st_no='"'+ edit1.Text+'"';
se:=se+' group by g1.st_no';
query4.Close;
query4.sql.Clear;
query4.sql.add(se);
query4.Open;
edit2.Text:=query4.Fields[1].Text;
end;
procedure TForm8.Exit1Click(Sender: TObject);
begin
close;
end;

procedure TForm8.Button2Click(Sender: TObject);
begin
query1.Delete;
end;


procedure TForm8.FormClose(Sender: TObject; var Action: TCloseAction);
begin
action:=cafree;
end;

end.
```

8- GPA Calculation of Students Source Code:

```
unit Unit10;

interface

uses
```

```
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  DBTables, Db, Menus, StdCtrls, Grids, DBGrids, ExtCtrls, Mask, DBCtrls;
type
  TForm10 = class(TForm)
    Panel1: TPanel;
    Label2: TLabel;
    Label3: TLabel;
    Button1: TButton;
    DBGrid1: TDBGrid;
    Edit2: TEdit;
    ComboBox1: TComboBox;
    MainMenu1: TMainMenu;
    Exit1: TMenuItem;
    DataSource1: TDataSource;
    Query1: TQuery;
    Edit1: TEdit;
    Label4: TLabel;
    DataSource2: TDataSource;
    Table1: TTable;
    DataSource3: TDataSource;
    Table2: TTable;
    DBNavigator1: TDBNavigator;
    DBEdit1: TDBEdit;
    Query2: TQuery;
    DataSource4: TDataSource;
    Query3: TQuery;
    DataSource5: TDataSource;
    DBEdit2: TDBEdit;
    Edit3: TEdit;
    Label1: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Table3: TTable;
    Label7: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure Exit1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form10: TForm10;

implementation
```

```
{$R *.DFM}

procedure TForm10.Button1Click(Sender: TObject);
VAR
sw,sr,st,se: string;
s,d,f:double;
begin

sw:='select * from grade where';

if edit1.Text <> '' then
sw:=sw+' st_no='''+ edit1.Text+''' and ';
if edit2.Text <>'' then
sw:=sw+' academic_year='''+ edit2.Text+''' and';
if combobox1.Text <> '' then
sw:=sw+' semester='''+ combobox1.Text+''' ';
query1.Close;
query1.sql.Clear;
query1.sql.add(sw);
query1.Open;
 sr:='select sum(grt) from grade where';
if edit1.Text <> '' then
sr:=sr+' st_no='''+ edit1.Text+''' and ';
if edit2.Text <>'' then
sr:=sr+' academic_year='''+ edit2.Text+''' and';
if combobox1.Text <> '' then
sr:=sr+' semester='''+ combobox1.Text+''' ';

 query2.Close;
query2.sql.Clear;
query2.sql.add(sr);
query2.Open;
st:='select sum(gredit) from grade where';
if edit1.Text <> '' then
st:=st+' st_no='''+ edit1.Text+''' and ';
if edit2.Text <>'' then
st:=st+' academic_year='''+ edit2.Text+''' and';
if combobox1.Text <> '' then
st:=st+' semester='''+ combobox1.Text+''' ';
 query3.Close;
query3.sql.Clear;
query3.sql.add(st);
query3.Open;
dbgrid1.Visible:=true;
s:=strtofloat(dbedit1.Text);
d:=strtoint(dbedit2.Text);
f:=s/d;
str(f:3:2,se);
edit3.Text:=se;
end;
```

```
procedure TForm10.Exit1Click(Sender: TObject);
begin
close;
end;

procedure TForm10.FormCreate(Sender: TObject);
begin
edit2.text:=table3.Fields[1].Text;
end;

procedure TForm10.FormClose(Sender: TObject; var Action: TCloseAction);
begin
action:=cafree;
end;
```

9- Update Courses of Students Source Code:

```
unit Unit20;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Db, Menus, DBTables, DBCtrls, Grids, DBGrids, StdCtrls, ExtCtrls;

type
  TForm20 = class(TForm)
    Panel1: TPanel;
    Label1: TLabel;
    Button1: TButton;
    Edit1: TEdit;
    Panel2: TPanel;
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    Table1: TTable;
    MainMenu1: TMainMenu;
    Exit1: TMenuItem;
    DataSource3: TDataSource;
    Table2: TTable;
    Label2: TLabel;
    procedure Exit1Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Save1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
```

```
    { Public declarations }
  end;

var
  Form20: TForm20;

implementation

{$R *.DFM}

procedure TForm20.Exit1Click(Sender: TObject);
begin
close;
end;

procedure TForm20.Button1Click(Sender: TObject);
begin
table2.Filtered:=false;
table2.filter:='st_no='+edit1.Text;
table2.Filtered:=true;
end;

procedure TForm20.Save1Click(Sender: TObject);
begin
table2.Post;
end;
procedure TForm20.FormClose(Sender: TObject; var Action: TCloseAction);
begin
action:=cafree;
end;
procedure TForm20.FormCreate(Sender: TObject);
begin
end;

end.
```

10- Update Semester Courses of students Source Code:

```
unit Unit21;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  DBTables, Db, Menus, DBCtrls, StdCtrls, Grids, DBGrids, ExtCtrls;

type
  TForm21 = class(TForm)
    Panel1: TPanel;
    Label2: TLabel;
```

```
    Label3: TLabel;
    Label4: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    ComboBox1: TComboBox;
    DBNavigator1: TDBNavigator;
    MainMenu1: TMainMenu;
    Exit1: TMenuItem;
    DataSource1: TDataSource;
    Query1: TQuery;
    DataSource2: TDataSource;
    Table1: TTable;
    DataSource3: TDataSource;
    Table2: TTable;
    Table3: TTable;
    Query2: TQuery;
    button3: TButton;
    DBGrid1: TDBGrid;
    Label1: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure Exit1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure button3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form21: TForm21;

implementation

{$R *.DFM}

procedure TForm21.Button1Click(Sender: TObject);
var sw:string;
begin
sw:='select * from grade where';

if edit1.Text <> '' then
sw:=sw+' st_no='''+ edit1.Text+''' and ';
if edit2.Text <>'' then
sw:=sw+' academic_year='''+ edit2.Text+''' and';
if combobox1.Text <> '' then
sw:=sw+' semester='''+ combobox1.Text+''' ';
query1.Close;
query1.sql.Clear;
```

```
query1.sql.add(sw);
query1.Open;
dbgrid1.Visible:=true;
end;

procedure TForm21.Exit1Click(Sender: TObject);
begin
close;
end;

procedure TForm21.FormCreate(Sender: TObject);
begin
edit2.text:=table3.Fields[1].Text;
end;
procedure TForm21.Button2Click(Sender: TObject);
var sr:string;
begin
sr:='delete from grade where course_code=" '+query1.Fields[1].Text+'" ';
showmessage('delete from grade where course_code=" '+query1.Fields[1].Text+'" ');
query2.Close;
query2.sql.Clear;
query2.sql.add(sr);
query2.ExecSQL;
query1.Close;
query1.open;
end;
procedure TForm21.button3Click(Sender: TObject);
begin
table1.Filtered:=false;
table1.Filter:='    st_no='+edit1.Text+'    and    semester='"+combobox1.text+'"    and
                academic_year='"+edit2.text+"";
table1.Filtered:=true;
end;

end.
```

11- Semester and Year Selection Source Code:

```
unit Unit22;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Db, DBTables, StdCtrls, ExtCtrls, DBCtrls, Mask, Menus;

type
  TForm22 = class(TForm)
    DBEdit1: TDBEdit;
    Semester: TDBRadioGroup;
```

```
    Table1: TTable;
    DataSource1: TDataSource;
    Label1: TLabel;
    MainMenu1: TMainMenu;
    Exit1: TMenuItem;
    Label2: TLabel;
    procedure Exit1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form22: TForm22;

implementation

{$R *.DFM}

procedure TForm22.Exit1Click(Sender: TObject);
begin
close;
table1.Post;

end;

procedure TForm22.FormClose(Sender: TObject; var Action: TCloseAction);
begin
action:=cafree;
end;
end
```

12- Grades of students Report Source Code:

```
unit Unit18;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  DBTables, Db, Menus, Qrctrls, QuickRpt, DBCtrls, StdCtrls, Grids,
  DBGrids, ExtCtrls;

type
  TForm18 = class(TForm)
    Panel1: TPanel;
```

```
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Button1: TButton;
    DBGrid1: TDBGrid;
    Edit2: TEdit;
    ComboBox1: TComboBox;
    DBLookupComboBox1: TDBLookupComboBox;
    QuickRep1: TQuickRep;
    QRGroup1: TQRGroup;
    QRDBText5: TQRDBText;
    QRDBText6: TQRDBText;
    QRDBText7: TQRDBText;
    QRBand1: TQRBand;
    QRDBText1: TQRDBText;
    QRDBText2: TQRDBText;
    QRDBText3: TQRDBText;
    QRDBText4: TQRDBText;
    QRGroup2: TQRGroup;
    QRLabel1: TQRLabel;
    QRLabel2: TQRLabel;
    QRLabel3: TQRLabel;
    QRLabel4: TQRLabel;
    QRBand2: TQRBand;
    QRSysData1: TQRSysData;
    QRSysData2: TQRSysData;
    Button2: TButton;
    Table1: TTable;
    MainMenu1: TMainMenu;
    Exit1: TMenuItem;
    DataSource2: TDataSource;
    Table2: TTable;
    DataSource3: TDataSource;
    DataSource1: TDataSource;
    Query1: TQuery;
    Query2: TQuery;
    MainMenu2: TMainMenu;
    MenuItem1: TMenuItem;
    Table3: TTable;
    DataSource4: TDataSource;
    QRBand3: TQRBand;
    QRLabel5: TQRLabel;
    QRDBText8: TQRDBText;
    Table4: TTable;
    procedure Button2Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Exit1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
```

```
 public
  { Public declarations }
 end;

var
 Form18: TForm18;

implementation

{$R *.DFM}

procedure TForm18.Button2Click(Sender: TObject);
begin
form18.QuickRep1.Preview;
form18.Visible:=false;
end;

procedure TForm18.Button1Click(Sender: TObject);
VAR
str: string;
 begin
str:='select * from grade where';

if DBLookupComboBox1.Text <> '' then
str:=str+' course_code='''+ DBLookupComboBox1.Text+''' and ';
if edit2.Text <>'' then
str:=str+' academic_year='''+ edit2.Text+''' and';
if combobox1.Text <> '' then
str:=str+' semester='''+ combobox1.Text+''' ';
query1.Close;
query1.sql.Clear;
query1.sql.add(str);
query1.Open;
 dbgrid1.Visible:=true;

end;

procedure TForm18.Exit1Click(Sender: TObject);
begin
close;
end;

procedure TForm18.FormCreate(Sender: TObject);
begin
edit2.text:=table4.Fields[1].Text;
end;

end.
```

13- All Courses of student Report Source Code:

```
unit Unit19;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Db, Grids, DBGrids, StdCtrls, DBTables, Qrctrls, QuickRpt, ExtCtrls,
  Menus;

type
  TForm19 = class(TForm)
    QuickRep1: TQuickRep;
    QRGroup1: TQRGroup;
    QRBand1: TQRBand;
    QRGroup2: TQRGroup;
    QRBand2: TQRBand;
    QRDBText1: TQRDBText;
    QRDBText2: TQRDBText;
    QRDBText3: TQRDBText;
    QRDBText4: TQRDBText;
    Query1: TQuery;
    Button1: TButton;
    Edit1: TEdit;
    DBGrid1: TDBGrid;
    DataSource1: TDataSource;
    Button2: TButton;
    QRDBText5: TQRDBText;
    QRDBText6: TQRDBText;
    QRDBText7: TQRDBText;
    Ret: TQRDBText;
    QRExpr1: TQRExpr;
    QRExpr2: TQRExpr;
    QRExpr3: TQRExpr;
    QRDBText8: TQRDBText;
    QRDBText9: TQRDBText;
    QRLabel1: TQRLabel;
    QRLabel2: TQRLabel;
    QRLabel3: TQRLabel;
    QRLabel4: TQRLabel;
    QRLabel5: TQRLabel;
    QRLabel6: TQRLabel;
    QRLabel7: TQRLabel;
    QRLabel8: TQRLabel;
    QRLabel9: TQRLabel;
    QRLabel10: TQRLabel;
    QRLabel11: TQRLabel;
    QRLabel12: TQRLabel;
```

```
    QRLabel13: TQRLabel;
    Panel1: TPanel;
    Label1: TLabel;
    QRBand3: TQRBand;
    QRSysData1: TQRSysData;
    MainMenu1: TMainMenu;
    Exit1: TMenuItem;
    Label2: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Exit1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form19: TForm19;
implementation

{$R *.DFM}

procedure TForm19.Button1Click(Sender: TObject);
VAR
str: string;

begin

str:='select*from grade where';

if edit1.Text <> '' then
str:=str+' st_no='"+ edit1.Text+'"  ';
query1.Close;
query1.sql.Clear;
query1.sql.add(str);
query1.Open;
 end;
procedure TForm19.Button2Click(Sender: TObject);
begin
form19.QuickRep1.Preview;
form19.Visible:=false;
end;

procedure TForm19.Exit1Click(Sender: TObject);
begin
close;
end;
```

```
procedure TForm19.FormClose(Sender: TObject; var Action: TCloseAction);
begin
action:=cafree;
end;

end.
```

14-Update Undergraduate Curriculum:

```
unit Unit25;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Menus, StdCtrls, ExtCtrls, DBCtrls, Grids, DBGrids, Db, DBTables;

type
  TForm25 = class(TForm)
    Table1: TTable;
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    Label1: TLabel;
    MainMenu1: TMainMenu;
    Exit1: TMenuItem;
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure Exit1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form25: TForm25;
implementation
{$R *.DFM}
procedure TForm25.FormClose(Sender: TObject; var Action: TCloseAction);
begin
action:=cafree;
end;
procedure TForm25.Exit1Click(Sender: TObject);
begin
close;
end;
end.
```